Springer Tracts in Electrical and Electronics Engineering

Debasis Gountia
Dilip Kumar Dalei
Subhankar Mishra *Editors*

Information Technology Security

Modern Trends and Challenges

Springer

Springer Tracts in Electrical and Electronics Engineering

Springer Tracts in Electrical and Electronics Engineering (STEEE) publishes the latest developments in Electrical and Electronics Engineering - quickly, informally and with high quality. The intent is to cover all the main branches of electrical and electronics engineering, both theoretical and applied, including:

- Signal, Speech and Image Processing
- Speech and Audio Processing
- Image Processing
- Human-Machine Interfaces
- Digital and Analog Signal Processing
- Microwaves, RF Engineering and Optical Communications
- Electronics and Microelectronics, Instrumentation
- Electronic Circuits and Systems
- Embedded Systems
- Electronics Design and Verification
- Cyber-Physical Systems
- Electrical Power Engineering
- Power Electronics
- Photovoltaics
- Energy Grids and Networks
- Electrical Machines
- Control, Robotics, Automation
- Robotic Engineering
- Mechatronics
- Control and Systems Theory
- Automation
- Communications Engineering, Networks
- Wireless and Mobile Communication
- Internet of Things
- Computer Networks

Within the scope of the series are monographs, professional books or graduate textbooks, edited volumes as well as outstanding PhD theses and books purposely devoted to support education in electrical and electronics engineering at graduate and post-graduate levels.

Review Process

The proposal for each volume is reviewed by the main editor and/or the advisory board. The books of this series are reviewed in a single blind peer review process.

Ethics Statement for this series can be found in the Springer standard guidelines here https://www.springer.com/us/authors-editors/journal-author/journal-author-helpdesk/before-you-start/before-you-start/1330#c14214

Debasis Gountia · Dilip Kumar Dalei ·
Subhankar Mishra
Editors

Information Technology Security

Modern Trends and Challenges

Springer

Editors

Debasis Gountia
Odisha University of Technology
and Research (OUTR)
Bhubaneswar, India

Dilip Kumar Dalei
Defence Research & Development
Organisation (DRDO)
Bengaluru, India

Subhankar Mishra
National Institute of Science Education
and Research (NISER)
Bhubaneswar, Odisha, India

ISSN 2731-4200 ISSN 2731-4219 (electronic)
Springer Tracts in Electrical and Electronics Engineering
ISBN 978-981-97-0406-4 ISBN 978-981-97-0407-1 (eBook)
https://doi.org/10.1007/978-981-97-0407-1

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Paper in this product is recyclable.

Preface

Cyber-attacks are a ripe rising target phenomenon in the today's digital world. The security threats and breaches in cyberspace have rapidly become more common, creative and critical. Many of these attacks are carried out to bypass an organisation's information security and gain access to sensitive data, which can be stolen and damaged. For this reason, the Information Technology (IT) in an organisation is a critical component that needs to be strengthened to thwart a security breach.

IT security is a set of cyber security strategies and techniques that prevent unauthorised access to digital assets of an organisation such as computers, networks, and data. It protects digital information from malicious threats and potential security breaches that can have a huge impact on the organisation. The main thrust area of IT security comprises Network security, Endpoint security, Internet security, Cloud security, Application security etc. This book looks at the current trends and challenges in the field of IT security. It covers a wide range of topics related to IT security in today's cyber threat landscape. The individual chapters are crafted by industrial practitioners and academic experts from security domain.

Bhubaneswar, India Debasis Gountia
Bengaluru, India Dilip Kumar Dalei
Bhubaneswar, India Subhankar Mishra

Contents

Editors and Contributors

About the Editors

Debasis Gountia was awarded his Ph.D. degree in Information Security from the Indian Institute of Technology (IIT), Roorkee, Master of Technology degree in Computer Science and Engineering from IIT Kharagpur, and Bachelor of Technology degree in Computer Science and Engineering from the University College of Engineering (UCE), Burla, India. His research interests include cryptography, electronic design automation of microfluidic Lab-on-a-Chips, computer security, artificial intelligence, blockchain, Internet of Things, machine learning, and distributed systems. He has authored 19 international referred journals, 18 conference proceedings, 3 books, and 5 book chapters, and 4 filed patents in the aforementioned areas. He has carried out several international and national projects including SERB Govt. of India, Microsoft, COVID-19 Grant, ICIT, DETECT-X for Innovate4Health design sprint, submitted proposals for NISAR and Geospatial project of ISRO, DST, OURIIP, BIRAC, etc. He has been a reviewer of various journals and conferences of national and international repute. He is a professional member of ACM, IEEE, SMIE, FSIESRP, and IFERP.

Dilip Kumar Dalei received his Master's degree in Information Technology Security (ITS) from Masaryk University, Czech Republic and M.Tech. in Computer Science & Engineering (CSE) from Indian Institute of Technology (IIT), Kharagpur. He has completed his B.Tech. degree in Computer Science & Engineering from National Institute of Technology (NIT), Rourkela. Currently, he is working as a Scientist in Centre for Artificial and Robotics (CAIR), DRDO, Bengaluru, India. He has 18 years of R&D experience in defence software product development. His research interests are information security, computer security, artificial intelligence (AI), machine learning (ML), data visualisation, cloud computing, GPU computing, algorithm design and geographical information system (GIS). He has authored several international conference proceedings in the aforementioned areas.

Subhankar Mishra received his Ph.D. degree in Computer Science from the University of Florida, USA, in 2016. He is a Reader F in the School of Computer Sciences, NISER Bhubaneswar, India. His research interests include critical infrastructure cyber security, Machine Learning and privacy. Previously, he worked as an Assistant Professor at Indian Institute of Technology (IIT) Roorkee and Research Associate at Oak Ridge National Laboratory. He finished his B.Tech. in Computer Science and Engineering from National Institute of Technology (NIT), Rourkela, India in 2010. He has authored several international conference proceedings, journals, and transactions in the aforementioned areas.

Contributors

Pravas Ranjan Bal Birla Institute of Technology Mesra, Ranchi, Jharkkand, India

Padmalochan Bera Indian Institute of Technology Bhubaneswar, Bhubaneswar, India

Trishul V. Biradar Department of Computer Science and IT, Jain Deemed-to-Be-University, Bengaluru, India

Susil Kumar Bishoi DRDO, CAIR, Bangalore, India

Ikshit Chaturvedi Birla Institute of Science and Technology Pilani, Dubai, UAE

Vijay Kumar Chaurasiya Department of Information Technology, Indian Institute of Information Technology-Allahabad, Prayagraj, India

Sangay Chedup Department of ECE, Jigme Namgyel Engineering College, Dewathang, Bhutan

Saswat Das National Institute of Science Education and Research, an OCC of Homi Bhabha National Institute, Odisha, India

Mohan Kumar Dehury Amity Institute of Information Technology, Amity University Jharkhand, Ranchi, India

Suman Garai Department of Computer Science and IT, Jain Deemed-to-Be-University, Bengaluru, India

R. Harish TIFAC-CORE in Cyber Security, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India

Akash Kumar Department of CSE, Siksha ‘O’ Anusandhan, Bhubaneswar, India

Sumit Kumar Department of CSE, Thapar Institute of Engineering and Technology, Patiala, India

Praveen Likhar Centre for Artificial Intelligence and Robotics (CAIR), Defence R&D Organisation (DRDO), Bengaluru, India

Vashek Matyas Masaryk University, Brno, Czech Republic

Subhankar Mishra National Institute of Science Education and Research, an OCC of Homi Bhabha National Institute, Odisha, India

Bhabendu Kumar Mohanta Department of CSE, Koneru Lakshmaiah Education Foundation, Vijayawada, AP, India

Raja Muthalagu Birla Institute of Science and Technology Pilani, Dubai, UAE

Pranav M. Pawar Birla Institute of Science and Technology Pilani, Dubai, UAE

K. Praveen TIFAC-CORE in Cyber Security, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India

Jitendra Kumar Rout Department of CSE, National Institute of Technology Raipur, Raipur, India

Kasturi Routray Indian Institute of Technology Bhubaneswar, Bhubaneswar, India

Sujit Sangram Sahoo Department of Information Technology, Indian Institute of Information Technology, Allahabad, Prayagraj, India

P. S. Tamizharasan Birla Institute of Science and Technology Pilani, Dubai, UAE

Ravi Shankar Yadav Centre for Artificial Intelligence and Robotics (CAIR), Defence R&D Organisation (DRDO), Bengaluru, India

Taskeen Zaidi Department of Computer Science and IT, Jain Deemed-to-Be-University, Bengaluru, India

Chapter 1
Zero Trust Security Architecture for Digital Privacy in Healthcare

**Ikshit Chaturvedi, Pranav M. Pawar, Raja Muthalagu,
and P. S. Tamizharasan**

Abstract Ensuring the security and privacy of confidential medical data has become increasingly crucial with the widespread digitization of healthcare institutions. This chapter proposes the adoption of a Zero Trust Security Architecture (ZTSA) to enhance digital privacy in hospitals. By challenging the traditional notion of trust, zero trust security provides robust protection against unauthorized access to sensitive patient data and mitigates the risks of data theft or leakage. The core principle of ZTSA assumes that no user or device on a network can be trusted by default, necessitating continuous verification for secure access and data protection. Zero trust security offers significant benefits for patient privacy in the healthcare industry. Through the implementation of stringent access controls and encryption techniques, ZTSA serves as a critical safeguard for protecting patient privacy. This study contributes to the field by implementing ZTSA within a Django-based hospital system, incorporating conditional access controls, authentication measures, and encryption methods to fortify system security. The utilization of conditional access controls in ZTSA enables fine-grained control over resource access, reducing the risk of unauthorized entry. Additionally, the implementation of multi-factor authentication ensures that only authenticated and authorized users can access vital information. Robust encryption methods employed by ZTSA ensure the security of data during transmission and storage, significantly reducing the likelihood of data breaches and unauthorized data access. The implementation of ZTSA yields notable improvements in the security of the hospital system, as evidenced by the outcomes of our study. By adopting Zero Trust principles, the system provides robust protection for patient data and preserves

I. Chaturvedi · P. M. Pawar (✉) · R. Muthalagu · P. S. Tamizharasan
Birla Institute of Science and Technology Pilani, Dubai Campus, Dubai, UAE
e-mail: pranav@dubai.bits-pilani.ac.in

I. Chaturvedi
e-mail: f20190111@dubai.bits-pilani.ac.in

R. Muthalagu
e-mail: raja.m@dubai.bits-pilani.ac.in

P. S. Tamizharasan
e-mail: tamizharasan@dubai.bits-pilani.ac.in

D. Gountia et al. (eds.), *Information Technology Security*, Springer Tracts in Electrical
and Electronics Engineering, https://doi.org/10.1007/978-981-97-0407-1_1

individual privacy. The effectiveness of ZTSA in enhancing digital privacy in hospitals is demonstrated conclusively. In conclusion, this chapter highlights the significance of Zero Trust Security Architecture in enhancing digital privacy in hospitals. Through the enforcement of conditional access controls, authentication procedures, and encryption techniques, the system becomes resilient against threats, ensuring the security of critical patient data and safeguarding individual privacy.

Keywords Zero Trust Security Architecture · Digital privacy · Hospitals · Healthcare data · Conditional access controls · Encryption techniques

Introduction

In the current digital era, data are incredibly important for influencing decisions, spurring innovation, and fostering growth [1]. Data are essential for enabling wise strategic decisions and operational optimization [1]. Big data analysis has altered processes in the fields of technology, marketing, healthcare, and finance by enhancing client experiences and stimulating innovation. Technology behemoths like Google, Facebook, and Amazon accumulate enormous amounts of customer data, enabling customized services and targeted advertising. In order to advance patient care and research, the healthcare industry stores a vast amount of medical records and genetic data. However, mishandling data can have detrimental consequences. Personal information becomes a prime target for cybercriminals exploiting it for identity theft, fraud, and social engineering. The erosion of privacy, trust, and civil liberties occurs through state-sponsored surveillance, data breaches, and unauthorized data sharing [2]. Furthermore, the aggregation of personal information enables targeted manipulation, influencing political environments and shaping public opinion.

The security and privacy of sensitive data, especially in the healthcare industry, have emerged as significant concerns in the digital age. The attacker's primary targets are still in the healthcare sector. The industry's growing adoption of cutting-edge medical technologies exposes it to more cyber threats, necessitating a change in security policy [3]. Every week, news reports discuss data breaches that reveal personal health information. These occurrences can have a substantial impact on people whose information is released, as well as a loss of confidence in the compromised institutions. For the harmed firms, a significant financial hit from lost revenue, reputational harm, and possible fines related to the breach can also be a serious concern. Any breach or data loss event has a meaningful impact on a healthcare company, regardless of whether it was caused by unintentional exposure or a deliberate act.

The digitization of patient records, medical systems, and healthcare services has led to the accumulation of substantial amounts of personal information within hospitals and other healthcare institutions. Therefore, safeguarding the security of digital patient records is crucial. Among various industries, the healthcare sector stands out as a highly data-intensive domain. Healthcare organizations collect and store comprehensive patient records, encompassing personal, medical, and financial data such as patient records containing highly sensitive information, including medical history, test results, diagnoses, and treatment plans. This concentration of data makes the healthcare industry an attractive target for malicious actors seeking to exploit vulnerabilities and gain unauthorized access to confidential information. The consequences of data breaches in healthcare can be profound, with potential implications such as identity theft, medical fraud, and risks to patient safety. Additionally, patient records often include personally identifiable information (PII) such as names, addresses, social security numbers, and insurance details. When combined with other stolen data, this information can facilitate fraudulent activities and identity theft. Additionally, data that healthcare institutions have is highly sought-after on the black market. One such example of a cyberattack on the healthcare industry is SingHealth Data Breach [4]. In 2018, 1.5 million patients were impacted by a significant data breach at Singapore's largest healthcare organization, SingHealth. Personal data, including names, addresses, and national identification numbers, were stolen by hackers who illegally acquired access to the group's database. The incident highlighted the necessity for strong cybersecurity measures and showed flaws in the organization's security infrastructure.

The majority of healthcare businesses use traditional cyber security measures, such as firewalls, to defend the network's perimeter while presuming all internal communications are secure and allowed. Threat actors are deploying sophisticated attack vectors, such as phishing, fileless malware, ransomware, and zero-day assaults, to infiltrate the network by preying on this assumption. Malware may divulge a user's login information, which would grant them restricted access to the network of their company. The ability to travel across the private network and install further malware on privileged devices that are closer to the data is made possible by network access [5].

The cycle is then repeated. These assaults encounter minimal resistance and spread swiftly once they are inside because many healthcare organizations primarily rely on perimeter security and a trust-based security strategy without sufficient internal network segmentations. All of these attack techniques share the trait of preying on the implicit faith in the security posture of the majority of healthcare companies. Trust has developed into a weakness that is just as risky as any other. The shortcomings of perimeter-centric security, as well as the outdated hardware and software required to achieve it, are fixed by zero trust.

The most dependable course of action is to develop a zero-trust security architecture (ZTSA) to fight against both internal and external attacks. The foundation of the zero-trust security concept is the idea that no connection should be trusted unless it has been specifically authorized. The adoption of zero-trust security heralds a paradigm shift from reactive to proactive security, where the objective is to prevent the breach rather than react after it has occurred. Regardless of the user's location or network limits, ZTSA focuses on securing access to resources based on identification, context, and risk assessment. ZTSA considerably lowers the danger of unwanted access, data breaches, and the compromising of digital patient records by using granular conditional access controls, robust authentication systems, and cutting-edge encryption techniques.

There are many benefits to implementing Zero Trust Security Architecture (ZTSA) in healthcare [5] settings, including improved data security and the protection of digital patient information. Healthcare organizations can establish exact control over access to patient records by implementing ZTSA principles, guaranteeing that only authorized people can view and edit sensitive data. Granular conditional access controls can be implemented, which is a significant advantage of ZTSA. This indicates that particular criteria, such as the user's role, location, device security posture, and network environment, are taken into account when deciding whether to give access to patient records. Healthcare organizations can impose stringent control over who can access specific patient records and under what circumstances by creating fine-grained access restrictions. This lessens the chance that unauthorized people will access sensitive data and aids in preventing data breaches. Real-time risk evaluations and adaptive security measures are also made possible by ZTSA [6]. ZTSA can identify and assess potential security threats in real time by continually monitoring user activity, device health, and network circumstances. Any suspicious or unusual activity can immediately send out notifications, enabling security personnel to look into it and take appropriate action. The integrity of digital patient records is guaranteed by this proactive approach to security, which also helps to reduce risks.

In conclusion, the protection of digital patient records is crucial in the healthcare industry due to the sensitivity of the information involved and the potential risks associated with unauthorized access. The application of ZTSA in healthcare environments provides a thorough security architecture that solves the particular difficulties healthcare organizations confront in safeguarding digital patient records. Strong authentication systems, real-time risk assessment capabilities, precise access control, and encryption tools all help to improve the security and privacy of patient data. By deploying ZTSA, healthcare organizations may create a strong security posture, reduce the likelihood of data breaches and guarantee the confidentiality and integrity of electronic patient records. The subsequent sections of this introduction will delve into the history and components of the concept known as zero trust. Furthermore, within Sects. 1.3 and 1.4 of this paper, we have extensively examined the application of zero trust principles to enhance digital privacy in the healthcare domain. Notably, in Sect. 1.5, we have meticulously compared various research endeavors, while Sects. 1.6 and 1.7 elucidate the comprehensive design, functionality, and implementation details of the proposed healthcare system.

Zero Trust Security for Digital Privacy

Introduction to Zero Trust

In the rapidly evolving landscape of cybersecurity, conventional perimeter-based security strategies are proving ineffective at properly protecting sensitive data and systems. The weaknesses of perimeter defenses have been exposed by the emergence of sophisticated cyber attacks, the rise of remote labor, and the spread of cloud services. In an approach to counter fight these issues, a developing collection of cybersecurity concepts known as zero trust (ZT) has evolved that shifts the focus of defenses away from rigid, network-based perimeters and toward users, assets, and resources. Planning industrial and enterprise infrastructure and workflows using zero-trust principles is known as a zero-trust architecture (ZTA). Zero trust presupposes that there is no implicit trust given to assets or user accounts based only on their geographic or network location (for example, local area networks as opposed to the internet) or on the type of ownership of the asset (for example, corporate or private ownership). In zero trust, Rose et al. [7] authentication and authorization are performed separately for both the user and the device before a session with an enterprise resource is established. This means that users and devices are not trusted by default, and they must be authenticated and authorized before they are granted access to enterprise resources. This helps to protect against cyberattacks, as it makes it more difficult for attackers to gain access to sensitive data or systems. Bring your own device (BYOD), remote users, and cloud-based assets that are not situated inside an enterprise-owned network perimeter are examples of trends in enterprise networks that have prompted the development of zero trust. As the network location is no longer considered to be the primary component of the security posture of the resource, zero trust focuses on securing resources (assets, services, workflows, network accounts, etc.), not network segments.

History and Background of Zero Trust

The concept of zero trust has a history predating its official term. A more secure enterprise approach, known as "black core" (BCORE), emerged from a collaboration between the Department of Defense and the Defense Information Systems Agency (DISA). BCORE introduced a shift from a perimeter-based security model to a transaction-based one. Additionally, the notion of deperimeterization gained popularity in 2004 through the efforts of the Jericho Forum, emphasizing the limitations of relying solely on network location and static protections over extended network segments. In 2010, Kindervag [8], a Forrester Research analyst, proposed the term "zero trust" as a response to the shortcomings of conventional security methods. Kindervag recognized the need for a new strategy to address risks posed by both external and internal threats. The fundamental principle of zero trust is rooted in the belief that no person or device, regardless of their location or network boundaries, should be inherently trusted. Traditional security approaches, often referred to as the “castle-and-moat” strategy, relied on perimeter defenses like firewalls and Virtual Private Networks (VPNs) to protect internal networks from external threats. However, the evolving interconnectedness and mobility of modern enterprises have blurred the lines between internal and external networks. Several factors have contributed to the development of zero trust, such as the rise of cloud computing and the use of Software-as-a-Service (SaaS) applications [9], which has shifted the focus from securing the network perimeter to safeguarding access to cloud-based resources. Traditional security methods are no longer effective due to the widespread use of mobile devices and remote work [8]. A more individualized and identity-based approach to security is necessary, prioritizing user authentication and device posture over network-centric controls. Zero trust has gained recognition for its effectiveness in mitigating cyber risks and has become well-known across various industries, including finance, healthcare, and government, where safeguarding sensitive data are paramount [8]. The National Institute of Standards and Technology (NIST) has endorsed the concept of zero trust, providing recommendations and standards for its implementation.

Key Principals and Benefits of Zero-Trust Architecture

A secure and resilient environment is made possible by a set of fundamental principles that are the foundation of zero-trust architecture. These rules make ensuring that security precautions are constantly used and modified to reflect the changing threat environment. The following are some of the main zero-trust architecture tenets:

a. **Never Trust, Always Verify**: One of the cornerstones of zero-trust security is the maxim “never trust, always verify.” It underlines the necessity of operating with the assumption that any user, object, or software could pose a risk. Every access request must be authenticated, authorized, and confirmed before access is granted, rather than depending on trust based on network location or previous encounters. The foundation of this theory is the notion that the network perimeter is no longer an effective barrier against cyberattacks. Attackers can now simply get around standard perimeter protections like firewalls and VPNs. As a result, it is crucial to put security controls in place that confirm each user’s, device’s, and applications before allowing them access to resources [10].
b. **Least Privilege Access**: According to the security principle of least privilege access, users, devices, and programs should only be given the minimal amount of access required to do their duties. By restricting the attacker’s ability to move laterally within the network or increase their privileges, this strategy lessens the potential damage in the event of a security breach. Least privilege access can be implemented in a variety of ways. The usage of role-based access control (RBAC) is one popular strategy. RBAC enables you to create roles that specify

the rights connected to specific job functions. Then, users are given roles based on the duties of their jobs [11].

c. **Micro-segmentation**: A security tactic called micro-segmentation separates a network into smaller, more secure sections. Because each segment has its own security guidelines and access restrictions, it is possible to guard against unwanted access to private information and resources. As the attacker would only have access to the information or resources in the segment they have infiltrated, this strategy also minimizes the possible damage in the event of a breach. Firewalls, network segmentation tools, and identity and access management (IAM) systems are just a few of the technologies that can be used to execute micro-segmentation. The particular technology employed will be determined by the requirements of the enterprise. Despite being a relatively new security tactic, micro-segmentation is swiftly gaining acceptance because of how well it defends against contemporary threats [12]. Micro-segmentation offers several advantages:

    i. Enhanced security: Micro-segmentation increases security by making it more difficult for hackers to access critical information or resources, as they would need to breach multiple segments.
    ii. Reduced risk: By limiting the impact of a breach, micro-segmentation helps to mitigate the risk of data breaches and other cyberattacks.
    iii. Improved operational effectiveness: Micro-segmentation reduces the reliance on manual security measures, leading to enhanced operational efficiency.
    iv. Compliance support: Micro-segmentation aids businesses in meeting data protection regulations such as HIPAA and GDPR, ensuring compliance with legal requirements.

d. **Multifactor Authentication**: Before allowing access, make users submit many kinds of identification verification. Since MFA requires the attacker to go beyond multiple stages of authentication, it lowers the danger of unauthorized access caused by stolen credentials.
e. **Context-aware Access Controls**: Based on contextual information such as user role, device posture, location, and risk considerations, make access decisions. With this strategy, access controls are guaranteed to change with the environment and accommodate for any threats or vulnerabilities that may exist [13].
f. **Continuous Monitoring and Validation**: To spot potential dangers or anomalies, monitor user activity, device health, and application behavior in real time. Verify security configurations, rules, and controls frequently to make sure they work to counter evolving threats.
g. **Data Protection**: Implementing encryption, tokenization, and other data security measures will help to safeguard sensitive data while it is at rest, in transit, and during processing. To stop unauthorized access to sensitive data, put in place strict access restrictions and monitoring.

h. **User and Entity Behavior Analytics (UEBA)**: Analyze user and entity behavior to spot trends and oddities that could point to threats or malicious action. Organizations may respond proactively to possible security issues and reduce the potential harm thanks to UEBA [14].

Logical Component of Zero Trust Architecture

A ZTA deployment in an organization is made up of five main logical parts, i.e. *subject*, *resource, policy decision point* (PDP), *policy enforcement point* (PEP), and *supplement*. Both on-premises and cloud-based services can be used to operate these components. A user or any device requesting access to company resources is referred to as a subject. Resource, as its name suggests, refers to the corporate or enterprise resource that a subject is requesting. Depending on the request's content, a resource may consist of a single or a collection of resources [8, 15].

- **Policy Engine**: The final decision regarding whether to permit access to a resource for a particular subject is made by the policy engine (PE). To give, deny, or revoke access to the resource, the PE employs enterprise policy as well as input from external sources as input to a trust algorithm. The policy administrator component is coupled with the PE. The policy administrator puts the decision into action when the policy engine has made and logged the decision (as accepted or rejected).
- **Policy Administrator**: The policy administrator (PA) is in charge of opening or closing the communication channel between a subject and a resource. Any session-specific authentication, authentication token, or credential that a client uses to access an enterprise resource would be generated by it. It is directly related to the PE and depends on its choice of whether to approve or reject a session at the end. If the session is approved and the request is verified, the PA sets up the PEP to permit the session to begin. The PA instructs the PEP to disconnect in the event that the session is rejected (or a prior authorization is reversed). Both the PE and PA together make another logical component known as Policy Decision Point (PDP). It is up to a policy decision point (PDP) to decide whether to establish or end communication between a subject and the resource being requested as well as whether to grant or refuse access to the corporate resource.
- **Policy Enforcement Point**: An enterprise resource's connections with a subject are enabled, monitored, and ultimately terminated by a system known as a policy enforcement point (PEP). When requesting information or receiving policy changes from the PA, the PEP speaks with the PA. The policy enforcement point (PEP) receives a request from a subject to access an enterprise resource. The request is forwarded to PDP by PEP. The PDP then issues a command to PEP to enable or stop communication between the subject and the resource after deciding what to do with the request. PEP serves as a gateway between the subject and resources, as we shall see. PEP is responsible for monitoring the network traffic between the subject and the requested resource in addition to regulating the communication flow. The working is described in Fig. 1.1.

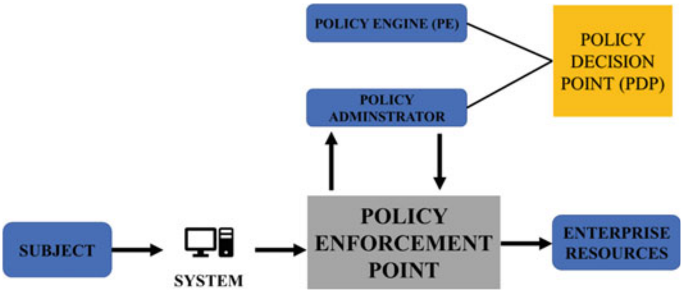


Fig. 1.1 Zero trust architecture core components. *Source* NIST SP 800-207

Zero Trust Architecture for Digital Privacy

Zero Trust Architecture is the key to achieving a secure digital environment and ensuring privacy. Traditional security strategies built on implicit trust and perimeter defense are proving ineffective as cyber threats keep evolving and data breaches increase in frequency. For example, in January 2014 [16], authorities disclosed a security breach involving the personal information of 40% of South Koreans. This was the consequence of a reputable IT contractor being able to download millions of client records to portable media over the course of an 18-month period.

Another such example of a digital privacy crime where Zero Trust Architecture (ZTA) was not used is the 2017 Equifax data breach [17]. A significant hack on Equifax, one of the biggest credit reporting companies, led to the loss of 147 million people's very sensitive personal data. Because of this, Equifax experienced serious repercussions due to stolen personal information, brand harm, legal ramifications, and financial losses by failing to execute a Zero Trust approach. The incident demonstrated the necessity of deploying strong security measures, such as ZTA, to protect sensitive data from illegal access and cyberattacks and to secure digital privacy.

Zero Trust Architecture may have added more security layers to safeguard the sensitive data if it had been put into practice. The "never trust, always verify" tenet of ZTA would have necessitated ongoing user access and activity verification, authorization, and monitoring. The attackers' ability to travel laterally within the Equifax network and gain access to such a vast amount of data would have been more challenging with this strategy.

The use of Zero Trust Architecture has various advantages for online privacy. First off, by reducing the attack surface and making it harder for unauthorized users or hostile actors to access sensitive information or systems, it dramatically improves security. Organizations can successfully reduce the risks of data breaches, identity theft, and illegal access by implementing a zero-trust strategy. Second, Zero Trust Architecture complies with privacy standards and legal obligations. Organizations are under increasing pressure to safeguard user data and maintain privacy compliance as a result of the adoption of strict data protection rules like the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA) [18]. By putting a heavy emphasis on identity-based access restrictions and data protection measures, ZTA offers a framework that makes compliance with these rules easier.

Zero Trust Architecture also accommodates the fluidity and evolution of digital environments. Traditional network borders have blurred due to the growth of cloud services, mobile devices, and remote work. ZTA offers granular control over resource access, regardless of the user's location or the network to which they are connected, enabling enterprises to react to these changes. With no compromise to data privacy, this flexibility boosts productivity and permits safe collaboration.

In light of escalating cyber risks, Zero Trust Architecture has become a groundbreaking strategy for protecting digital privacy. Organizations may improve security, adhere to regulations, and adapt to the changing digital ecosystem by challenging conventional conceptions of trust and employing continuous verification and stringent access controls. Adopting Zero Trust Architecture is crucial for enterprises looking to safeguard their sensitive data and keep users' trust as the value of digital privacy continues to rise.

Zero Trust Architecture Needed in Medical Data Privacy

To secure the security of its interconnected networks, devices, and sensitive patient data, healthcare organizations must employ a Zero Trust security architecture. Adopting a Zero Trust approach is crucial in today's digital environment, where cyber threats are pervasive and the healthcare sector is a prime target, to reduce risks and improve overall security. Healthcare institutions are frequently the target of cyberattacks because patient data are some of the most valuable information that cybercriminals can use as a weapon. The use of stolen credentials and account privilege escalation is one of the most prevalent forms of cyber exploitation [19]. This kind of behavior can result in ransomware, denial of service assaults, or other attacks that could undermine the networks of healthcare institutions and possibly have an effect on patient care. Healthcare institutions that lack identity maturity may be simple prey for cybercriminals, who can then conduct their operations covertly. For instance, a comprehensive multifactor authentication strategy is lacking in many healthcare institutions. A malicious payload can be dropped within a private network by an attacker who takes advantage of this flaw. Healthcare firms can begin by developing a solid identification strategy since zero trust can address this issue.

As already discussed, Never Trust, Always Verify is the guiding philosophy of the Zero Trust security model. It disproves the conventional idea of implicit confidence in networks and creates a framework that demands ongoing user access and activity authentication, authorization, and monitoring, regardless of location or network borders. This strategy makes sure that every person and device is thoroughly vetted before being given access to private information or resources. [8]. Large amounts of sensitive health data, including patient personal and medical information, are handled by the healthcare industry. Cybercriminals can use this data for a variety of nefarious activities, including identity theft, financial fraud, and even ransomware assaults. Additionally, the interconnectedness of healthcare networks, with their myriad interconnected devices and systems, broadens their attack surface and possible weak spots. Healthcare businesses can put in place granular access controls, reliable authentication procedures, and cutting-edge encryption methods by switching to a Zero Trust security paradigm. By taking these precautions, the likelihood of unwanted access, data breaches, and the compromise of private health information is greatly decreased. Additionally, Zero Trust gives businesses the ability to identify and respond to possible threats instantly, improving their capacity for incident response and lessening the severity of security problems.

The use of Zero Trust in healthcare settings is encouraged by a number of industry frameworks and guidelines. In order to secure healthcare systems and data, the National Institute of Standards and Technology (NIST) has underlined the value of Zero Trust architecture [8]. The requirement for access restrictions and security measures to protect electronic protected health information (ePHI) is further emphasized by the Health Insurance Portability and Accountability Act (HIPAA) Security Rule [20]. In order to strengthen the security posture of critical infrastructure, including healthcare systems, the Cybersecurity and Infrastructure Security Agency (CISA) has also suggested the application of Zero Trust principles [21].

Zero Trust Architecture in Healthcare Digital Privacy: Related Work

The security and privacy of sensitive data, particularly in the healthcare industry, have emerged as top issues in today's digital age. Huge volumes of personal information have accumulated within healthcare facilities as a result of the digitalization of patient data, medical systems, and healthcare services [22]. As a result, protecting digital patient records and maintaining the confidentiality of healthcare data have drawn a lot of attention. Healthcare has special and intricate security problems. Numerous patient records, including medical histories, test results, diagnoses, and treatment plans, are gathered and stored by healthcare institutions [23]. Data breaches in the healthcare industry have serious potential implications, including compromised patient privacy, identity theft, medical fraud, and even potential patient injury. In addition, the importance of healthcare data and the possible financial rewards linked with its exploitation make the healthcare sector a top target for cyberattacks.

There has been a lot of study and work done on security for digital privacy in healthcare to solve these issues. To safeguard healthcare data and guarantee the privacy, accuracy, and accessibility of patient information, a number of strategies and methods have been established. In order to reduce the dangers of unapproved access, data breaches, and the compromising of digital medical records, these initiatives attempt to put strong security measures in place. Several important topics have been the focus of this field's research. The creation and application of secure access control techniques to safeguard patient records is one area of focus. To guarantee that only persons with permission may access sensitive data, this includes implementing authentication mechanisms, encryption approaches, and user authorization frameworks. The use of cutting-edge encryption algorithms and secure communication protocols is a crucial component of security for digital privacy in healthcare. These safeguards are designed to keep patient data private and are difficult to intercept or tamper with during transmission and storage. Research has also been done on the use of privacy-preserving methods and safe data exchange frameworks. The objective is to make it possible for authorized parties to share healthcare information while protecting the privacy and confidentiality of specific patient information. To find a balance between data sharing and privacy protection, methods including differential privacy, homomorphic encryption, and safe multiparty computing have been investigated. Additionally, the introduction of cutting-edge technologies like blockchain has spurred interest in their potential uses for protecting digital privacy and safeguarding healthcare data. Blockchain-based solutions increase the reliability and security of healthcare data by providing decentralized and immutable data storage, improved transparency, and auditability. In conclusion, a crucial area of study and development is the protection of digital privacy in healthcare. Significant efforts have been made to put in place strong security measures in order to safeguard patient data against unauthorized access, breaches, and privacy violations. Researchers and practitioners are attempting to build a safe environment that protects the privacy and security of healthcare data by utilizing improvements in access control, encryption, secure data sharing, and upcoming technologies (Table 1.1).

ZTA offers a number of significant benefits for safeguarding healthcare systems as compared to other security technologies including multi-access edge computing (MEC), blockchain, and cryptography. These benefits include a comprehensive security strategy that emphasizes real-time user and device identification, granular access control based on a number of criteria, network segmentation to prevent lateral movement, and interoperability with current systems. ZTA stands out as a potent framework that addresses user authentication, access control, continuous monitoring, and network segmentation. It provides healthcare systems with a multi-layered security strategy for protecting sensitive data and ensuring system integrity and availability. MEC, blockchain, and cryptography all contribute to specific security aspects. Healthcare systems may create a strong security framework that reduces risks and improves overall security by integrating ZTA with other technologies.

Proposed Healthcare System Based on Zero Trust Architecture

In order to ensure the integrity and accessibility of healthcare services, the healthcare sector urgently needs to establish reliable and secure technologies that emphasize patient privacy. We’ve put in place a cutting-edge healthcare ecosystem based on zero trust architecture (ZTA) principles to solve this problem. By using ZTA, we

Table 1.1 Comparative-related work of zero trust architecture in healthcare digital privacy

Reference number	Objective	Methodology	Advantages	Disadvantages
[24]	Enhancing the Security for Healthcare Data using Blockchain Technology	Using blockchain to implement an e-voting system	Blockchain in healthcare improves data security, efficiency, and privacy	Blockchain technology in healthcare has limitations, such as redundancy, complexity, and security problems
[25]	A security awareness and protection system based on zero-trust architecture is proposed for a 5G-based smart medical platform	Addressing issues of 5G network, implementing ZTA to address these issues, integrating 5G network security	Comprehensive testing ensured system functionality, performance, and security	ZTA-based security solutions face challenges in technical implementation and public awareness
[26]	Developing a guide to transitioning healthcare organizations to a Zero-Trust Network Architecture	Testing different security protocols and methods, framework development and testing, analysis of packet flows and latency	Enhanced security, compatibility with older equipment, Granular network segmentation, Consideration of latency	ZTA implementation faces compatibility, financial, simulation, and implementation time limitations
[27]	Using a multi-access edge computing zero-trust security model to improve healthcare cyber resilience	A layered architecture with ZTS and MEC will be implemented to secure healthcare networks	Proposed architecture enhances security with ZTS, MFA, and multiple identity factors	Challenges in implementation, usability, and scalability
[28]	Blockchain-based new generation healthcare applications: security and privacy mechanisms	A 5G-enabled e-healthcare framework with blockchain and privacy-preserving mechanisms was developed	The developed e-healthcare framework offers a stable, secure, and efficient platform for managing health information	The e-healthcare framework faces challenges in implementation and data security

(continued)

Table 1.1 (continued)

Reference number	Objective	Methodology	Advantages	Disadvantages
[29]	Research on a Zero-Trust Medical Security System	Design a zero-trust medical security system with dynamic access control based on RBAC and user behavior risk evaluation	Zero-trust medical security system enhances information security, access control, and effectiveness	The zero-trust medical security system may have low work efficiency due to cumbersome authentication
[30]	Using modern cryptography to secure healthcare	A novel integration of blockchain and cryptographic protocols enhances healthcare IT security	Cryptography enhances healthcare data security, transparency, and efficiency	Blockchain and cryptography-based healthcare infrastructure face challenges in implementation, updates, and adoption

have created a highly secure environment where every person and device is regarded as untrusted and where access is allowed in accordance with a thorough set of conditional access criteria. Our healthcare system uses a variety of high-tech security precautions, and various condition-based access policies such as time-based, location, role-based constraints, etc. Along with this we also make use of multi-factor authentication (MFA) and reliable identity and access management (IAM) protocols. By restricting user rights and imposing exacting authentication processes, these methods aid in establishing strong access restrictions. By utilizing MFA, we provide a second layer of authentication, requiring users to present many forms of identity in order to access critical healthcare services and data.

The developed healthcare system also embraces the idea of fine-grained access control, making sure that users are only given access to the particular information and resources required for their approved duties. The danger of unwanted access or data breaches is reduced because of this granular approach to access management. Additionally, our system uses cutting-edge encryption methods to protect the privacy and accuracy of healthcare data while it is being transmitted and stored.

Concept Behind Conditional Access

A key idea in zero trust architecture (ZTA) is conditional access, which controls whether access to digital resources is granted or denied in accordance with a wide range of requirements [31]. The process of conditional access is shown in Fig. 1.2. By seeing every user, device, and request as potentially malicious, it departs from the conventional perimeter-based security strategy and demands ongoing authentication

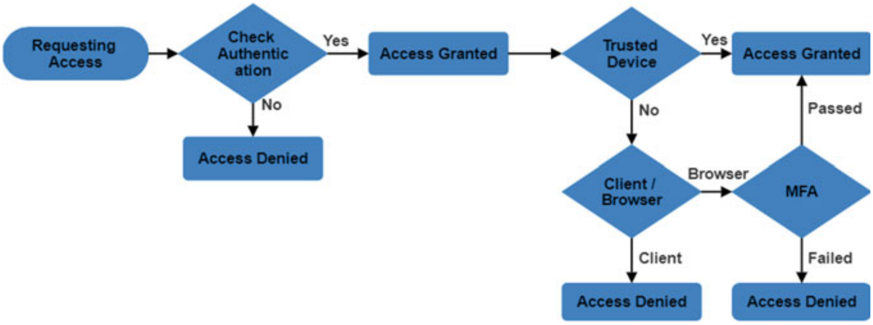


Fig. 1.2 Conditional access policies

and authorization throughout the user session [32]. Organizations may build granular access controls and reduce security risks by imposing conditional access by limiting access to only approved persons, devices, and places.

Different forms of conditional access restrictions are used in zero-trust architecture to determine access rights [32]. These restrictions are made to make sure that access is only provided in accordance with certain criteria, such as user characteristics, device traits, network circumstances, and contextual elements. Let's examine a few prevalent categories of conditional access restrictions:

i. **Device-based restrictions**: These restrictions consider the traits and security posture of the device asking for access. To identify access, variables including device type, operating system version, security updates, and the existence of encryption tools may be taken into consideration [32].
ii. **Location-based restrictions**: These limitations place access restrictions according to the location of the request. Organizations can designate places as trusted or untrusted, and can then grant or refuse access in accordance. By doing so, you may be able to stop illegal access attempts from dangerous or unknown areas [32].
iii. **User-based constraints**: These restrictions are based on the characteristics of the user, such as their identity, group membership, or role. Access choices may be made based on a user's job function, level of seniority, or certain permissions [32].
iv. **Network-based restrictions**: These restrictions evaluate the safety and reliability of the network from which the request is originated. Access decisions may be influenced by variables such as IP address reputation, network type (such as corporate network or public Wi-Fi), and network security mechanisms (such as firewall, intrusion detection system) [32].
v. **Time-based restrictions**: Time-based restrictions limit access based on predetermined time periods. Access, for instance, may only be permitted during regular business hours or on particular days of the week, prohibiting unwanted access at off-peak times [32].

vi. **Risk-based restrictions**: Risk-based restrictions take into account the general degree of risk connected to the access request. To give risk scores, factors like user behavior, device health, and anomalous activity can be assessed. Access can be granted, refused, or subject to additional authentication factors depending on the risk assessment [32].

Implementation and Working of Developed System

The work used Django, a strong and extremely secure web framework, to create our healthcare system. Django provides a wide range of features and functions that are ideal for our project's needs. It is the best option for building a safe and customized healthcare system due to its built-in security features and customization flexibility [33].

Django is an open-source Python web framework well known for its effectiveness, scalability, and wide range of tools and modules. It adheres to the model-view-controller (MVC) architectural paradigm, making it simple for developers to create intricate online applications. Django offers a structured and organized programming environment and encourages the DRY (Don't Repeat Yourself) tenets.

Django's Security Features

Django has a number of strong security measures that are intended to protect online applications against common flaws. Django has a number of significant security features, such as:

- **Cross-Site Scripting (XSS) Protection**: Django has built-in security features to reduce the impact of XSS attacks. It enforces appropriate encoding and automatically escapes user-generated text, stopping dangerous scripts from running [34].
- **Protection from Cross-Site Request Forgery (CSRF)**: Django has built-in defenses against CSRF threats. In order to ensure that requests come from reliable sources, it creates distinctive tokens for each user session and checks them on form submissions [34].
- **SQL Injection Prevention**: By automatically cleaning user inputs, Django's Object-Relational Mapping (ORM) technology defends against SQL injection attacks. By parameterizing searches, it does away with the necessity for manual cleanup and lessens the possibility of malicious database manipulation.
- **Password Hashing**: Django uses safe password hashing techniques to store user credentials. It employs bcrypt by default, a powerful hashing algorithm that considerably slows down brute-force assaults.

- **User Authentication**: Django has a strong user authentication mechanism that securely maintains user sessions and passwords. It supports a number of authentication techniques, including username and password, email confirmation, and interaction with third-party authentication providers.
- **Access Control and Permissions**: Django provides a versatile mechanism for creating user roles, permissions, and access control. It allows for fine-grained control over the features that are available to various user categories, guaranteeing correct authorization and reducing illegal access.
- **Support for HTTPS and SSL/TLS**: Django enables the incorporation of SSL/TLS certificates and provides secure communication over HTTPS. This prevents eavesdropping and manipulation while ensuring the confidentiality and integrity of the data during transmission.
- **Clickjacking Prevention**: Django uses X-Frame-Options headers to perform clickjacking prevention. These headers reduce the possibility of clickjacking attacks by prohibiting the presentation of web pages within iframes on other sites.
- **Session Management**: Using session encryption and defenses against session hijacking and fixation attempts, Django securely handles user sessions. It provides a variety of session backends, including alternatives that are database- and cache-based and allow customization based on particular project needs.

Working of Developed System

Using the Django framework, we have created a complete healthcare system that includes a number of capabilities and strong security measures. The system provides flexible login choices for various users, such as user login, user sign-in, and staff login. The procedure of user login and staff login is shown in Fig. 1.3 and Fig. 1.4, respectively. We have used different roles such as **patient, doctor, accountant, admin**. We have put in place a number of safeguards to make sure user accounts are secure. Users must enter their email address and password during the registration process, with the password being safely kept in an encrypted format. To further ensure that only verified accounts have access to the system, an email verification link is also given to the user's email address for account activation. To protect the system from ghost accounts, new accounts must be verified within 24 h of creation. Accounts that are not verified will be deleted. Additionally, users who repeatedly create same accounts and do not verify them will have their accounts banned. This policy is in place to prevent malicious actors from creating fake accounts for the purpose of spamming, trolling, or other harmful activities.

Users can view pertinent medical information on their customized panel after successfully authenticating. We have additionally secured the reports so that a patient can only access them after entering their medical record number as their password

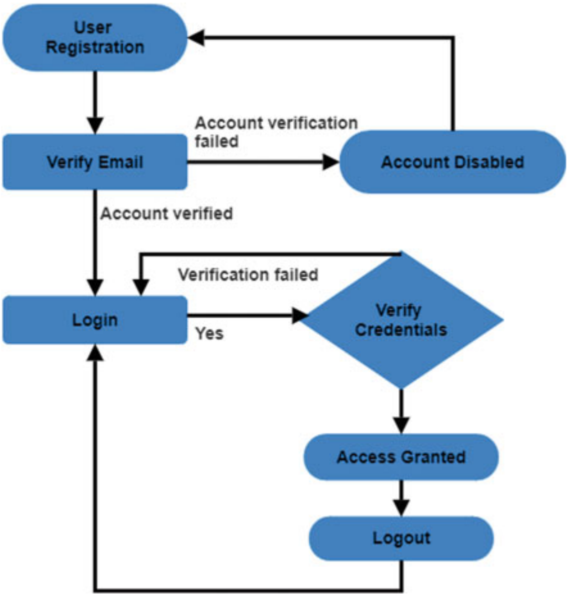


Fig. 1.3 User login process

whenever they are downloaded or requested. Multi-factor authorization is one of the specific login criteria for personnel, which boosts security. To secure sensitive data, their access to patient information, particularly appointment information, is closely controlled. To guarantee secrecy, we have applied the least privilege access concept. Only patient appointments and information like name, ID, phone number, and supporting papers are accessible to doctors. The accountant position, a fourth role, now has unique privileges. In accordance with the least privilege principle, accountants have read-only access to billing information so they may create invoices. In order to accurately specify roles and permissions for each user, we have also integrated entity access management and role-based login procedures. This ensures adequate access control and prevents data mixing or misplacing.

Administrators, who have the greatest level of access and administrative powers, must authenticate using both their passwords and Google Authenticator, each time they log in, adding an additional layer of security. The Google Authenticator will provide them with a 6-digit number that changes every 30 s, reducing the chance of a brute-force assault. The Google Authenticator will be installed on the admin's second device like phone or a separate laptop, ensuring that even if the administrator's system is hacked or stolen, an attacker will still need to enter a code in order to log in. The admin interface offers total control over the system and gives access to both patient and physician data, except their passwords. The only person who has the authority to modify or set the staff's password is the admin. The administrator may

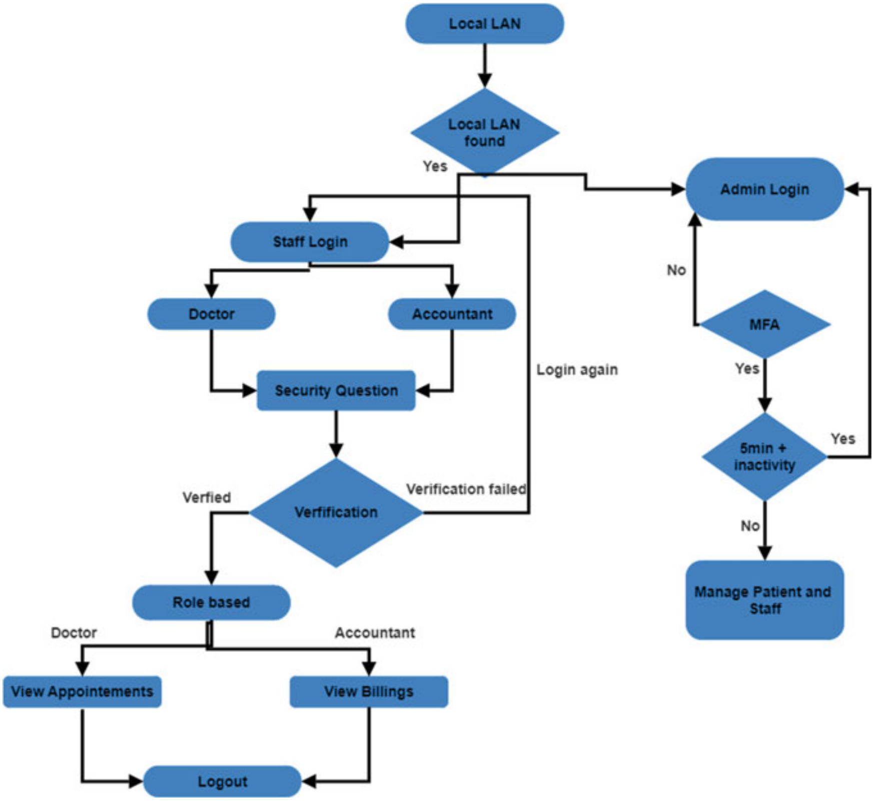


Fig. 1.4 Staff login

respond to a request from the employees to change their password by sending them a link. The admin panel has time-based access restriction to improve overall security, lowering the danger of unwanted access by automatically logging out administrators after a period of inactivity.

The database, which is protected with strong security measures, houses all system data safely. In order to protect stored data's integrity and confidentiality from SQL injection attacks, we have put safety measures in place.

In order to increase security, network-based control has been incorporated in addition to the previously described features. If any other network is found, the system restricts access for staff and administrators and only uses the default hospital LAN network. Patients can still use the system, though, because in this situation their contribution is not seen as being crucial. We have developed micro-segmentation of the database, splitting it into separate sections with specific security methods, to further protect data. This segmentation isolates and safeguards important information within the system, preventing simple assaults on patient data.

Overview of Used Conditions with Help of Pseudo Code

```
Start

User Initiates Access Request
if user is authenticated:
    Proceed to next step
else:
    Prompt user for authentication

Apply Conditional Access Constraints
if user meets required attributes and permissions:
    Proceed to next step
else:
    Deny access

if device is compliant and secure:
    Proceed to next step
else:
    Deny access

if access request originates from a trusted location:
    Proceed to next step
else:
    Deny access

if network is secure and trusted:
    Proceed to next step
else:
    Deny access

if access request is within allowed time frames:
    Proceed to next step
else:
    Deny access

if access request poses high-risk factors:
    Apply additional authentication factors
else:
    Proceed to next step
Grant Access to Authorized Resources
Provide access to the requested healthcare resources

End
```

Results and Discussions

The developed healthcare system's adoption of a Zero Trust strategy has greatly improved data security. This is mostly because conditional access is used and restricted access is implemented, where access is only allowed to those who need to know. The danger of illegal access has also been reduced thanks to the adoption of Identity and Access Management (IAM) and Multi-Factor Authentication (MFA), which has ensured that only authorized and authenticated users have access to certain resources. Following the CIA trinity, we put confidentiality first with access restrictions, uphold data integrity with strong database security, and guarantee availability through resilient infrastructure. A trustworthy and safe healthcare system is guaranteed by our dedication to thorough security.

Although the data security has been greatly improved by our extensive security procedures, we understand that no security system can provide 100% protection. To find and address any vulnerabilities, we proactively upgrade our security processes and regularly monitor them. We are still dedicated to keeping our healthcare system as secure as it can be while also remaining watchful and flexible in the face of changing attack vectors.

Summary and Conclusions

In summary, the adoption of a healthcare system based on conditional access controls and zero trust architecture has enormous potential to improve data security and privacy in the healthcare sector. Organizations may move from a perimeter-based security model to a more resilient and dynamic system that continually validates and approves every person, device, and request by implementing a zero-trust strategy. By using conditional access limits, companies may customize access rights depending on a variety of factors, including user characteristics, device characteristics, location, network circumstances, time of day, and risk levels. The attack surface is greatly reduced, security risks are mitigated, and only authorized users are able to access critical healthcare data thanks to this granular approach to access control. Additionally, it makes it possible for companies to adhere to legal specifications and retain the privacy, accuracy, and accessibility of patient data. But it's crucial to be aware of the possible dangers of conditional access, such as its complexity, effects on user experience, false positives and negatives, DDoS vulnerabilities, insider threats, and difficulties with policy enforcement. To overcome these issues, organizations should periodically evaluate and update their conditional access policies, carry out rigorous risk assessments, install MFA, watch and examine access logs, and keep up with new security technology.

References

1. Boyd D, Crawford K (2012) Critical questions for big data: provocations for a cultural, technological, and scholarly phenomenon. Inf Commun Soc 15(5):662–679
2. Acquisti A (2004) Privacy in electronic commerce and the economics of immediate gratification. In: Proceedings of the 5th ACM conference on Electronic commerce, pp 21–29
3. Kruse CS, Frederick B, Jacobson T, Monticone DK (2017) Cybersecurity in healthcare: a systematic review of modern threats and trends. Technol Health Care: Off J Eur Soc Eng Med 25(1):1–10
4. SingHealth Data Breach (2018) Wikipedia. https://en.wikipedia.org/wiki/2018_SingHealth_data_breach. Last Accessed 2 Oct 2018
5. Zero-trust securing healthcare data. https://colortokens.com/blog/zero-trust-securing-healthcare-data/. Last Accessed 31 May 2023
6. Dumitru I-A (2022) Zero trust security. In: Proceedings of the international conference on cybersecurity and cybercrime (IC3), n. Pag.
7. Rose S, Borchert O, Mitchell S, Connelly S (2019) Zero trust architecture
8. Polato (2021) Zero trust network architecture with John Kindervag. https://www.Paloaltonetwork.com/resources/videos/zero-trust
9. Kavitha M, Damodharan P (2020) Software as a service in cloud computing
10. He Y, Huang D, Chen L, Ni Y, Ma X (2022) A survey on zero trust architecture: challenges and future trends. Wirel Commun Mob Comput 2022:n. Pag.
11. Sandhu RS, Coyne EJ, Feinstein HL, Youman CE (1996) Role-based access control models. Computer 29:38–47
12. Sheikh NI, Pawar MD, Lawrence V (2021) Zero trust using network micro segmentation. In: IEEE INFOCOM 2021-IEEE conference on computer communications workshops (INFOCOM WKSHPS), pp 1–6
13. Psarra E, Patiniotakis I, Verginadis Y, Apostolou D, Mentzas G (2020) Securing access to healthcare data with context-aware policies. In: 2020 11th International conference on information, intelligence, systems and applications (IISA), pp 1–6
14. Shashanka M, Shen M-Y, Wang J (2016) User and entity behavior analytics for enterprise security. In: 2016 IEEE international conference on big data (Big data), pp 1867–1874
15. Songpon T, Tetsutaro U, Atsuo I (2021) Migrating to zero trust architecture: reviews and challenges. Secur Commun Netw 2021:1–10. https://doi.org/10.1155/2021/9947347
16. Yan S, Kwon KJ (2014) Massive data theft hits 40% of South Koreans. CNN Money. http://money.cnn.com/2014/01/21/technology/korea-data-hack/. Last Accessed 21 Jan 2014
17. Rajna G (2018) Equifax data breach. viXra, n. Pag.
18. Bukaty P (2019) The California Consumer Privacy Act (CCPA)
19. Gregory M (2023) Why healthcare organizations should begin their zero-trust implementations with identity. Technology Solutions That Drive Healthcare
20. Lawson NA, Orr JM, Klar DS (2003) The HIPAA privacy rule: an overview of compliance initiatives and requirements. The privacy rule contains a maze of mandates and exceptions requiring that entities covered by HIPAA need the best of health care counsel. Def Couns J 70:127
21. Cybersecurity and Infrastructure Security Agency (CISA) insights. Defending Against Malicious Cyber Activity Originating from Tor. https://www.cisa.gov/publication/insights-defending-against-malicious-cyber-activity-originating-tor
22. Elrod J, Abernathy P (1995) Healthcare information systems: opportunities and challenges. Commun ACM 38(8):49–51
23. Terry MB (2018) Protecting privacy in the era of digital health. PLOS Med 15
24. Devi T, Kamatchi SB, Deepa N (2023) Enhancing the security for healthcare data using blockchain technology. In: 2023 International conference on computer communication and informatics (ICCCI), pp 1–7

25. Chen B, Qiao S, Zhao J, Liu D, Shi X, Lyu M, Chen H, Lu H, Zhai Y (2021) A security awareness and protection system for 5G smart healthcare based on zero-trust architecture. IEEE Internet Things J 8:10248–10263
26. Tyler D, Viana T (2021) Trust no one? A framework for assisting healthcare organisations in transitioning to a zero-trust network architecture. Appl Sci 11(16):7499
27. Ali B, Gregory MA, Li S (2021) Uplifting healthcare cyber resilience with a multi-access edge computing zero-trust security model. In: 2021 31st International telecommunication networks and applications conference (ITNAC), pp 192–197
28. Singh J, Ghai K (2021) Security and privacy mechanisms for the new generation healthcare applications using blockchain technology. In: 2021 9th International conference on reliability, Infocom technologies and optimization (Trends and future directions) (ICRITO), pp 1–6
29. Wang Z, Yu X, Xue P, Qu Y, Ju L (2023) Research on medical security system based on zero trust. Sensors 23(7):3774. https://doi.org/10.3390/s23073774
30. Huang Y, Tang (2016) Securing healthcare IT infrastructure with blockchain and modern cryptography
31. Kindervag J (2010) Build security into your network's DNA: the zero trust network architecture. Forrester Research Inc.
32. Microsoft (n.d.). Conditional access in azure active directory. https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/overview
33. Django (n.d.). https://www.djangoproject.com/
34. Django documentation: cross-site scripting (XSS) protection (n.d.). https://docs.djangoproject.com/topics/security/#cross-site-scripting-xss-protection

Chapter 2
Security Issues and Challenges in Deploying a CPS Using WSN

Mohan Kumar Dehury, Bhabendu Kumar Mohanta, and Sangay Chedup

Abstract The domains of wireless sensor networks and Mobile Ad Hoc Networks (MANETs) have seen a lot of research activity during the last two decades (WSN). The cyber-physical system (CPS) has lately come to light as a viable avenue for enhancing interactions between the real and virtual worlds. In this chapter, first have presented some WSN research directions, covering networking, deployment, and coverage difficulties. Then, we examined some CPS platforms and systems that have recently been created, including applications for social networking, gaming, health care, navigation, and rescue. Through these evaluations, we have tried to illustrate how CPS applications use the tangible data gathered by WSNs to connect the physical and virtual worlds and highlight significant research issues pertaining to CPS designs. Also, in a distributed controlled environment, the incorporation of wireless sensor networks into CPSs offers enormous benefits. However, due to its layered structure and wireless sensor network, the CPS is vulnerable to both internal and external assaults. These dangers could result in network losses in terms of money or structure. In order to address such security issues and challenges, we have organized this chapter to classify layer-wise assaults which are generated both from inside or outside on WSNs and CPSs. We have also listed some already known methods for detecting security and defense mechanisms that are used to counter threats in the case of WSNs and CPSs. In the end of this chapter, we have presented a comparative study of the methods used to protect WSNs and CPSs from such threats.

Keywords CPS · WSN · Security · Privacy · IoT · Layer-to-layer attacks

M. K. Dehury (✉)
Amity Institute of Information Technology, Amity University Jharkhand, Ranchi, India
e-mail: mohankdehury@gmail.com

B. K. Mohanta
Department of CSE, Koneru Lakshmaiah Education Foundation, Vijayawada, AP, India

S. Chedup
Department of ECE, Jigme Namgyel Engineering College, Dewathang, Bhutan
e-mail: sangaychedup@jnec.edu.bt

D. Gountia et al. (eds.), *Information Technology Security*, Springer Tracts in Electrical and Electronics Engineering, https://doi.org/10.1007/978-981-97-0407-1_2

Introduction

Cyber-Physical Systems (CPSs) enabled by Wireless Sensor Networks (WSNs) have gained significant attention due to their ability to integrate physical and computational elements, enabling real-time data collection and analysis. CPS comprises physical entities, such as sensors, actuators, and control systems, along with computational components, including communication networks and data processing algorithms. WSN plays a crucial role in CPS by facilitating the collection of sensory data, enabling interconnectivity, and supporting seamless communication between physical and virtual components.

The architecture of CPS enabled by WSN typically consists of four layers: the physical layer, the sensor layer, the network layer, and the application layer [1]. The physical layer encompasses physical entities such as sensors, actuators, and control systems. The sensor layer includes the wireless sensor nodes responsible for data collection. The network layer handles the communication between sensor nodes and facilitates data transmission. Lastly, the application layer encompasses the software and algorithms that process and analyze the collected data to derive meaningful insights. CPS enabled by WSN find applications across various domains, including health care, smart cities, environmental monitoring, transportation, and industrial automation. In health care, WSN facilitates continuous monitoring of patients' vital signs, enabling early detection of abnormalities and timely medical interventions. In smart cities, CPS–WSN integration allows for efficient management of resources, such as energy, water, and waste, enhancing sustainability and improving quality of life. Environmental monitoring leverages WSN to gather data on air quality, water quality, and weather conditions, aiding in environmental protection and disaster management [2]. In transportation, CPS–WSN integration enables traffic monitoring, intelligent transportation systems, optimizing production processes, and enhancing operational efficiency.

Security plays a crucial role in CPS enabled by WSN due to the critical nature of the data and operations involved. CPS–WSN systems integrate physical and computational components, creating a highly interconnected environment. The collected data from WSN is used to make critical decisions and control physical processes. Therefore, ensuring the security of CPS–WSN systems is paramount to safeguard against unauthorized access, data manipulation, and system compromise. One key aspect of security in CPS–WSN is protecting the integrity and confidentiality of the data transmitted and processed. Unauthorized access to sensitive data can lead to privacy breaches and compromise the system's reliability. Implementing encryption techniques and access control mechanisms can help prevent unauthorized access and protect data integrity. Another significant security concern is the prevention and detection of cyber-attacks. Malicious entities may attempt to disrupt the system's operation through Denial-of-Service (DoS) attacks, node compromise, or data spoofing. Implementing intrusion detection and prevention mechanisms can aid in identifying and mitigating such attacks, ensuring the system's continuous operation and integrity.

Furthermore, the physical components of CPS–WSN systems can be vulnerable to physical attacks, such as tampering with sensors or injecting false signals. Physical security measures, including tamper-proofing sensors and physical access restrictions, are essential to protect the system's integrity and prevent unauthorized physical manipulation. The significance of security in CPS–WSN extends beyond data protection. It also encompasses the system's reliability, safety, and resilience. A compromised CPS–WSN system can have severe consequences, leading to physical damage, financial losses, or even endangering human lives.

This chapter is organized as follows. In Sect. 2.2, we present different CPS platforms that enable the integration of physical and computational components. In Sect. 2.3, we present security concerns related to CPS enabled by the use of WSN. Different types of external and internal attacks on such CPSs are discussed in Sect. 2.4. In Sect. 2.5, we present some future directions and research challenges in CPS systems. And in Sect. 2.6, we present the conclusion of this chapter.

CPS Platforms

There have been several recent platforms developed for CPS that enable the integration of physical and computational components. Here are some notable CPS platforms:

ROSA: ROSA (Robot Operating System for Automation) is a platform specifically designed for industrial automation of CPS. It provides a framework for developing and deploying robotic systems in industrial environments, integrating sensors, actuators, and control systems.

SmartThings: SmartThings is a popular platform for building smart home CPS. It allows users to connect and control various devices, such as lights, thermostats, and security systems, through a unified interface. SmartThings also supports interoperability among different manufacturers' devices.

Azure IoT: Azure IoT is a comprehensive platform by Microsoft for developing and managing CPS applications. It offers a wide range of services, including device management, data ingestion and analytics, and integration with cloud services. Azure IoT enables the development of scalable and secure CPS solutions.

LabVIEW: It is a development platform widely used in scientific and industrial CPS applications. It provides a graphical programming environment for designing and deploying measurement and control systems, allowing integration with sensors, actuators, and data analysis tools.

IBM Watson IoT: It is an IoT platform that extends to CPS applications. It offers device management, data analytics, and cognitive capabilities. The platform enables the development of intelligent CPS solutions, leveraging machine learning and AI technologies.

ThingWorx: It is an IoT platform focused on industrial CPS. It provides tools for connecting devices, managing data, and developing applications. ThingWorx supports real-time monitoring, predictive analytics, and remote control of industrial processes.

These platforms provide developers with the necessary tools and frameworks to build, deploy, and manage CPS applications. They offer various features and capabilities, allowing for customization and integration with different sensors, actuators, and control systems. The choice of platform depends on the specific requirements and domain of the CPS application. Also, data requirements in CPS platforms are essential for enabling the effective functioning and decision-making processes of CPSs. The following points present the significance of data requirements and their role in CPS platforms.

- CPS platforms rely on accurate and timely data to monitor, analyze, and control physical processes. The data requirements encompass several key aspects. Firstly, data collection involves identifying the types of data needed, such as sensor readings, environmental parameters, or machine states. Determining the frequency and granularity of data collection is crucial to capture relevant information for real-time monitoring and analysis.
- Secondly, data transmission requirements focus on the efficient and reliable transfer of data from sensors to the CPS platform. This includes considerations like data rate, latency, and network protocols. CPS platforms must ensure that data is transmitted securely and in a timely manner to support timely decision-making and control actions.
- Thirdly, data storage and management requirements address the storage capacity, scalability, and data retention policies of CPS platforms. Efficient storage and management systems are necessary to handle the large volumes of data generated by CPS applications. Data storage should also comply with privacy and regulatory requirements to protect sensitive information.
- Lastly, data processing and analytics requirements involve techniques for extracting meaningful insights from the collected data. This includes data pre-processing, filtering, aggregation, and analysis of algorithms. Real-time and predictive analytics enable CPS platforms to detect anomalies, make informed decisions, and optimize system performance.

Meeting these data requirements is crucial for the success of CPS platforms. Accurate and timely data collection, transmission, storage, and processing contribute to the reliability, efficiency, and effectiveness of CPS applications. By understanding the specific data requirements of CPS platforms, developers can design robust data architectures and implement appropriate data handling techniques to support the complex and dynamic nature of CPSs.

Security Concerns

From the perspective presented in [3], CPS and WSN both fall under the IoT's general heading. Thus, the majority of problems and worries, particularly those related to security, are comparable. The primary distinction is that the CPS integrates WSN and M2M technologies in a larger-scale system. Multiple clusters and WSN hierarchies could exist within a single CPS. Multiple clusters and WSN hierarchies could exist within a single CPS. All authors in [4, 5] highlighted how vulnerable WSNs are to different types of assaults. The main cause for concern is the lack of resources, which makes it impossible to integrate advanced security measures with the memory and processor levels that are now accessible. Authenticity, confidentiality, integrity, authorization, secrecy, survivability, scalability, efficiency, availability, and confidentiality are further security challenges [6, 7]. In a CPS, security concerns expand beyond WSN-only elements to include actuators, calculations for making decisions, to maintain flow of information in bi-directional way, and also to maintain communication among different types of devices. Major security-related issues in CPSs have been identified by authors in [8–10]. These issues include availability, confidentiality, validation, authenticity, reliability, integrity, robustness, and trustworthiness. According to authors in [8, 11, 12], the aforementioned problems can lead to a number of security threats, including denial of service, routing attacks, man-in-the-middle, network-based intrusion, malware, eavesdropping, resonance attacks, compromised key attacks, jamming attacks, and integrity attacks. In Sect. 2.4, it will be discussed how WSNs and CPSs differ in terms of layer-to-layer attacks.

Attacks at Different Layers of CPS Using WSN

CPS using WSN is organized using layered architecture; this network's layered architecture is more vulnerable to flaws and could result in significant losses and harm from a variety of attacks, as was covered in the preceding sections. The layers of CPS using WSN will be explained in this section, along with their varied detection strategies for identifying attacks and offering active or passive defensive mechanisms for network prevention. The attacks on these layers are classified as attacks from within the layers and attacks from outside the layers.

Physical Layer: Attacks from Outside

At the physical layer of CPS and WSN, a variety of attacks are feasible, including eavesdropping, traffic analysis attacks, device tempering, frequency jamming, Sybil, and path-based DoS attacks. Table 2.1 shows the impact of an attack on the physical layer of the WSN and CPS together with their defenses. Traffic analysis attacks in

Table 2.1 Attacks from outside on physical layer

Attacks	Effects	WSN		CPS	
		Methods of detection	Methods of defense	Methods of detection	Methods of defense
Traffic analysis	Low packet throughput, excessive packet collision, and altered traffic [13]	Analysis of typical rate monitoring and temporal correlation attacks using statistics [13]	Fractal propagation, a random walk, and a multi-parent routing method	Statistical analysis [4]	Industrial WSNs use Modbus TCP to analyze model-based data traffic [4]
Monitoring the communication line closely during Eavesdropping	decreased data privacy, obtaining important information, exposure to adversaries further attacks [14, 15]	Statistical analysis, Misbehavior detection techniques [14]	Advanced encryption, peripatetic security solution, scattered processing, access constraint, systematic access control, and strong encryption technology [14, 15]	Behavior, behavior-specification, knowledge	Shin technique [16]
Jamming: cause interference by introducing intensive radio signals of identical frequency	Excessive energy, disrupt communication, occupy entire bandwidth, corrupt data packets, deceive network's defensive mechanisms, resource exhaustion [14, 15, 17]	Statistical information, threshold for channel utility degradation, background noise [14, 15, 17]	Secure encryption, CRC check, Lower duty cycle, higher broadcast power, hybrid FHSS/DSSS, ultra-wideband, change of antenna polarization, use of directional transmission, message prioritization, blacklist [14, 15, 17]	Statistical information	Spread spectrum techniques, FHSS/DSSS, encryption [18]
Device tampering: direct physical capture of sensor. Attack at the base station	Captivate & destroy captured node to clone it to annex WSN using software liabilities [15, 17, 19]	Internodal isolation, monitoring, key management, misbehavior [15, 17, 19]	Hardware and software alertness, disguising sensors, restriction access, data reliability and privacy, node detection [15, 17, 19]	Internodal isolation, monitoring, key management, misbehavior (Networks 2016)	Host identity protocol (HIP): a new trust model (Networks 2016)
Sybil attack: system sabotaged by counterfeiting characters	Causes network inaccessibility [15, 17]	Low overhead and delay of signals [15, 17]	Physical shield of nodes [15, 17]	Low overhead and delay of signals (Newsome et al. 2004)	Radio Resource testing, random key redistribution, registration, position verification, code attestation (Newsome et al. 2004)
Path-based DoS: typical mixed jamming like attacks	Exhaustion of nodes' battery, network disturbance, deceitful rejection of nodes [15]	Misbehavior [15]	Gray-listing, redundancy, anti-attack, and acknowledgment validation [15]	Behavior, specific behaviors, understanding of message type and packet arrival rate [16]	System for detecting intrusions [16]

Table 2.2 Attacks from outside on data link layer

Attacks	Effects	WSN		CPS	
		Methods of detection	Methods of defense	Methods of detection	Methods of defense
Collision	interferences, control or data packet corruption, discard the little packets, overuse of energy, impact on cost [15, 19]	Misbehavior	Codes for fixing errors, time variation [15, 19]	Statistical analysis [20]	Error-correcting codes [20]
Resource exhaustion	Resources exhaustion, compromise availability [15, 21]	Misbehavior [15, 21]	limiting MAC rate, backs off at random intervals, TDM, Link response rate regulation and ID safety [15, 21]	Misbehavior [20]	IED relay settings are monitored in real-time while correlating event records is done [22], Rate restrictions [20]
Traffic engineering	Aggressive channel utilization, poor signal quality, ineffective WSNs, traffic fabrication, and purposefully induced contention [17]	Misbehavior [14, 17]	Link Layer encryption, collision protection, and traffic analysis. MAC request volume [14, 17]	Behavior, behavior-specification, knowledge [22]	Utilization of SNMP and other network management protocols [22]
Eavesdropping	Remove crucial information without reducing privacy protection [23]	Statistical analysis	Possession of accessl processing that is distributed and extreme encryption [23]	Behavior, behavior-specification, knowledge [16]	Shin approach [16]
Killing the cluster leader and directing nodes in the incorrect direction are examples of impersonation.	Routing table disruptions and sensor kills. a crowded network. a network leak production of false data. overuse of resources. leak important data and encryption keys [21, 23]	Misconduct, dishonesty, deceptive routing, and collision [21, 23]	Secure identity, secure routing, and small packet frames with a limited MAC rate [21, 23]	Misconduct and false identity. Router fraud and collision [24]	Employing symmetric keys, a keying scheme that offers both forward and backward key secrecy [24]
Wormholes	Untrue routes, Overused routing race circumstances. a topological change in the network. Protocol for path detection has failed. destruction of packets [23, 25]	Incorrect routing data. Methods for Packet Restrictions [23, 25]	Border control protocol, DAWWSEN protocol, graphic position system, and multidimensional scaling method. Ultrasound, global clock sync. Link layer encryption with authentication. global shared key [23, 25]		Wormhole attack and protection technique for CPSs using the Gianluca model [26]
Unfairness	Efficiency loss and channel access demand. Capacity restriction for channel access [15, 23]	Misbehavior [23]	Small frames usage [23]	Statistical analysis [20]	Small frames usage [20]
De-synchronization	Disruptive communication drain of resources [15, 17, 23]	De-established connections [15, 17, 23]	Strong authentication mechanisms Time synchronization [15, 17, 23]	De-established connections between the nodes [20]	Strong authentication [20]
DoS attack	Prevent hosts from accessing the local network [15, 23]	destroys network connections [15, 23]	All of the above	Knowledge, behavior, and behavior-specification [22]	To enable encrypted and authenticated communication with the master, use a SCADA Security Device (SSD) [22]

Table 2.3 Attacks from within the data link layer

Attacks	Effects	WSN		CPS	
		Methods of detection	Methods of defense	Methods of detection	Methods of defense
Recognizing spoofing	Falsification of data, packet loss, steering loops and adjusting their duration, dissemination of error messages, modification and replay of data monitored data [15, 23]	Behaving inappropriately	Use a new route to validate, encrypt, and secure link layer and global shared key approaches [15, 23, 27]	Receiving data in a time-based manner Park et al. 2010	Identity verification using IDS will make the essential negotiating process stronger [4]

Table 2.4 Attacks from outside on network layer

Attacks	Effects	WSN		CPS	
		Methods of detection	Methods of defense	Methods of detection	Methods of defense
Eavesdropping	Extracting crucial information and exposing privacy [23]	Statistical procedures [23]	Possession of access. Processing that is distributed; robust encryption [23, 25]	Knowledge, behavior, and behavior-specification [28]	Mechanism distributes traffic across various pathways using the SDN facilities to help SCADA networks defend against illegal flow interception [28]
Subversion node	Dishonesty, resource exploitation [25]	No ACK from the compromised sensor [25]	Masking sensors, correct procedures, Access restriction and data privacy [25]	Physical anomalies [29]	An IP traced back technique is used to track down and neutralize compromised nodes by tracking packet behavior [29]
Flooding	Exhaustion of resources and decreased WSN traffic [14, 19]	The entire network slows [14, 19]	Dual-way authentication [14, 19]	Message formats	Communication channels, signatures/verifications, key management, and TCP/IP defenses are all slowed down by the use of encryption techniques
Spoofing	Fragmented networks, excessive resource utilization, network lifetime reduction and routing data shedding [14, 19]	Passively static MAC, secure ARP protocol Kernel-based patches [14, 19]	MAC encryption utilizing a different path when transmitting messages again [14, 19]	Knowledge, behavior, and behavior-specification	Premaratne technique (IDS), layer-by-layer operating
Wormholes	Distorted false pathways, routing race circumstances that are overused network topology change procedure for path detection failing, destruction of packets [14, 19]	Packet restrains and false routing information [14, 19]	BCP, GPS, ultrasound, a global clock, directional antennae, a multidimensional scaling algorithm, the DAWWSEN protocol, a global shared key [14, 19]	Message formats [18]	By implementing an isolation strategy of malicious nodes utilizing a particular threshold, these assaults can be lessened [4]
DoS attacks	Overload a target network with data until it breaks [14, 19]	Packet restrains and false routing information [14, 19]	BCP, GPS, ultrasound, a global clock, directional antennae, a multidimensional scaling algorithm, the DAWWSEN protocol, a global shared key [14, 19]	Knowledge, behavior, and behavior-specification [30]	Whitelists are automatically generated by the IndusCAP-Gate system for traffic analysis, and repeated filtering is carried out based on those whitelists to prevent illegal access from outside networks [30]

Table 2.5 Attacks from within in network layer

Attacks	Effects	WSN		CPS	
		Methods of detection	Methods of defense	Methods of detection	Methods of defense
Misdirection	Faulty routing tables, excessive resource use [19]	Predicting throughput and delay [19]	A system for routing in hierarchy [19]	Message formats	Security measures to measure covert attacks
Rushing	Ignoring sincere requests not being able to find any useful routes [27]	Not being able to find routes with more than two hops [27]	Rushing attack prevention (RAP) [27]	Message formats	An effective certificate-less signature system called McCLS is based on the random oracle model's bilinear Diffie–Hellman assumption
Homing	Attacking actively key resources extrapolate crucial network data, privacy of data at risk [27]	Statistical methods [27]	Encryption [27]	Message formats	To specifically permit traffic from the plant data historian to be exchanged with the SCADA or control system over specified ports at specified rates, a firewall rule or access control rule should be developed
Selective forwarding	Network disruptions from upcoming assaults [6, 27]	Statistical methods [6, 27]	Source routing and network monitoring [6, 27]	Message formats [4]	Selecting the next hop dynamically from a group of candidates allows for mitigation, presuming that none of the nodes in this group are compromised [4]
Sybil attack	Causes data integrity and accessibility issues on the network [6, 25]	To improve its own reputation, disregard suspicious nodes by matching the trust ranking criteria to fictitious nodes that conform to them. Additionally, doing this will assist others' reputations suffer [6]	Physical shield of nodes [6, 25]	Maintaining little signal delay and overhead [4]	Enhance the crucial bargaining process. Additionally, verify the nodes' identities [4]
Black hole	Suppressing broadcast along a spurious shortest way, using the hub as a data packet black hole to disrupt the network routing table,	Disregarding distrusted nodes by matching rules of trust ranking [6]	Sensor network automatic intrusion detection system (SNAIDS), Multipath Routing, Decentralized IDS	Drop in traffic	An effective certificate-less signature system called McCLS is based on the random oracle model's bilinear Diffie–Hellman assumption

(continued)

Table 2.5 (continued)

Attacks	Effects	WSN		CPS	
		Methods of detection	Methods of defense	Methods of detection	Methods of defense
Spoofing	break in the network, resource overuse, reduction in network lifetime Loss of route information [25, 27]	ARP protocol security fixes, kernel-based patches [25, 27]	Re-sending the message using encryption and MAC multi-path authentication [25, 27]	Knowledge, behavior, and behavior-specification [10]	Access control whitelists, protocol-based whitelists, and behavior-based rules are all used by a SCADA-specific IDS to detect external hostile assaults and internal accidental abuse [10]
ACK spoofing	Packet fraud using a targeted forward assault [25, 27]	Using kernel-based modifications, the secure ARP protocol can be detected passively [25, 27]	MAC-ARP header sniffer, traffic filter, spoof detector, and spoof alert [25, 27]	Knowledge, behavior, and behavior-specification [31]	Smart grid spoofing defense technique using cross-layer defense [31]
Hello flood	Sending erroneous HELLO packets from neighbors to the entire network [14]	Dual-direction link [14]	Client ambiguity [14]	Dynamic-static jamming [18, 28]	To prevent and minimize data flooding, control devices should be provided rate limiting commands [18]
Internet smurfing	Computer of the victim is completely stud down [14, 27]	Overpopulated network link with unnecessary data [14, 27]	Put the attacker's node to sleep and disable IP broadcasting at the router [14, 27]	Dynamic-static jamming [30]	Filtering of IP packets and network intrusion detection [30]
Flooding	running out of resources delayed accessibility decreased WSN traffic [27]	The entire network slows [14]	Dual-way authentication [14]	Dynamic-static jamming [30]	Filtering of IP packets and network intrusion detection [30]
Gray hole	a network interruption that went undiscovered [14, 27]	Calculated check time for neighbor dynamic routing using RREQ and RNPS [14, 27]	Checkpoint-based multipath routing and multi-hop acknowledgment [14, 27]	Drop only a few packets as opposed to all of them. Behavior-specification	Barbosa technique, IDS
Gratuitous detour	Constrained resources misdirection, network breakout, looping, irregular routes [23]	Performance of the network declining [23]	Adopt validation techniques, pair-wise authentication, network layer authentication, and a central certificate authority [23]	Dynamic-static jamming	Prevailing pruning technique. The nodes in both lists periodically shake hands in order to remove any nodes that are not present

WSN are divided into two groups by authors in [13]: rate monitoring attack and time correlation attack. The quantity of packets that the nodes send to the attacker is the focus of the rate monitoring attack. The time correlation attack looks at the sending times of nearby nodes in correlation. Furthermore, a threat model is added to simulate rate monitoring attacks using various anti-traffic analysis techniques, while the effectiveness of time correlation attacks is reduced by adding random false pathways. There are some important schemes like random walk (RW), fractal propagation with different forking probabilities (DEFP), multi-parent routing scheme (MPR), enforced fractal propagation (EFP), and differential fractal propagation (DFP) which have a unique capacity to cover up the base station's position. These schemes which form the basis for methods are described in [13]. However, none of these methods has been practically applied to any sensor network. In contrast, authors in [4] found that Zigbee Pro is a more practical answer to the traffic analysis attack since it chooses a random way for communication in SCADA networks.

As a proposed remedy for the eavesdropping in WSNs, Jadidoleslamy [23] and Virmani et al. [14] offered strong encryption technology, systematic access control, and enhanced encryption. In an experiment using one-hop clustering for SCADA applications a protocol was designed using the current WSN algorithms by Shin et al. [16]. The protocol was created to help prevent the use of false routing information and sinkhole attacks. Another danger is interference in powerful radio transmissions, which has been researched by several authors [14, 15, 17]. The authors suggested a number of countermeasures to minimize the effect of jamming, including directional transmission, safe encryption, CRC check, increased broadcast power, hybrid FHSS/DSSS, lower duty cycle, and ultra-wideband. Mo et al. [18] recommend using several spread spectrum approaches to protect against jamming attempts in CPS. The suggested approach, however, can only evaluate the replay assault through simulation, whereas that paper just provides an overview of the strategies that are now accessible. In [15], the authors evaluated the network deterioration and harm to the sensor nodes' functionality caused by P-DoS attacks. Additionally, they looked into how P-DoS attacks could be mitigated by leveraging network packet redundancy, acknowledgment validation, and the gray-listing of infected nodes. In order to fight against P-DoS attacks on CPSs, Shin et al. [16] calculated the packet receiving rate as well as a preset packet arrival threshold.

Data Link Layer: Attacks from both Outside and Inside

Similar to the application layer of CPS and WSN, the data connection layer is likewise vulnerable to a variety of internal and external assaults, including DoS attacks, resource depletion, traffic manipulation, eavesdropping, impersonation, wormholes, unfairness, and resource exhaustion. Tables 2.2 and 2.3 present these attacks on data link layer. The threats and security protocols of wireless sensor protocols are reviewed by authors in [19], along with the fundamental information and security needs for WSNs. According to the study, error correction codes and transmitting numerous copies of the same signal at various times (time diversity) can be used to reduce interference and packet data corruption, respectively [19]. The use of error correction codes for faulty data in CPSs was also recommended by Ali et al. in [20]. The paper gives a theoretical overview of network embedded systems' security aspects. Researchers in [15, 21] examined sensor node misbehavior brought on by persistent data collisions. Multiple approaches, including data rate restriction or limitation, switching from pulse-code modulation to time division multiplexing, and identity protection of sensor network nodes, can be used to manage fatigued nodes brought on by frequent collisions [21]. The study is set up to offer countermeasures, but there are certain restrictions because the suggested approach hasn't been tried out in practice or on a testing ground. Aniket ensured in [22] that a real-time monitoring strategy with event log correlation aids in resolving the issues brought on by resource depletion. The study's findings can be accepted because they cover the development and use of CPS device security. In order to combat the threat of false data on networks, the authors also looked at the consequences of impersonation in WSNs' data link layer and used the sheltered routing approach with short packet frames.

In their model, authors in [24] used symmetric keys for back-and-forth communication. The proposed model can be partially or fully incorporated into any SCADA network and is compatible with the current SCADA network. In order to defend against the wormhole attack in WSNs, some authors studied the multidimensional DAWWSEN protocol [15]. In [26], the authors investigated a framework to defend against wormhole attacks on CPSs. Through simulation testing, the framework was able to determine the types and grades of attacks that could be defended against. Authors in [15] investigated the use of strong authentication and time synchronization mechanisms at the data connection layer as a defense strategy against denial of service (DoS) and de-synchronization assaults. Strong authentication can be used to counter de-synchronization attacks, according to authors in [20]. Although the data link layer is also susceptible to internal acknowledgment spoofing attacks, Aniket et al. created a SCADA security device to enable encrypted and authentic communication in CPSs [22]. This was made clear by studies that found that using an encrypted link layer with global shared keys helped to mitigate the consequences of acknowledgment spoofing [15, 27]. Alcaraz and Lopez examined how the key negotiation procedure combined with node identity checking can increase acknowledgment spoofing [4]. In [25], the authors explored the DoS assault on data that could be thwarted by physical node shielding and routine key changes.

Network Layer: Attacks from Outside

Attacks like node subversion, eavesdropping, spoofing, flooding, DoS, and wormhole attacks are all capable of exploiting the network layer of CPS and WSN. Strong encryption methods and distributed processing have been identified by authors in [23, 25] as solutions to the privacy problem in WSNs. Table 2.4 presents the comparison of methods of detection and methods of defense for attacks from outside on network layer. Data privacy can be obtained by distributing traffic over a number of channels, according to authors in [28]. This work is based on an experimental investigation that takes into account the current smart grid model for securing personal data by using a mechanism called anti-eavesdropping in CSPs. In a WSN, authors in [25] determined the remedy for a node subversion assault. According to the study, node subversion assaults can be prevented by using appropriate protocols, data privacy, and controlled access on nodes. Mcevoy and Wolthusen described a protocol that can be used to monitor packet activity and combat compromised nodes in [29]. A probabilistic approach that could successfully defend against node-based attacks was also suggested by the study. The model was mathematically assessed and is considered to be a significant step in the direction of a low computing cost model.

The authors in [19] investigated the resource depletion caused by flooding in WSNs, which has the detrimental impact of completely slowing down the network. The scientists found that a system based on bidirectional packet authentication can be crucial to fend off the detrimental network slowdown. On the other hand, Javier Lopez and Wolthusen in [32] found that the CPS can combat the negative impacts of flooding by using encryption techniques, key management mechanisms, and delays in communication channels. Also, Virmani et al. in [14] stated that for managing the spoofing in network layer of a WSN, passively static MAC and secure ARP will be useful. A cross-layer mechanism was found by Zhang et al. [31] to be a more effective method of preventing spoofing in CPSs. In order to mitigate the impact of wormholes, authors in [19] investigated a multidimensional scaling method in conjunction with a global shared key. In their study in [4], the authors found that a wormhole attack may be thwarted by isolating hostile nodes. With regard to the security flaws, dangers, and available security methods in these security standards, the study presented an examination of Zigbee PRO, WirelessHART, and ISA100.11a. In-depth security guidelines for attacks on crucial CPSs were also presented in the study. Indus CAP-Gate is a system that Kang et al. in [30] suggested to study application layer assaults and network layer attacks and block harmful activities when accessed from the external network. Due to SCADA's restricted access to the external network and the proposed intrusion detection/prevention system's compatibility requirements, the system will be vulnerable to external threats. Therefore, in order to improve SCADA security at the network layer, researchers must address the problem with security techniques.

Network Layer: Attacks from Inside

Numerous attacks can be launched against WSNs and CPSs at the network layer. This includes attacks like selective forwarding attack, misdirection of nodes, Sybil attack, sinkhole attack, black hole attack, spoofing, acknowledgment spoofing, hello flood, smurfing, flooding, gray hole, and gratuitous detour. Table 2.5 shows the impact of an assault on the WSN and CPS network layer along with their defenses. Egress filtering, authentication, and hierarchical monitoring mechanisms can stop a misdirection attack in a CPS, according to authors in [19]. Security metrics were examined by [32] to quantify the misdirection attack. In order to examine the attack for both single-path routing and multipath routing, the paper gives the effective algorithms with their numerical output. The findings demonstrated that if attack costs are increased by two or three times, then the intensity of attacks gets decreased by 50% [20]. Numerical analysis was used to evaluate and validate the mathematical model, although it has not yet been used in practice. In their study, Singh et al. [27] examined network rushing, homing, and selective forwarding assaults and provided countermeasures for these attacks, including attack prevention techniques, encryption, and network monitoring with source routing. The paper gives a general overview of the many methods for reducing the impact of cybersecurity assaults on WSN layers, but it does not address CPSs. In order to defend against the internal attack of rushing on the network layer of WSNs, [14] offer a certificate-less signature technique. The study offered a theoretical examination of another CLS protocol, but it did not evaluate it using the same criteria as the McCLS method. In order to prevent the SCADA network from being affected by homing attacks, [19] specified an access control rule. The study is founded on the technical suggestions made for the US Department of Energy. The study placed a strong emphasis on safety and technical standards for SCADA system installations in energy production zones.

According to authors in [4], it is possible to mitigate selective forwarding by dynamically choosing nodes from the collection of uncompromised nodes. In their study of the Sybil and sinkhole attacks on WSNs in [30], the authors found a trust-based strategy to mitigate these assaults' effects. According to the study, an attack might either be mitigated by trust or it could really break the trust/reputation system. The paper offers a thorough analysis of the trust mechanisms that are available for attacks and defenses. The study did not invent any new methods, thus that is not what made it special. Authors in [4] found that the Sybil attack and sinkhole assault can be reduced in a CPS by strengthening the key negotiation process and implementing an isolation policy for hostile nodes, respectively. The paper presents an overview of the security measures that are currently in place for very intensive cybersecurity; however, it is not sufficiently original to uncover any new security mechanisms.

According to authors in [30], spoofing in CPSs can be prevented using either a protocol-based whitelist or a behavior-based whitelist. The paper suggested a SCADA-specific intrusion detection system to discriminate between internal and external attacks using a protocol that can identify compromised nodes' harmful activities. The study is unique in that it was modeled using a test platform designed specifically for SCADA. As opposed to Kaur and Singh, who noted that ACK spoofing can be stopped by sniffing the ARP MAC, ARP header, and a spoof detector or spoof filter in [25]. A cross-layer defense mechanism was discovered by [28] to stop spoofing in the smart grid system. The study proposed a cross-layer defense mechanism that is distinctive in that it offers a novel approach to the security of CPSs and is also special in that it has been simulated for multiple CPS threats. In the network layer of a WSN, Virmani et al. studied an identity verification protocol as one of the defenses against a hello flood attack [14]. According to [25], the usage of rate-limiting orders through control devices in CPSs can be used to prevent data limiting and device flooding in the event of a hello flood assault. The paper addresses the security mechanisms available for 6LoWPAN along with IP security and cutting-edge encryption techniques for CPSs and has no hardware heritage. Since it only discusses the security mechanisms that are currently in use, the paper lacks originality.

In their study of Internet smurfing, flooding, and gray hole attacks in WSNs, Virmani et al. [14] found that the countermeasures for these attacks in the network layer of WSNs included deactivating IP broadcasting at network routers, bidirectional authentication, and multi-hop acknowledgment schemes, respectively. The analysis is based on literature that is currently available regarding several WSN threats. Regarding any simulation or testing of assaults on any platform, the study is not unique. IP filtering and intrusion detection techniques were recognized by [30] as a practical response to the flooding attack in a CPS. The research is unusual in that it is developing and researching industrial intrusion detection systems for various network layer and application layer protocol assaults. A behavior-based intrusion detection system was created by Barbosa and Pras (2010) for the purpose of detecting abnormalities in smart CPSs. A vulnerability assessment of SCADA networks for threats to CPSs in the real world is being done as a result of the study's hardware heritage. Jadidoleslamy noted that the impacts of gratuitous detour could be mitigated by using a central certificate authority in WSNs and pair-to-pair authentication with network layer authentication. The paper reviews assaults, their responses, and the effects they have on WSNs. The study's uniqueness sets it apart from previous research in the field since many aspects of WSN security have been thoroughly covered. A dominant pruning method was presented by Vijayalakshmi and Rabara (2011) and is the most effective way to mitigate the effects of unwarranted detours in CPSs. The method uses a handshaking mechanism to remove non-existent network nodes from lists that are routinely updated.

Transport Layer: Attacks from Outside

CPSs and WSNs both have vulnerable transport layers. The impacts of external attacks on CPS and WSN transport, along with their detection and defense mechanisms, are shown in Table 2.6. In [21], the authors noted that bidirectional authentication of a link using an encrypted echo back mechanism can be used to prevent node depletion caused by flooding. The event-based flooding assault in a CPS was researched by [14]. The study examined a flooding assault on a SCADA system using the DNP-3 protocol and its defenses, and it made recommendations for crypto-based solutions for packet data exchange authentication. For the purpose of authorization in SCADA systems, the study also suggests puzzle-based identification techniques and DNP secure authentication mechanisms. In a brief study of de-synchronization attacks in WSN transport layer, authors in [19] found that the consequences of the attacks might be mitigated by establishing authentication between the levels. The paper examined assaults on WSN layers and suggested various defenses against

Table 2.6 Attacks from outside in transport layer

Attacks	Effects	WSN		CPS	
		Methods of detection	Methods of defense	Methods of detection	Methods of defense
Flooding	Running out of resources, poor availability and traffic [14, 19]	The entire network slows [14, 19]	Bidirectional authentication, the identity verification protocol, and the route error (RERR) message [21]	Dynamic-static jamming	Strong authentication methods based on cryptography, identification methods based on puzzles, DNP3 Secure authentication
De-synchronization	Network disruption, loss of SYNC, and a lack of resources [19]	Warping and delays in the performance [19]	Two-way path check [19]	Mismeasurement of quantities	Estimating the linear condition of nearby areas using PMU readings

them. In order to mitigate the effects of de-synchronization in CPSs, [21] investigated the linear state estimation algorithm. The results of the study are based on power-based SCADA network simulation. A distinctive feature of the study is that it offers a path toward a state estimate approach for PMU measurement in power-based SCADA networks.

Application Layer: Attacks from Inside

Various internal assaults, including distortion of aggregated data, non-repudiation, software tampering, SQL injection, cookie replay, and dictionary attack are also possible at the application layer of a CPS and a WSN. Table 2.7 displays the effects of internal attacks on the application layer of WSN and CPS along with their detection and defense mechanisms. Non-repudiation attack was described by authors in [19] as a selfish assault since nodes in this attack refuse to collaborate with other nodes. In [19], the authors recognized a suitable detection mechanism for the identification of sensor nodes as a workable remedy to mitigate the effect of non-repudiation in WSNs. According to [33], there is a unique method through which authorized operators can eliminate compromised nodes as a result of non-repudiation. In addition to examining the security mechanism in SCADA, this research also provided an experimental model for the IEC 60870-5-104 protocol with a security hardener, which is effectively a single-board computer, for the implementation of security mechanisms in the application layer of SCADA. In contrast, Shin et al. [16] adopted a two-level hierarchical strategy for data aggregation and distortion. The study's strength is based on the numerical analysis of the results, although there are a few issues with non-hardware implementation.

Software tempering can be stopped by using antivirus software and malware scanners, according to authors in [34]. Strong encryption technology was highlighted by Kim (2012) as a deterrent to software tampering. According to research in [34], the WSN can be fixed by reducing the number of attempts on the user account on the application layer. To protect the system from dictionary attacks, [28] emphasized the importance of having a robust password policy and account-locking procedures. The paper gives an introduction to intrusion detection strategies for securing the system from cyberattacks as well as cyberattack mitigation approaches for securing vital industrial systems. The post hasn't given any precise information regarding its implementation and simulated results, making its practicality unclear. In order to prevent cookie replay attacks in the application layer of WSNs, [25] highlighted the use of synchronization session tokens and time stamping techniques. The time stamping technique should be used in the CPS's data transmission protocol, according to [33]. The study offers a hardware legacy to combat cookie replay's effects in a CPS. AES (128 bits) algorithm-accompanied authentication security model has also been implemented to improve the concept's feasibility.

Table 2.7 Attacks from within in application layer

Attacks	Effects	WSN		CPS	
		Methods of detection	Methods of defense	Methods of detection	Methods of defense
Non-repudiation	Selective forward attack setup, packet fraud [19]	Existence of incorrect or misleading log files [19]	Identity of the sensor node, detection method [19]	Existence of incorrect or misleading log files	Eliminated by using operator authentication, each person has their own set of authentication credentials
Data aggregation distortion	Cross-layer assaults, improper environment monitoring, and disrupted data aggregation	Warping and delays in the performance	Two-way authentication	Warping and delays in the performance combined with intelligence [16]	To strike a compromise between security and effectiveness, a hierarchical two-level clustering technique can be applied [16]
SQL injection	Database changes, administrative actions, and O.S. commands	host IDS, alien vault USM network	System for detecting database intrusions based on anomalies, signatures, or honey tokens	Host IDS, alien vault USM network	System for detecting database intrusions based on anomalies, signatures, or honey tokens
Software tampering	Unreliable binary patch Code substitution	Software audit	antivirus programs, malware scanners, and anti-tampering software	Knowledge, behavior, and behavior-specification [30]	Using a robust encryption system, simple identification and isolation of harmful code [30]
Dictionary attacks	Tap a document or message that is encrypted	Password trial failed	Limiting the amount of tries and locking accounts after failed logins	Message formats	Strong password requirements and account locking
Cookie replay	Network Masking	Data conflict brought on by replay, message transmission through illegitimate intermediaries	Synchronization, one-time passwords, session tokens, and time stamping masking	Behavior, particular behavior	Approaches for time stamping in data transfer protocols

Future Directions and Research Challenges

CPS and WSN security is always changing due to new trends and technology. The main trends and technologies in CPS and WSN security are highlighted as follows. The use of machine learning and artificial intelligence (AI) techniques in security measures is one significant trend. Large datasets can be analyzed using AI/ML algorithms to find anomalies, spot potential dangers, and improve real-time decision-making in CPS and WSN. These technologies make it possible to implement preventative security measures and flexible defenses against changing assaults. Blockchain technology usage for improved security and trust in CPS and WSN is another development. Blockchain enables decentralized and tamper-proof data storage, guaranteeing the accuracy and transparency of the stored information. In a dispersed context, it provides secure and verified data sharing between many entities, lowering the danger of data modification and unwanted access. Additionally, Software-Defined Security (SDS) is becoming more popular in CPS and WSN. SDS enables dynamic and programmable security rules that can change in response to evolving system needs and threat environments. It allows for the precise control and application of security controls, improving the robustness and adaptability of CPS and WSN deployments.

Exploring prospective avenues is necessary to advance security in CPSs made possible by WSNs. Following are some crucial areas for development. First, improving protection without sacrificing system performance can be achieved by creating energy- and resource-efficient, lightweight security protocols specifically suited to resource-constrained WSNs. Second, incorporating sophisticated encryption, authentication, and access control techniques can improve data privacy, confidentiality, and integrity. Thirdly, real-time threat identification and mitigation can be made possible by utilizing anomaly detection and machine learning methods. The overall security of CPS–WSN systems can also be improved by addressing trust management, secure communication protocols, and secure hardware designs. Fostering resilient and secure CPS–WSN deployments in important applications can be achieved by putting more emphasis on research in these areas.

Conclusion

Recent studies have taken into account the risks and available defenses for both WSNs and CPSs. Cyber-physical and wireless systems have potential applications. These applications require an efficient security protection solution due to their strategic nature. Due to their layered structures, wireless connectivity, and shared nature, CPSs and WSNs are both more vulnerable to intrusion. The study provides a layer-by-layer analysis of security measures for both wireless and CPSs. The study also focused on contrasting the various cyber-physical and WSN defensive strategies that were available. There is also a layer-by-layer database (Tables 2.1 to 2.7) that lists the assault type and related counter-defensive strategies. We have also presented research directions in a CSP using WSN system.

References

1. Ali S, Qaisar SB, Saeed H, Khan MF, Naeem M, Anpalagan A (2015) Network challenges for cyber physical systems with tiny wireless devices: a case study on reliable pipeline condition monitoring. Sensors 15(4):7172–7205
2. Mohanta BK, Dehury MK, Al Sukhni B, Mohapatra N (2012) Cyber physical system: security challenges in internet of things system. In: 2022 sixth international conference on I-SMAC (IoT in social, mobile, analytics and cloud) (I-SMAC). IEEE, pp 117–122
3. Wan J, Chen M, Xia F, Di L, Zhou K (2013) From machine-to-machine communications towards cyber-physical systems. Comput Sci Inf Syst 10(3):1105–1128
4. Alcaraz C, Lopez J (2010) A security analysis for wireless sensor mesh networks in highly critical systems. IEEE Trans Syst Man Cybern Part C (Appl Rev) 40(4):419–428
5. Mahmood MA, Seah WKG, Welch I (2015) Reliability in wireless sensor networks: a survey and challenges ahead. Comput Netw 79:166–187
6. Yanli Yu, Li K, Zhou W, Li P (2012) Trust mechanisms in wireless sensor networks: attack analysis and countermeasures. J Netw Comput Appl 35(3):867–880
7. Savithri G, Mohanta BK, Dehury MK (2022) A brief overview on security challenges and protocols in internet of things application. In: 2022 IEEE international IOT, electronics and mechatronics conference (IEMTRONICS). IEEE, pp 1–7
8. Saqib A, Anwar RW, Hussain OK, Ahmad M, Ngadi MA, Mohamad MM, Malki Z, Noraini C, Anthony Jr B, Nor RNH et al (2015) Cyber security for cyber physical systems: a trust-based approach. J Theor Appl Inf Technol 71(2):144–152
9. Shafi Q (2012) Cyber physical systems security: a brief survey. In: 2012 12th international conference on computational science and its applications. IEEE, pp 146–150
10. Lu T, Zhao J, Zhao L, Li Y, Zhang X (2014) Security objectives of cyber physical systems. In: 2014 7th international conference on security technology. IEEE, pp 30–33
11. Govindarasu M, Hann A, Sauer P (2012) Cyber-physical systems security for smart grid. Power Systems Engineering Research Center
12. Aloul F, Al-Ali AR, Al-Dalky R, Al-Mardini M, El-Hajj W (2012) Smart grid security: threats, vulnerabilities and solutions. Int J Smart Grid Clean Energy 1(1):1–6
13. Deng J, Han R, Mishra S (2004) Countermeasures against traffic analysis attacks in wireless sensor networks. Comput Sci Tech Rep. cu-cs-987-04
14. Virmani D, Soni A, Chandel S, Hemrajani M (2014) Routing attacks in wireless sensor networks: a survey. arXiv:1407.3987
15. Mohammadi S, Jadidoleslamy H (2011) A comparison of physical attacks on wireless sensor networks. Int J Peer to Peer Netw 2(2):24–42
16. Shin S, Kwon T, Jo G-Y, Park Y, Rhy H (2010) An experimental study of hierarchical intrusion detection for wireless industrial sensor networks. IEEE Trans Industr Inf 6(4):744–757
17. Sabeel U, Maqbool S (2013) Categorized security threats in the wireless sensor networks: countermeasures and security management schemes. Int J Comput Appl 64(16)
18. Mo Y, Kim TH-J, Brancik K, Dickinson D, Lee H, Perrig A, Sinopoli B (2011) Cyber–physical security of a smart grid infrastructure. Proc IEEE 100(1):195–209
19. Sun F, Zhao Z, Fang Z, Lidong D, Zhihong X, Chen D (2014) A review of attacks and security protocols for wireless sensor networks. J Netw 9(5):1103
20. Ali S, Al Balushi T, Nadir Z, Hussain OK, Ali S, Al Balushi T, Nadir Z, Hussain OK (2018) WSN security mechanisms for CPS. In: Cyber security for cyber physical systems, pp 65–87
21. Fatema N, Brad R (2014) Attacks and counterattacks on wireless sensor networks. arXiv:1401.4443
22. Rodrigues A, Best T, Pendse R (2011) Scada security device: design and implementation. In: Proceedings of the seventh annual workshop on cyber security and information intelligence research, pp 1–1
23. Jadidoleslamy H (2014) A comprehensive comparison of attacks in wireless sensor networks. Int J Comput Commun Netw 4(1):2289–3369

24. Taylor CR, Shue CA, Paul NR (2014) A deployable SCADA authentication technique for modern power grids. In: 2014 IEEE international energy conference (ENERGYCON). IEEE, pp 696–702
25. Kaur D, Singh P (2014) Various OSI layer attacks and countermeasure to enhance the performance of WSNs during wormhole attack. Int J Netw Secur 5(1):62
26. Dini G, Tiloca M (2014) A simulation tool for evaluating attack impact in cyber physical systems. In: Modelling and simulation for autonomous systems: first international workshop, MESAS 2014, Rome, Italy, May 5–6, 2014, Revised Selected Papers 1. Springer, pp 77–94
27. Singh H, Agrawal M, Gour N, Hemrajani N (2014) A study on security threats and their countermeasures in sensor network routing. Prevention 3(2)
28. da Silva EG, Dias Knob LA, Araujo Wickboldt J, Paschoal Gaspary L, Zambenedetti Granville L, Schaeffer-Filho A (2015) Capitalizing on SDN-based SCADA systems: an anti-eavesdropping case-study. In: 2015 IFIP/IEEE international symposium on integrated network management (IM). IEEE, pp 165–173
29. McEvoy TR, Wolthusen SD (2011) Defeating network node subversion on SCADA systems using probabilistic packet observation. Int J Crit Infrastruct 9:32–51
30. Kang D-H, Kim B-K, Na J-C (2014) Cyber threats and defence approaches in SCADA systems. In: 16th international conference on advanced communication technology. IEEE, pp 324–327
31. Zhang Z, Trinkle M, Li H, Dimitrovski AD (2013) Combating time synchronization attack: a cross layer defense mechanism. In: Proceedings of the ACM/IEEE 4th international conference on cyber-physical systems, pp 141–149
32. Svendsen NK, Wolthusen SD (2012) Modelling approaches. In: Critical infrastructure protection: information infrastructure models, analysis, and defense, pp 68–97
33. Pidikiti DS, Kalluri R, Senthil Kumar RK, Bindhumadhava BS (2013) SCADA communication protocols: vulnerabilities, attacks and possible mitigations. CSI Trans ICT 1(2):135–141
34. Sastry AS, Sulthana S, Vagdevi S (2013) Security threats in wireless sensor networks in each layer. Int J Adv Netw Appl 4(4):1657

Chapter 3
Firewall: A Vital Constituent of Network Security

Ravi Shankar Yadav and Praveen Likhar

Abstract With the rapid advancement in technology, exchanging information over networks has never been easier than it is today. However, connecting to external networks without compromising internal network assets remains an eternal challenge faced by most establishments and individual users. In the current circumstances, where attackers can easily exploit vulnerabilities in systems and network infrastructure, the firewall continues to play a pivotal role and remains the first line of defence for protecting network infrastructure. In recent years, the number of internet users has increased exponentially, and the use of the internet has changed significantly. Attackers have also become more sophisticated, and threats have increased significantly. Traditional firewalls are no longer sufficient to protect current internet users from emerging threats. To effectively deal with these present emerging threats, firewalls need to be more sophisticated, proactive, and fortified with advanced technologies. Gartner Research has termed these evolved and enhanced firewalls as "Next Generation Firewalls," which are designed to address the current emerging threats in network security. Next-generation firewalls are equipped with a wide range of techniques and features, including application control, IDS, IPS, sandboxing, AV, DLP, threat intelligence, advanced analytics, and many more to tackle modern threats. This chapter will provide details about firewalls, different firewall technologies, their advantages, and disadvantages. It will also present next-generation firewall technologies designed to counter the recent advances in network security threats.

Keywords Firewall · Next gen firewall · Stealth firewall · Futuristic firewall

R. S. Yadav (✉) · P. Likhar
Centre for Artificial Intelligence and Robotics (CAIR), Defence R&D Organisation (DRDO), Bengaluru, India
e-mail: ravi.yadav.cair@gov.in

P. Likhar
e-mail: praveen.likhar.cair@gov.in

D. Gountia et al. (eds.), *Information Technology Security*, Springer Tracts in Electrical and Electronics Engineering, https://doi.org/10.1007/978-981-97-0407-1_3

History

The internet started with the vision of openness for collaboration and information sharing among a community of trusted and attuned users. However, this vision did not last long and came to an end due to various intrusions and attacks. In the year 1988, the Morris worm and many other intrusion attacks were reported in the following years [1, 2]. These attacks, especially the Morris Worm [3], were among the most popular attacks and forced the internet community to reconsider the openness of the internet and develop network protection solutions. These incidents were sufficient to highlight the need for a protection solution to safeguard internet users from online threats. In this chapter, we will describe the most popular network protection solution, the firewall.

The use of the term 'firewall' can be traced back to 1764 when Lightoler used it in the context of building architecture to refer to walls that separated one part of a building from another, specifically to prevent the spread of fire, such as in a kitchen [4]. Later, this term was also used in connection with steam train engines to prevent fire from spreading between the train engine and passenger compartments [5]. These historical examples help explain the use of the term 'firewall' in the technical and computer network context. A firewall is a single system or a group of systems designed to create a barrier between internal users and the potentially precarious external environment. Its objective is to prevent or slow down the spread of malicious activities [6, 7]. Firewalls have been in use for more than three decades and have continually evolved to counter the rising security threats [8]. Even after three decades, the firewall remains a vital network protection solution for organizations and individual users. A brief evaluation of the firewall is provided in Table 3.1.

Table 3.1 Firewall evolution

SN	Year	Major milestone
1	1987	First firewall implementation [4]
2	1989	First firewall concept paper published [9]
3	1992	DEC "securing/screening external access link (SEAL)" first firewall in commercial domain [4]
4	1994	"Stateful packet filter firewall" [10]
5	2004	"Unified threat management (UTM)" coined by IDC [11, 12]
6	2009	Gartner introduces "next generation firewall (NGFW)" [13, 14]

What Is Firewall?

A firewall is a network device with security mechanisms designed to monitor and control incoming and outgoing network traffic, as shown in Fig. 3.1. It is created to implement specific security policies to protect against unauthorized access to information and resources. The primary objective of a firewall is to establish a barrier between trusted internal networks and un-trusted external networks, such as the internal network and the Internet. There are various ways to implement a firewall, resulting in various types of firewalls. However, one common feature among all firewalls is that they filter network traffic at one or more layers of the TCP/IP network, depending on their implementation. Some firewalls even filter traffic at multiple layers of the TCP/IP network. Firewalls typically filter network traffic based on criteria such as source, destination, service, and content.

To effectively implement a firewall, it should be placed at the periphery of the network to be protected. It should mediate all traffic between networks and have an efficient filtering mechanism to enforce specified policies. Additionally, it should be robust, secure, fail-safe, and possess capabilities such as resource monitoring, a stealth mode of operation, low latency, high throughput, accounting, and auditing. The essential and desirable features of the firewall are listed in Table 3.2.

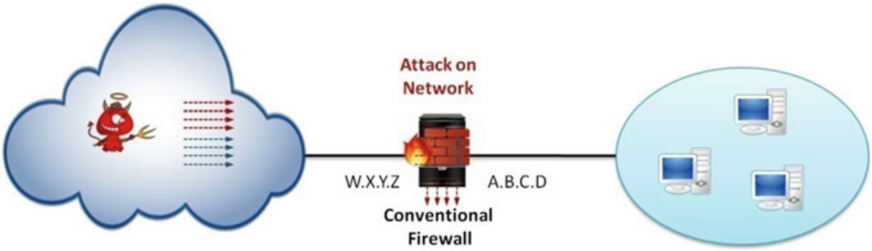


Fig. 3.1 Conventional firewall

Table 3.2 Firewall features

Essential features	Desirable features
• Places at periphery of network to be protected • Mediate all the traffic between protected and external networks • Efficient filtering capabilities • Fail safe	• Robust against the security attacks • Capability for fail safe configuration • Capability for accounting and auditing • Capability to monitor network resources • Capability of rate limiting of network traffic • Capability to remain invisible in the network • Capability for policy verification and optimization • Low latency and high throughput

Basic Definitions

Frequently used basic definitions related to firewalls are described as follows.

Host

A host is an end system connected to a network, usually a computer. A compromised host system becomes threat to entire internal network hence needs to be protected from the outside attacks.

Dual/Triple/Multi-homed Host

A dual/triple/multi-homed host is a computer system with two/three/multiple network interfaces. Based on the deployment requirements, a firewall can be in any of these forms.

Bastion Host

A bastion host is a computer system that serves as the main contact point for internal users for data/information transactions with the Internet. It must be highly secured due to its exposure to the Internet. This term was popularized by Marcus Ranum [15] in the professional community.

TCP/IP Packet

A TCP/IP packet, generally referred to as a packet, is the basic unit of data communication on computer networks. In TCP/IP all communication takes place in the form of packets.

Screening/Packet Filtering

Screening or packet filtering is the action taken by a filtering device (firewall) to control the flow of data in a network. Packet filtering is performed based on specified parameters.

De-Militarized Zone (DMZ)

DMZ is an intermediate network between a protected internal network and an un-trusted external network, serving as a supplementary layer of security. A De-Militarized Zone (DMZ) network is sometimes also referred to as a perimeter network.

Proxy Server

A proxy server is a system with a program to receive, process, and forward traffic intended for another system (server/client). For a client program, it pretends to be an application server, and for an application server, it pretends to be a client program.

Policy

It is a set of rules that describe what is permitted and what is not permitted. A clear, consistent and unambiguous policy is very important for firewall system to be effective.

Working of Firewall

The main function of any firewall is to implement a security policy. Firewalls implement security policies by applying filters to network traffic. Security policies are defined as sets of rules, and appropriate actions are taken on network traffic based on these rules. For example, a rule might specify that internal users can access a web server on the internet, while another rule might prohibit external users from accessing servers hosted in the internal network. To implement a policy, a firewall creates rules specifying the source, destination, service, and the corresponding actions to be taken. If any network traffic passing through the firewall matches a rule, the corresponding action is applied. Organizational policy is specified as a set of rules, and firewalls enforce it on network traffic. Since it is practically impossible to specify rules for all potential scenarios, a default policy is defined for unspecified traffic. In most cases, the default policy only defines the action (Allow, Block, Log, etc.) to be applied to traffic that doesn't match any rules from the policy. Great care should be taken when defining the default policy [15, 17].

Firewalls Techniques

Firewalls use the following four techniques to implement access control and security policy [16].

Service Control

This is the most basic control used by all firewalls. It is used to identify which internet services are allowed. To achieve service control, packet filter firewalls inspect and filter network traffic based on IP address, port number, and protocol. This can also be achieved through proxy firewalls, which inspect each service transaction before forwarding.

Direction Control

Direction control identifies the direction of the service request, whether it is inbound (ingress) or outbound (egress). To achieve direction control, firewalls inspect and filter network traffic based on the service and its direction.

User Control

User control identifies the user who is attempting to access the service. Packet filter firewalls typically depend solely on the source IP address, but modern firewalls inspect additional information, like login monitoring, to identify end users.

Behavior Control

This control identifies how a particular service can be used. For example, allowing some web services while blocking other web services based on the specified security policy.

Firewall Design Goals

Almost two decades ago, Bellovin et al. described firewall design goals in their publications on network firewalls [17]. These design goals are still relevant for modern firewalls and are explained below.

Complete Mediation

Firewall should be at choke point, and all the traffic should pass through it. This is achieved by blocking all the external access to user network except through the firewall. Various network configurations can be deployed as per the requirements.

Network Traffic Authorization

Only network traffic authorized by the security policy should be allowed to pass through the firewall. A variety of firewalls can be used to implement this security policy.

Firewall Robustness

The firewall should be robust enough to withstand penetration attacks. Therefore, the underlying hardware, software, and firmware should be hardened and trusted.

Types of Firewall

Firewall technologies heavily depend on the underlying network model. To understand different firewall types, it is necessary to comprehend the "Open Systems Interconnection (OSI)" Model. The OSI network model is a fundamental concept in computer networking, categorizing network data processing into seven distinct groups, each with a specific function for each group. These processing groups are commonly known as network layers. The very first layer interfaces directly with the underlying hardware, while the topmost layer serves as the primary interface for user-level applications and services. Some literature refers to the first layer as the "hardware layer" and the topmost layer as the "software layer." The intermediate layers facilitate the transfer of data between hardware and software. The layers of the "Open Systems Interconnection (OSI)" model are as follows:

- Layer-7: Application Layer
- Layer-6: Presentation Layer
- Layer-5: Session Layer
- Layer-4: Transport Layer
- Layer-3: Network Layer
- Layer-2: Data Link Layer
- Layer-1: Physical Layer.

Firewalls can be implemented in various ways, with implementation techniques differing based on the "Open Systems Interconnect (OSI)" network layer at which they primarily operate. Different types of firewalls, categorized based on functionalities and network layer of implementations, are explained in the following section.

Packet Filtering Firewalls

Packet filtering is one of the most popular and oldest techniques for implementing firewalls. This type of firewall operates at the network layer of the OSI model, as depicted in Fig. 3.2. Packet filtering firewalls primarily perform filtering based on the fields of the network packet headers. The policy for packet filtering firewalls is defined based on the network packet header fields. Commonly used fields for filtering include source and destination IP addresses, protocol, source, and destination port numbers. These firewalls examine each packet for specific fields in the set of rules and determine the action for the packet, whether to allow or drop it. Packet filter firewalls check each network packet separately without maintaining any information about previously related packets. As a result, these firewalls are known as stateless firewalls. Due to their statelessness, these firewalls have limited filtering capabilities to defend against advanced attacks. Packet filtering firewalls inspect each network packet individually, which requires more processing compared to stateful firewalls.

Generally, packet filtering firewalls are effective, quick, and cost-effective. They inspect only packet headers and cannot scrutinize the payload data of the packets, making them unable to prevent packets from trusted IP sources with malicious contents. Additionally, due to their statelessness, these firewalls are vulnerable to various connection state related attacks. Despite their limited functionalities, packet filtering firewalls are easy to implement and served as the foundation for contemporary firewalls.

Advantages:

- Fast and low-cost
- Easy to implement
- Most simple and oldest firewall.

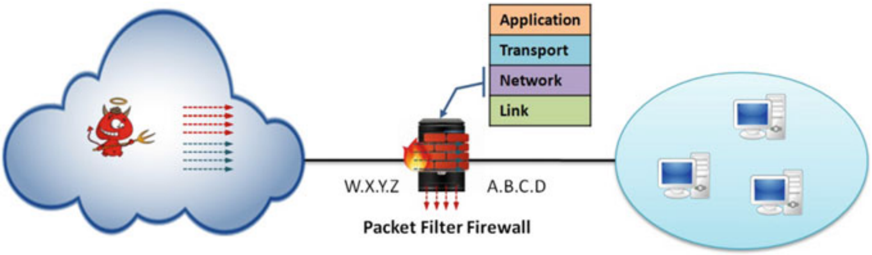


Fig. 3.2 Packet filter firewall

Disadvantages:

- Limited protection against advanced threats.

Circuit-Level Firewall

Circuit-Level Firewalls, also referred to as 'Circuit-Level Gateways,' are a different yet straightforward type of firewall. These firewalls are easy to configure and require minimal processing resources. They operate at the session layer of the OSI network model and the transport layer of the TCP/IP network stack, as shown in Fig. 3.3. Circuit-Level Firewalls primarily inspect Transmission Control Protocol (TCP) connections and TCP sessions to validate established TCP connections and monitor ongoing sessions. The main function of these firewalls is to determine the security status of established TCP connections. To achieve this objective, Circuit-Level Firewalls scrutinize established TCP sessions and ensure that each network packet belongs to a valid TCP session. When an internal host requests the initiation of an outbound TCP connection, Circuit-Level Firewalls establish a connection with an outside host on behalf of the initiating host. By doing this, these firewalls keep details such as the IP address of the internal host concealed from the outside world.

Circuit-Level Firewalls are similar to Packet Filtering Firewalls in a way that they check transaction information only and do not inspect the payload data of network packets. Consequently, these firewalls cannot prevent network packets (network traffic) containing malicious data but adhering to permitted and valid TCP sessions. Despite being simplistic, cost-effective, resource-efficient, and quick to implement, these firewalls are considered a good first level of defense but are not competent enough against advanced levels of network attacks. However, their inability to examine the payload of network packets imposes an obligation to deploy an additional, different type of firewall to achieve a higher level of protection.

Advantages:

- Easy and quick to implement
- Resource efficient

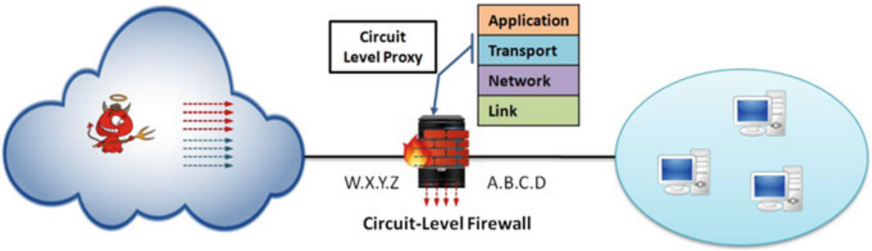


Fig. 3.3 Circuit-level firewall

- Simple and cost effective.

Disadvantages:

- Restricted protection
- Require additional layer of defense to achieve higher level of protection.

Stateful Multi-layer Inspection (SMLI) Firewalls

As the name suggests, Stateful Multi-Layer Inspection Firewalls work on multiple layers of the OSI network model, as shown in Fig. 3.4. These firewalls can scrutinize network traffic based on packet headers and can also verify TCP connection handshakes. In addition to these capabilities, SMLI firewalls can maintain the status of TCP connections in a data structure known as a state table, tracking connection status for validity. The state table stores TCP handshake and session information, which includes source IP Address, destination IP Address, source Port number, destination Port number, TCP connection status, etc. These firewalls dynamically update the state table, hence they are also referred to as Dynamic Firewalls. An example state table is shown in Table 3.3.

When a TCP connection passes through an SMLI firewall, a state entry is created in the state table to store connection state information. After the establishment of a TCP connection, subsequent network traffic flowing within that particular session is inspected according to the information in the state table. These additional features

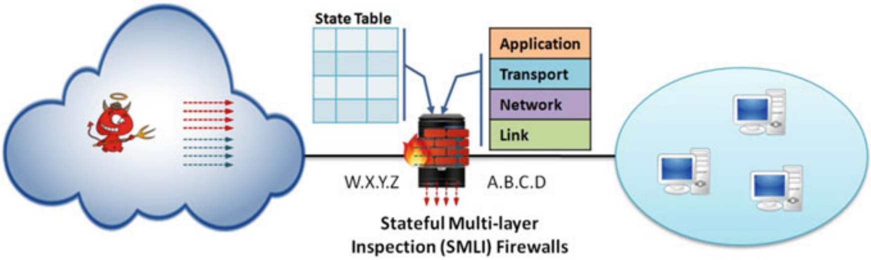


Fig. 3.4 Stateful multi-layer inspection (SMLI) firewall

Table 3.3 State table example

Source IP address	Source port	Destination IP address	Destination port	Connection state
10.11.12.1	1234	201.1.2.3	8080	Established
10.11.12.2	1235	202.11.2.1	80	Established
10.11.12.3	1236	203.12.2.2	25	Initiated
10.11.12.4	1237	204.13.2.3	110	Initiated

make SMLI firewalls more secure and advanced compared to simple stateless Packet Filtering Network Firewalls.

Advantages:

- Better security compared to stateless firewall
- Higher throughput.

Disadvantages:

- Considerable requirements of system resources.

Deep Packet Inspection Firewalls

These firewalls have the capability to inspect the application layer content of network packets along with stateful inspection, as shown in Fig. 3.5. ‘Deep packet inspection’ provides the mechanism for ‘stateful protocol analysis.’ Through ‘stateful protocol analysis,’ these firewalls can perform protocol-specific analysis on network traffic to determine protocol-specific compliance. For example, these firewalls can enforce the types of email attachments allowed or not allowed. Additionally, these firewalls can inspect unexpected command sequences, commands used to exploit various attacks like buffer overflow, Denial of Service (DoS), malware, and other attacks within application protocol data. These firewalls can also perform input validation for application layer protocols. Deep Packet Inspection Firewalls are available for almost all popular application layer protocols like HTTP, SMTP, POP, IMAP, VoIP, etc. [18].

Advantages:

- Enhanced protection through content inspection
- Can perform protocol-specific analysis to detect protocol specific attacks.

Disadvantages:

- Higher processing requirement
- Application specific implementation is required.

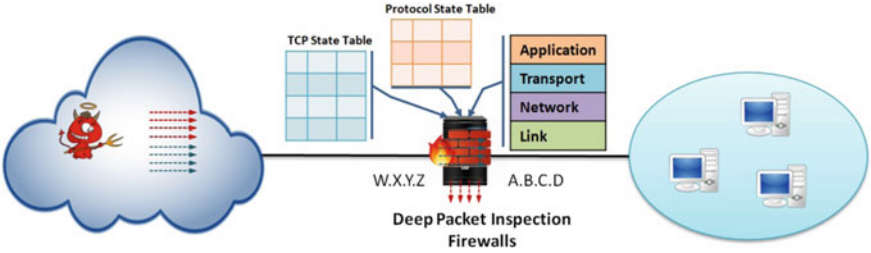


Fig. 3.5 Deep packet inspection firewall

Application Layer Firewalls (Proxy Firewalls)

Application Layer Firewalls operate at the application layer of the OSI model, as shown in Fig. 3.6. The Application Layer is the topmost layer of the OSI network model, responsible for functions related to applications. Application Layer Firewalls work at the application layer, earning them the name 'Application-Level Gateways.' Various application layer protocols are used in the OSI model and TCP/IP network model, including popular protocols such as "Secure Shell (SSH)," "Hypertext Transport Protocol (HTTP)," "File Transfer Protocol (FTP)," Telnet, "Simple Mail Transport Protocol (SMTP)," and more.

In contrast to other firewalls, Application-Level Firewalls break the direct connection between clients and servers to prevent direct communication between internal clients and external servers. To achieve this, these firewalls connect to servers on behalf of clients, and responses from servers are forwarded to the original clients. These firewalls initiate and forward data, pretending to be the client or server, and are therefore referred to as 'Proxy Firewalls.' This approach protects client-sensitive information, such as location and identity, from the outside network, adding an extra layer of protection by concealing the client's identity. For established TCP connections, these firewalls analyze the content of network traffic before forwarding it to clients or servers. These firewalls check network traffic for compliance with specified rules and forward it if deemed suitable. Proxy firewalls are slower compared to other types of firewalls.

Advantages:

- Performs protocol and content analysis to detect various attacks
- Provide additional security to end system by preventing direct connection between client and server.

Disadvantages:

- Application layer protocol specific implementation
- Higher processing and memory requirement.

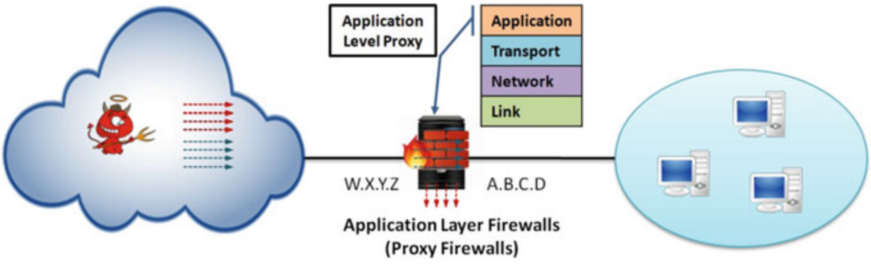


Fig. 3.6 Proxy firewall

Stealth Firewall

Stealth firewalls are IP-invisible in the network. These firewalls operate as network bridges on the data link layer of the network stack to remain IP-invisible. Likhar et al. [19] presented an approach to implement stealth firewalls using the netfilter framework of Linux. These firewalls have all the functionalities of stateful multilayer firewalls, and additionally, they remain IP-invisible in the network, as shown in Fig. 3.7. This IP invisibility provides additional security for the firewall itself.

In our work titled "Stealth Firewall: Invisible Wall for Network Security" [19], we presented an approach to implement stealth firewall using the Linux firewall framework to overcome the limitations of traditional firewall. In this work, the performance of stealth firewalls is also compared with traditional firewalls, as depicted in Fig. 3.8.

The performance of the stealth firewall can be improved by implementing it on the nftable framework, a new Linux firewall framework [20, 21].

Advantages:

- Stealth mode of operation for additional security
- Seamless deployment due to bridge mode
- No network changes required for deployment
- Additional capability to filter layer-2 traffic.

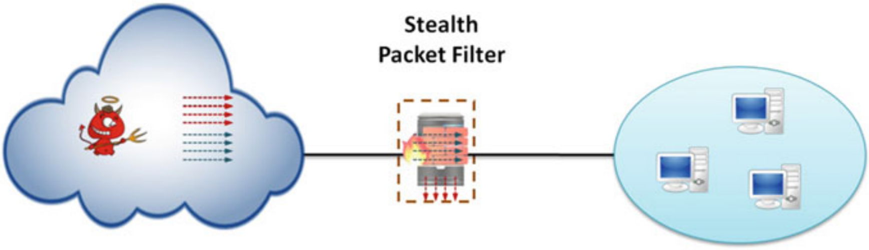


Fig. 3.7 Stealth firewall

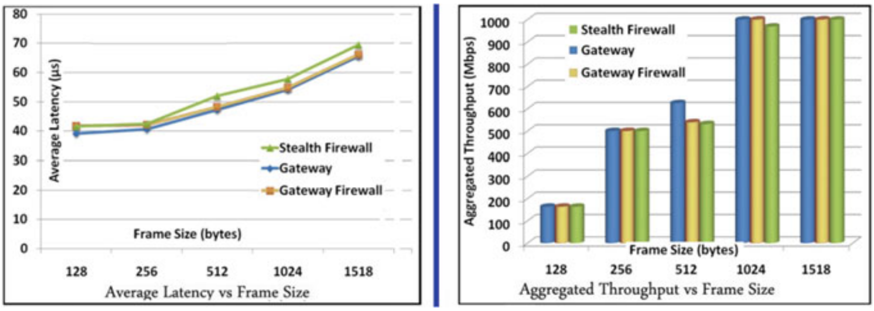


Fig. 3.8 Stealth firewall performance

Disadvantages:

- Need additional processing for implementation of stateful feature on layer-2 packets.

Unified Threat Management (UTM)

Unified Threat Management (UTM) are special types of firewalls that combine multiple network security features, such as antivirus, intrusion prevention, with stateful multi-layer inspection filtering. The primary reason for combining multiple features into a single device is to simplify the setup and maintenance of security policies. A UTM firewall should typically have stateful inspection, malware detection and prevention, prevention of malicious network probes, etc. The main advantage of UTM is the simplification of configuration and management. However, it comes with a performance tradeoff [11, 18].

Advantages:

- Provide multiple security solution from single console
- Simplified configuration and management.

Disadvantages:

- Performance tradeoff
- May lead to single point of failure.

Next Generation Firewall (NGFW)

Next-Generation Firewalls, popularly referred to as NGF, are defined by Gartner as “deep-packet inspection firewalls that, along with port/protocol inspection and filtering, also have the capability of inspection at the application level, intrusion detection-prevention, and consuming intelligence from the outside world” [13]. The main objective of Next-Generation Firewalls is to overcome the limitations of conventional firewalls and provide additional security features. NGFWs should offer a flexible architecture, deep packet inspection capabilities, along with conventional firewall features [13, 14].

Next-Generation Firewalls combine most of the functionalities of their predecessor conventional firewalls with better performance. These firewalls are more robust and provide deeper and wider protection compared to their predecessors. NGFWs detect malware and anomalies by performing deep packet inspection, and they perform resource and traffic analysis through application awareness functionality. They can prevent DDoS attacks to a greater extent, along with data breach protection through Secure Sockets Layer (SSL) packet decryption functionality. Unlike conventional firewalls that rely on IP addresses for user identification, NGFWs are capable of identifying users and their roles. NGFWs are also used to ensure compliance with various statutory standards, such as the ‘Payment Card Industry’ (PCI) and the ‘Health Insurance Portability and Accountability Act’ (HIPAA). These firewalls integrate various network security features into a single solution [13, 14].

Advantages:

- Heuristic based threat detection and prevention
- Prevents advance and latest security threats like malware and sophisticated intrusion
- Integrates various network security features into a single solution.

Disadvantages:

- Expensive and resource intensive.

Cloud Firewall

‘Firewall-as-a-service’ is also referred to as a cloud firewall. A cloud firewall is designed using cloud technologies to provide firewall functionalities in the cloud. These firewalls are maintained and operated by security service vendors. In recent scenarios where most work is carried out in the cloud, these firewalls play significant roles in providing network security for cloud-based applications.

Advantages:

- Simple and flexible to deploy
- Better scalability.

Disadvantages:

- Sometimes very expensive
- Need to rely on security vendors for operation and maintenance.

Performance

The Annual Report for 2018–2023 from Cisco estimated that there would be 5.3 billion Internet users by 2023 [22]. Internet traffic has seen a three to fourfold increase in the past four to five years, reaching the order of zettabytes. Given the current scenario, it can be predicted that Internet traffic will continue to increase exponentially in the near future. Therefore, the throughput performance of all network components is a vital requirement and will remain important in the future.

Firewalls, which are placed at the network choke points of network infrastructure, are essential not only for security but also for network performance. While advancements in computing hardware have provided some support to firewall developers in meeting performance requirements, this alone is not sufficient to address

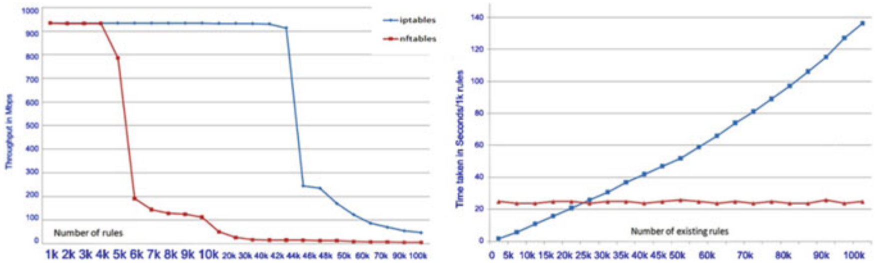


Fig. 3.9 Ruleset size and throughput, ruleset size & RTT

the herculean performance demands. As a result, the majority of firewall developers have adopted improved design methodologies to achieve the necessary performance.

One of the most popular firewall frameworks, Netfilter [23], has introduced the nftable [25] packet filter format to replace the iptables [24] framework. In our previous work [21], we conducted an empirical study to assess the performance advantages of nftables over iptables and obtained very encouraging results. The results from our study [21] are reproduced in Fig. 3.9, which illustrates a significant performance improvement of nftables over iptables.

Attacks on Firewall

A firewall is a crucial component of network security, handling all incoming and outgoing traffic. Firewalls, like any other network systems, are susceptible to attacks. In the worst-case scenario, an attacker could compromise the firewall, potentially cutting off all communication through it, disconnecting private networks from the outside world, and causing significant losses for organizations. Firewalls share vulnerabilities with other network systems, such as fingerprinting [26] and denial of service (DoS)/distributed denial of service (DDoS) attacks [27, 28].

Firewall Fingerprinting

In a study by Alex et al. they presented a technique for remotely fingerprinting firewalls and exploiting this information to launch attacks on them. Therefore, it is essential to implement recommended anti-fingerprinting methods as part of firewalls. These methods include silently dropping rejected traffic, altering default network attributes, and limiting or dropping system scan traffic. Working in stealth mode is also an effective firewall fingerprinting protection mechanism [19].

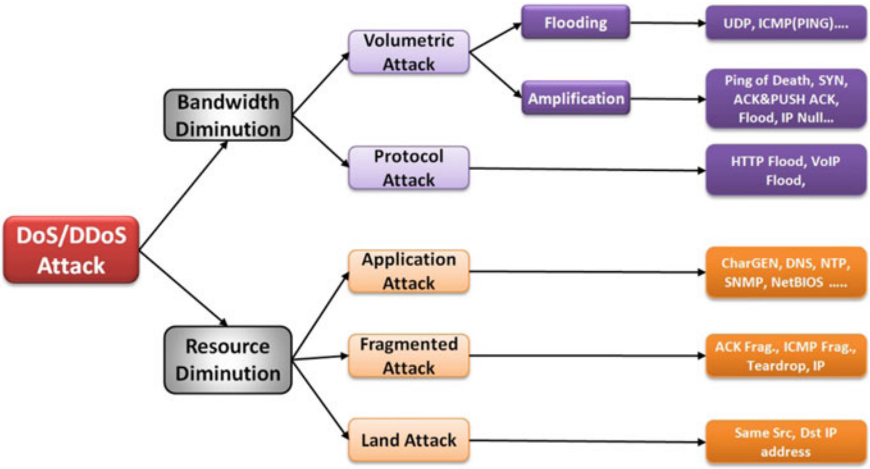


Fig. 3.10 Types of DoS/DDoS attacks

Distributed Denial of Service (DDoS) and Denial of Firewall (DoF)

Denial of service (DoS) is a broad class of attacks that disrupt network services and resources by overwhelming them with a massive volume of network traffic beyond the capacity of network devices to handle. DoS and distributed DoS (DDoS) attacks are similar, but they differ in scale. DoS attacks typically originate from a single system, whereas DDoS attacks involve multiple systems, often with spoofed internet addresses, and are distributed across various geographical locations. In these attacks, the firewall is inundated with falsified network traffic, rendering it unresponsive. There are numerous ways to generate DoS attacks, ranging from simple ping flooding to multi-vector attacks. Various types of attacks are summarized in Fig. 3.10.

Specialized DoS attacks designed for firewalls are known as denial of firewalling attacks (DoF). DoF attacks typically use a low volume of malicious network traffic to target firewalls. The BlackNurse attack is an example of a DoF attack that employs specially crafted ICMP error messages to target firewalls. A case study of this attack on various modern firewalls was published by Trabelsi et al. [29].

DoS defense mechanisms can be categorized into static and dynamic approaches. Static approaches involve preemptive measures like filtering malicious traffic to discard potentially malicious traffic and deploying distributed firewalls. Dynamic approaches can be divided into reactive and proactive methods, requiring the capability to monitor early signs of attacks and robust protective measures upon detection. These defense mechanisms are summarized in Fig. 3.11.

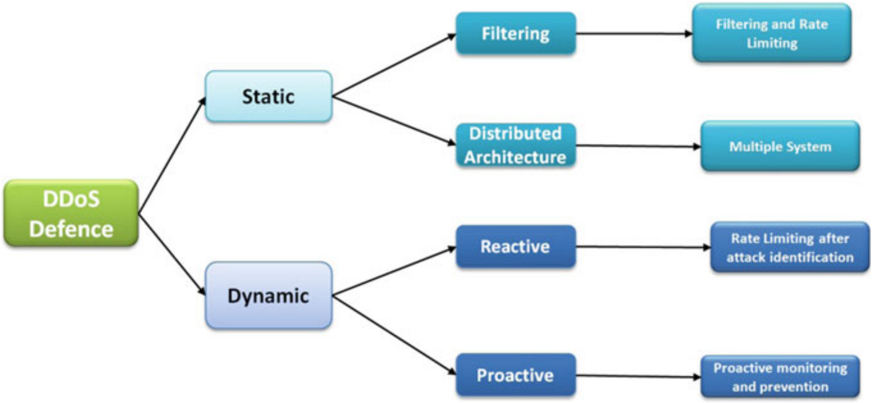

DDoS Defence
Static
Dynamic
Filtering
Distributed Architecture
Reactive
Proactive
Filtering and Rate Limiting
Multiple System
Rate Limiting after attack identification
Proactive monitoring and prevention

Fig. 3.11 DDoS/DoF defence mechanisms

Challenges for Modern Day Firewalls and Solution

In the present day, most enterprise networks no longer have external perimeters. Furthermore, due to advancements in network technologies like distributed applications and the Internet of Things, traditional firewall architectures based on a boundary between the external and internal networks have become less effective. To address the protection requirements of perimeter-less enterprise networks, a network virtualization (NV) based approach, popularly known as micro-segmentation, has evolved.

Micro-segmentation addresses the latest challenges of perimeter-less networks by creating multiple secure zones within the enterprise network at a fine-grained level. Unlike a secure perimeter around the entire infrastructure, micro-segmentation deploys software-defined and controlled security around each segment within the network. This enhances the security mechanism's ability to prevent attacks and halt the lateral spread of any security breaches.

The main advantage of micro-segmentation-based firewalling over traditional perimeter-based firewalling is fine-grained control and isolation. In traditional firewalling, a robust layer of security is intended to block all infiltration into the network, much like security guards at the entrance of an organization. In micro-segmentation, it is akin to having many security guards inside, protecting the organization. Network virtualization is the first step in implementing micro-segmentation, which involves dividing the hardware and software network infrastructure into specific segments and wrapping a firewall around individual segments in the network. This finer level of security provides flexibility in the network and safeguards against the spread of any network infections.

Discussion

The first firewall was developed more than three decades ago, and it still remains a vital component of network security. Firewalls hold significant importance due to their profound influence on contemporary network security techniques. These network security devices are used to filter malicious traffic and control the flow of network traffic in accordance with the network security policy. Firewalls have consistently demonstrated their significance over a period of more than three decades.

One of the most important principles of firewall design is to mediate all network traffic between the external and internal networks. Therefore, firewalls are often deployed at the network perimeter. To provide security against network-borne attacks, firewalls mediate all network traffic and filter out unwanted traffic by enforcing the network security policy. Over time, filtering technology has evolved into various types of firewalls. Depending on the security requirements and deployment levels, different types of firewalls can be implemented.

In today's scenario, there are rarely any networks completely disconnected from the external network. Hence, having a firewall to control network traffic is of utmost importance. Real-time detection of malicious data in network traffic is one of the primary requirements of modern firewalls.

Firewalls should be implemented not only to secure the network infrastructure but also to protect themselves, as improper implementation of a firewall can lead to its failure or compromise. In the event of a firewall compromise, the damage to network assets is extremely high, and it is exceedingly difficult to estimate or infer the extent of the damage. Therefore, the utmost care should be taken when implementing a firewall.

Conclusion

As enterprise technology continues to evolve and change, there has been a significant transformation in firewall design and approach. The advancement of networking technologies has given rise to a plethora of new internet applications, which are extensively used by internet users. In parallel, the threat landscape evolves continuously, becoming more complex with modern enterprise technology.

Traditional firewalling approaches are often ineffective against this modern threat landscape. Next-generation firewalls (NGFWs), equipped with advanced machine learning and deep learning technologies, and are poised to address these treacherous and emerging threats. The integrated approach with AI at its core will revolutionize the firewalling approach from reactive to proactive security. These firewalls enhance threat detection capabilities by employing intelligent threat detection engines and continuously updating and optimizing models based on new live data.

There is a pressing need for NGFWs to incorporate Artificial Intelligence to fortify and strengthen protection against advanced network threats and previously unknown threats.

References

1. Stoll C (1989) The Cuckoo's egg. Doubleday, New York, NY
2. Cheswick B (1990) The design of a secure internet gateway. In: USENIX summer conference
3. Eichin MW, Rochlis JA (1989) With microscope and tweezers: an analysis of the internet virus of November 1988. In: IEEE computer society symposium on security and privacy. IEEE Computer Society, Los Alamitos, CA, USA
4. Ingham K, Forrest S (2002) A history and survey of network firewalls. ACM J
5. Schneier B (2000) Secrets and lies: digital security in a networked world. Wiley, New York, NY
6. Lodin S, Schuba C (1998) Firewalls fend off invasions from the net. IEEE Spectrum
7. Cheswick W, Bellovin S (2003) Firewalls and internet security: repelling the Wily Hacker. Addison-Wesley
8. Chen S, Iyer R, Whisnant K (2002) Evaluating the security threat of firewall data corruption caused by instruction transient errors. In: International conference on dependable systems and network
9. Mogul JC (1989) Simple and flexible datagram access controls for Unix-based gateways. In: USENIX summer conference
10. Chapman D, Zwicky E, Cooper S (2000) Building internet firewalls, 2nd ed. O'Reilly
11. Wilson J (2005) The future of the firewall. Bus Commun Rev
12. https://www.firewalls.com/what_is_utm_firewall
13. Gartner. https://www.gartner.com/en/information-technology/glossary/next-generation-firewalls-ngfws
14. Audin G (2004) Next-gen firewalls: what to expect. Bus Commun Rev
15. Ranum MJ (1992) A network firewall. In: First world conference on system administration and security
16. Teach computer science. https://teachcomputerscience.com/firewall/#Firewall_Security_Techniques
17. Bellovin S, Cheswick W (1994) Network firewalls. IEEE Commun Mag
18. NIST Special Publication SP 800-41_Rev1, Sep 2009
19. Likhar P, Yadav RS (2020) Stealth firewall: invisible wall for network security. In: Innovations in computer science and engineering. LNNS, vol 103. Springer
20. Suehring S (2015) Linux firewalling, enhancing security with nftables and beyond, 4th ed. Addison-Wesley
21. Likhar P, Shankar Yadav R (2021) Impacts of replacing venerable iptables and embracing nftables in a new futuristic Linux firewall framework. In: 5th international conference on computing methodologies and communication (ICCMC)
22. Cisco Annual Internet Report (2018–2023). https://www.cisco.com/c/en/us/solutions/collateral/executiveperspectives/annual-internet-report/white-paper-c11-741490.pdf
23. The netfilter.org project: netfilter/iptables project. https://netfilter.org/projects/iptables/index.html
24. Purdy GN (2004) 'Linux iptables pocket reference. O'Reilly Media
25. McHardy P, Ayuso PN (2015) The nftables tutorial. In: Proceedings of the Netdev 0.1, Ottawa, Canada, February 2015
26. Alex L, Amir K, Joshua H, Zihui G, Dan P, Jia W (2017) Firewall fingerprinting and denial of firewalling attacks. IEEE Trans Inf Forensics Secur 12(7)

27. Salah K, Sattar K, Sqalli1 M, Al-Shaer E (2011) A potential low-rate DoS attack against network firewalls. Secur Commun Netw 4:136–146
28. Singh A, Gupta BB (2022) Distributed denial-of-service (DDoS) attacks and defense mechanisms in various web-enabled computing platforms: issues, challenges, and future research directions. Int J Semant Web Inf Syst (IJSWIS) 18(1):1–43
29. Trabelsi Z, Zeidan S, Hayawi K (2019) Denial of firewalling attacks (DoF): the case study of the emerging BlackNurseAttack. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2915792

Chapter 4
Exploring the Landscape of Password Managers for Individual Users Through Innovative Solution

Taskeen Zaidi, Suman Garai, and Trishul V. Biradar

Abstract With the increasing number of online services available, it is crucial for individuals to have control over their password management systems, which generate, store, and retrieve passwords, ensuring that they meet strict security standards to safeguard user information. As password management can be challenging, password managers have gained popularity, as users become more aware of the dangers of password reuse. This research evaluates the most popular password management tools on the market to assess their security and usability, examining aspects such as efficacy, convenience, and security, as well as multi-factor authentication and the program's security. The study aims to identify the best factors for creating the most secure password management system in today's world, providing insights on designing a secure and user-friendly password generation, storage, and retrieval system. By analyzing existing systems' algorithms, the research team proposes an innovative solution with a focus on security protocols, and effective password storage, and retrieval. Additionally, the research explores the potential for future improvements and advancements to the password manager. The results of this research will contribute to a more secure online environment for users and help to ensure that their personal information remains protected.

Keywords Encryption · Hashing · Cloud · Multifactor authentication · Derivation key · Generation · Storage · Auto fill

Introduction

Password-based authentication is still the most used form of authentication on the web, despite its well-known challenges. Security standards often require the use of complex and unique passwords, but these types of passwords are difficult for users to remember. As a result, users often create weaker passwords that are easier

T. Zaidi (✉) · S. Garai · T. V. Biradar
Department of Computer Science and IT, Jain Deemed-to-Be-University, Bengaluru, India
e-mail: t.zaidi@jainuniversity.ac.in

D. Gountia et al. (eds.), *Information Technology Security*, Springer Tracts in Electrical and Electronics Engineering, https://doi.org/10.1007/978-981-97-0407-1_4

to recall or use simple modifications to popular passwords. This puts them at a higher risk of being targeted by attackers. According to Lo, when asked to reset a password, users tend to utilize the same password with personally identifiable information appended, such as dates of birth or nicknames. However, because a dictionary attack may quickly create probable passwords based on commonly used phrases, these personal traits are worthless in defending against individual hacking efforts [1]. Additionally, human memory is limited and focuses on familiarity and repetition, making it difficult for individuals to store complex and random sequences of characters. Research has shown that people can only remember seven characters at a time, which is not enough to store the types of passwords that are considered secure in today's online world. Despite the growing number of data breaches, users still often reject security advice as it requires a significant effort, especially when the number of users affected by breaches is low. However, this situation leaves many individuals vulnerable to exploitation as the number of data breaches continues to rise [2]. Password management systems are increasingly being used by individuals to keep their online passwords secure. These systems generate, store, and retrieve passwords for the user, thus reducing the mental load of remembering multiple passwords. A password manager stores the user's login credentials, including usernames and passwords, in an encrypted vault, which can be accessed online across multiple devices. The vault is encrypted using a master password chosen by the user. The main purpose of a password manager is to reduce password reuse, as it generates strong passwords that are then saved and easily accessible to the user. Many password managers offer a password generation pop-up window to encourage users to create strong passwords.

However, there are security challenges associated with password managers. One such issue is the risk of auto fill functionality being exploited by attackers to steal user credentials. In addition, password vaults may not always be secure, as unencrypted metadata or side channel leaks from encrypted data can leave users' credentials vulnerable to attack. There is also a lack of Research examining the security of password generation in password managers. While password managers do offer many benefits, including reducing password reuse, it is important to be aware of the security challenges that they may present [3]. Further research is needed to ensure the security of password generation in password managers.

Related Work

In this review study, we carried out a thorough evaluation of different password management tools available in the market, taking into consideration several essential factors such as 2FA Support, Independent Security Audits, Open-Source Availability, Encryption Standards, IP Restrictions, Hosting Options, Public Bug Reporting Programs, Customizable Password Generation, Data Breach Notifications, and Vulnerability Analysis.

Our investigation of some of the most used password management solutions was useful in increasing our understanding of the security features present in these solutions. This information was then used to select a solution that meets our specific needs. The results of our investigation have been organized in an easy-to-read format, with a checkmark representing the presence of a feature and a cross representing its absence or unproven existence [4].

In the following pages, we will delve deeper into each of the criteria mentioned above and discuss their implications for the applications. Our study aimed to provide users with a comprehensive understanding of the various password management solutions available in the market and help them make an informed decision when selecting a solution that meets their specific needs. Now, let us explore the criteria based on which password managers are classified and the implications of each criterion.

Encryption standard is a protocol used to secure data by converting it into an unreadable format. For a password manager, it is crucial to have a strong and widely trusted encryption standard, such as AES or RSA, to protect the stored passwords from unauthorized access. The strength of the encryption is determined by the key length, with a longer key length providing a stronger encryption. Different password managers may use different encryption methods and have trade-offs between security and performance.

Two-factor authentication (2FA) or multifactor authentication (MFA) involves providing two or more means of verification in order to access an account or system. The first step typically involves entering a password, while the subsequent step involves furnishing either information that the user knows, possesses, or is (such as a security question, a device, or biometric information). In the context of a password manager, 2FA can provide an extra layer of security but can also impact usability by slowing down the access process or causing inconvenience if the second factor is lost.

Biometric authentication is a type of identity verification that uses biological traits such as fingerprints or facial recognition to identify an individual. For a password manager, biometric authentication can combine something the user is with something the user knows to add an extra layer of security. However, there are concerns about biometric data being compromised and privacy issues, making it important for password managers to implement strong security measures for biometric authentication.

Password generation refers to the process of automatically creating secure, random passwords to protect online accounts. In a password manager, password generation can simplify the process of using strong passwords and the password manager can store these passwords securely. Password managers typically allow users to customize the generated password parameters, such as length and character types, and may include a password health check feature.

Breach Alerts are notifications that inform a user or organization of a data breach, which could put the stored passwords and sensitive information in the password manager at risk. In the context of a password manager, a breach alert means users should change their passwords immediately and consider updating the security measures used to protect their password manager account. The password manager should have the capability to detect potential breaches and notify users in real-time to prevent damage.

Independent Audits evaluate the security and privacy measures of an organization, service, or product by a third-party organization. In the context of password management, this would involve evaluating the encryption algorithms, data storage and transmission methods, and overall security architecture of the password manager. The results of an independent audit can provide users with an objective assessment of the security and privacy features of a password manager and help organizations evaluate the security of their password management systems.

A public bug bounty program invites security researchers and ethical hackers to identify and report vulnerabilities in a company's software in exchange for a reward. This type of program can help improve the security and privacy of user data stored in a password manager by encouraging third-party experts to report any vulnerability that could be exploited by malicious actors. A public bug bounty program can also demonstrate the company's commitment to the security and privacy of their software, increasing customer confidence.

Past vulnerabilities in password managers refer to security flaws or weaknesses discovered in earlier versions of the software. This can lead to the loss of sensitive information such as login credentials and personal data. Present vulnerabilities refer to security flaws or weaknesses currently present in the software and can be discovered through various means, including security audits and reports from users. The implications of these vulnerabilities can result in the loss of sensitive information and identity theft. It is important for users to stay informed about the latest vulnerabilities in their password manager and take action to address them promptly.

The storage methods used by password managers can also impact the security and privacy of user data. Some password managers store data on a server controlled by the company, while others allow users to store their data locally on their own devices. Storing data on a server controlled by the company can make it easier for the company to maintain and secure the data, but it can also make the data more vulnerable to cyber-attacks and data breaches. On the other hand, storing data locally on the user's device can provide more control over the security of the data, but it can also make it more difficult for the user to access their data from multiple devices.

Another aspect to consider is whether the password manager is open-source or not. Open-source software is code that is publicly available for anyone to inspect, modify, or distribute. Open-source password managers can provide a higher level of transparency, as the code can be inspected and reviewed by security experts. However, open-source password managers can also be more vulnerable to security issues, as the code is publicly available and can be more easily exploited by malicious actors. On the other hand, proprietary password managers, which have their source code proprietary and only accessible to the company, can provide a higher level of security, as the code is not publicly available and can be more easily protected.

Choosing a password manager involves considering a number of important factors, which includes the results of encryption method used, MFA/Biometric Authentication presence, strong password generation capabilities, breach alerts, independent audits, the presence of public bug bounty programs, the history of past and present vulnerabilities, the availability of IP whitelisting and tracking features, the storage methods used, and whether the software is open-source or proprietary. By taking the time to consider these factors and make an informed decision, users can help to ensure the security and privacy of their sensitive information and protect themselves from potential data breaches and other security risks.

Password is a password management solution that employs AES-256 encryption for safeguarding user data and 2-Secret Key Derivation (2SKD) to ensure secure access to the account. A master password is only one of the two required secrets, while the other is a randomly generated, cryptographic string [5]. 1Password supports multi-factor authentication (MFA) through Authy or Microsoft Authenticator and the Android version of the app supports biometric authentication. The password generator generates random passwords with an option to select the length, letters, numbers, symbols, and characters used. The password generator can also generate memorable passwords and pins. It features “Watchtower,” which checks if any of the user’s passwords have been detected in password dumps [6]. The company has undergone multiple security audits, assessments, and penetration tests, with the most recent audit by Onica showing no high-risk issues. However, the most recent penetration test by Cure53 showed two medium-risk threats and that the 1Password Vault was vulnerable to compromise [7]. 1Password offers a public bug bounty program for vulnerabilities in its website, sign-up procedure, authentication, and in-app features, however, the program does not compensate for bugs resulting from scheduled infrastructure changes, header issues for session management, or exploits that necessitate elevated access like DDoS/DoS attacks. As of now, 1Password has two known CVEs, one for vulnerability in the SCIM Bridge platform that allowed for viewing of the TLS private key for internet connections [8] and another for the use of insecure RNG to generate keys [9]. The capability of IP whitelisting is available in 1Password for businesses; however, it is not a feature in the standard version of the password management tool as shown in Table 4.1.

Bitwarden is an open-source password manager that provides secure storage for all vault-related information. The data is safeguarded through the use of AES-256 encryption, and the AES keys are generated from the master password using the SHA256 algorithm. The encryption key for unlocking the data is kept in the system only when the Bitwarden app is accessible. The platform offers multifactor authentication options for both standard and premium members, which include an authenticator app, email, Duo Security, VubiKey, or FIDO U2F [10]. Both the Android and iOS apps support biometrics, and users can configure the feature in the settings. Bitwarden has a password generator that can generate passwords with lengths ranging from five to 128 characters and allows users to customize the style and characteristics of the password. Bitwarden offers a “Data Breach Report” function, which uses HavelBeenPwned [11] to identify password dumps and is available on the free version for checking one password at a time. Bitwarden undergoes regular independent audits, and results are posted in a public security assessment report [12]. Bitwarden operates a bug bounty program through the HackerOne platform, covering any vulnerability that may impact its products. Bitwarden has two previous CVEs, including a server-side request forgery issue from 2020 and a potential KDF

Table 4.1 Comparison of different popular password manager

Tool	2FA/ MFA support	Independent audit	Open source	Encryption standard	Self-hosting support	IP whitelisting	Public bug bounties	Class	Password generation options	Mobile app biometrics	Breach alerts	Present/past vulnerabilities	Privacy
1Password	✔	✔	**X**	AES256	**X**	**X**	✔	Web, Mobile	Length of 0–100 with letters, numbers and symbols	✔	✔	✔	1 h
BitWarden	✔	✔	✔	AES256	✔	**X**	✔	Desktop, Web, Mobile	Length of 5–128 with letters, numbers and symbols	✔	✔	✔	12 h
NordPass	✔	✔	**X**	XChaCha20	✔	**X**	**X**	Desktop, Web, Mobile	Length of 8–60 with letters, numbers and symbols	✔	**X**	**X**	16 h
Zoho Vault	✔	**X**	**X**	AES256	**X**	**X**	✔	Web, Mobile	Length of 4–100 with letters, numbers and symbols	✔	**X**	**X**	9 h

(continued)

Table 4.1 (continued)

Tool	2FA/ MFA support	Independent audit	Open source	Encryption standard	Self-hosting support	IP whitelisting	Public bug bounties	Class	Password generation options	Mobile app biometrics	Breach alerts	Present/past vulnerabilities	Privacy
LastPass	✔	X	X	AES256	X	X	✔	Web, Mobile	Length of 0–99 with letters, numbers and symbols	✔	X	✔	16 h
KeePass	X	✔	✔	AES TwoFish XChaCha	✔	X	X	Desktop, Mobile	Variable length with letters, numbers and symbols	✔	X	✔	X
Keeper	✔	X	X	AES256	X	X	✔	Web, Mobile	Variable length with letters, numbers and symbols	✔	✔	✔	½ hour
Enpass	X	✔	X	AES256	X	X	X	Web, Mobile	Variable length with letters, numbers and symbols	✔	✔	✔	5 Days

(continued)

Table 4.1 (continued)

Tool	2FA/ MFA support	Independent audit	Open source	Encryption standard	Self-hosting support	IP whitelisting	Public bug bounties	Class	Password generation options	Mobile app biometrics	Breach alerts	Present/past vulnerabilities	Privacy
iCloud Keychain	✓	X	✓	AES256	X	X	✓	Mobile	Variable length with letters, numbers and symbols	✓	X	✓	2 Weeks
Dashlane	✓	X	X	AES256	X	X	✓	Web, Mobile	Length of 4–40 with letters, numbers and symbols	✓	X	✓	X
LogMeOnce	✓	X	X	AES256	✓	X	✓	Desktop, Web, Mobile	Length of 6–50 with letters, numbers and symbols	✓	X	X	X
RoboForm	✓	X	X	AES256	✓	X	X	Desktop, Web, Mobile	Variable length with letters, numbers and symbols	✓	X	✓	X
Samsung Pass	✓	X	X	AES256	X	X	✓	Mobile	X	✓	X	✓	X

vulnerability in 2019 [13]. There is also possible remote code execution vulnerability through the auto-update feature [14]. The platform allows for self-hosting through local storage or Docker and hosts all its code on Github.

NordPass is a password manager that uses XChaCha20 encryption [15] to protect the user's passwords. It has a zero-knowledge architecture, meaning that the encryption keys are not stored on the developer's infrastructure, but instead, kept locally on the user's device [16]. The master password is not stored anywhere and can only be unlocked by the user. NordPass offers a range of MFAoptions, including Google Authenticator, Duo, and Authy, and also supports fingerprint authentication on its mobile app. The password manager also has a password generator that can create a password between 8 and 60 characters, with the option to include letters, numbers, and symbols, or avoid ambiguous characters. The premium version of NordPass offers a "Data Breach Scanner" feature that checks password dumps against the user's stored information [17].

The results of an independent audit by Cure53 are available for viewing by Nord account holders [18]. NordVPN, the company behind NordPass, operates a bug bounty program that does not extend to the NordPass infrastructure. At present, there have been no reported vulnerabilities or exploits affecting NordPass. However, NordVPN has previously been the target of CVEs like public exploits that involve code execution and elevated local privilege [19]. NordPass allows for local hosting through a password database and does not require a Nord account to use.

Zoho Vault employs the AES-256 encryption for all confidential information and does not keep the master password in its storage. The data is sent over the internet in AES encrypted form and is protected by TLS with strong ciphers for all connections [20]. Zoho Vault provides an added layer of security with its multi-factor authentication options including voice call, text message, Zoho OneAuth, Google Authenticator, and Yubikey. The mobile app of Zoho Vault also supports biometric authentication through Swift Login setting [21]. The password generator in Zoho Vault creates passwords of length four to 100 characters, with options for numbers, special characters, letters, starting with a letter and mixed case letters. Zoho has a self-hosted bug bounty program that covers all Zoho-branded products and applications, including Zoho Vault [22]. IP whitelisting [23] is supported in the standard, professional, and enterprise versions, but not in the free version.

LastPass is a password manager that aims to secure users' online accounts and personal information. The platform uses a master password to generate AES-256 keys, which are then hashed multiple times using the PBKDF2 SHA256 algorithm. After this process, the master password is further hashed and stored as an authentication hash [24]. LastPass provides multi-factor authentication in its free plan with a variety of options, such as Duo Security, Google Authenticator, LastPass authenticator, Microsoft Authenticator, Grid, and Toopher. The mobile app of LastPass offers biometric authentication, which can be set up right after installation. The user is prompted to scan their fingerprint to enable biometrics in the security settings. The platform also offers biometric account recovery [25]. The password generator offered by LastPass allows users to choose the length of the password, which ranges from 0–99 characters and there are three options available, "easy to say", "easy to read", and "all characters". LastPass also offers Dark Web Monitoring as a feature for its premium and above versions. This feature alerts users via email if their email addresses are detected in password dumps from Enzoic [26]. The platform does not have an independent audit and has a public bug bounty program that covers the product, browser extensions, desktop applications, and mobile applications. However, there are certain areas of security that the platform does not cover, such as two-factor authentication mobile apps, information leaked through memory dumps, desktop applications compromised by malicious software or browser extensions, and man-in-the-middle (MITM) attacks [27]. LastPass has six past CVEs [28], two of which are from 2020 and are disputed due to the vulnerability relying on a jailbroken device. LastPass, as a paid password manager, does not support IP whitelisting natively but offers it through its paid 'Identity' service. The Android application of LastPass has seven tracking features, which raises privacy concerns.

KeePass is an open-source, free-ware password manager that uses AES, TwoFish, and ChaCha20 algorithms to encrypt usernames, passwords, and notes, and SHA256 to hash the master password for authentication [29]. The Android version of KeePass, although not officially supported, provides the option of biometric authentication [30]. KeePass features a password generator with user-defined length and an extensive symbol set that includes letters, numbers, and special characters, as well as the ability to create passwords based on customizable patterns. However, it does not have a feature for breach alerts. However, a security audit of KeePass was carried out in 2016 by the Free and Open-Source Software Auditing project of the European Commission, with the full results made public. The audit discovered five medium-level issues, but no critical ones. It is unknown if the European Union's plan to fund KeePass for a bug bounty program in January 2019 was carried out, as there is no recent information available on the matter [31]. KeePass has seven past CVEs [32], with two from 2020 that could lead to data reading or modification, and two public denial of service exploits. It also provides the option for users to host the password manager themselves or use a hosting service such as Dropbox.

Keeper is a paid password management solution that utilizes AES-256 encryption to secure passwords and other data. This encryption is performed at the device level before being transmitted to the servers. The platform uses a zero-knowledge architecture, meaning the master password and data are not stored in plaintext form, and AES keys are generated from the user's master password [33]. Keeper offers multifactor authentication through various methods like text messages, Microsoft/Google Authenticator, Duo Security, Yubikey, etc. and also biometric authentication for iOS (Touch/Face ID) [34] and Android (fingerprint). Additionally, the app provides a password generator for generating random passwords with letters, numbers, and symbols. The BreachWatch feature for breach alerts is not included in the free version but is available as a paid business add-on [35]. Keeper claims to undergo regular audits by firms such as NCC Group, Secarma, Rhino Security, and Cybertest, and has enacted a bug bounty program through Bugcrowd. However, no present or past vulnerabilities were noticed during research. Whitelisting specific IP addresses is not available in the free version of Keeper, but it can be established using Active Directory in the business version of the platform.

EnPass secures passwords and data with AES-256 encryption, which is applied locally on the device. The encryption key is generated from the master password, which goes through 100,000 rounds of PBKDF2-HMAC-SHA512 [36]. The password manager supports biometric authentication through its mobile app [37] and has a comprehensive password generator that allows users to generate passwords with specific criteria, such as length and number of special characters, numbers, and uppercase letters. EnPass includes a feature that allows it to compare the passwords saved in its vault against the HaveIBeenPwned database and notify users if any of their passwords have been compromised [38]. EnPass has undergone a security audit by VerSprite in 2018, and the results showed a medium overall risk impact, with two vulnerabilities identified. One vulnerability was in the Windows desktop application and another was found in the Android app which disclosed the unencrypted master password. Additionally, EnPass has been affected by two previous security issues, including a local file inclusion attack in 2017 and a code injection vulnerability in 2020 [39]. The Keychain password manager uses two separate sets of AES-256 encryption keys, the table key and the per-row key, to secure its data [40]. This technology has been developed by Apple and is open-source. Keychain uses a combination of AES-256 encryption and Apple's "Secure Enclave" to secure the data stored in it. The table key is cached for improved performance, while the per-row key is protected by the "Secure Enclave" [41]. Keychain requires MFA with Apple ID and supports touch or face ID for authentication [42]. It has a password generator with adjustable length and options. There is no mention of breach alerts or a public bug bounty program, and no independent security audits have been conducted. Keychain has had several CVEs related to obtaining items, with the latest one reported in 2018 [43]. The source code for Keychain is available on Apple's open-source subdomain, Apple Public Source License (APSL), for developers to review and contribute to its development.

Dashlane uses AES-256 encryption for securing passwords and employs Argon2D to generate the AES keys. It does not store the master password and deals only with AES encrypted data. The desktop application provides the option to enable multi-factor authentication, and biometric authentication can be configured through security settings [44]. Dashlane's password generator offers a length range of 4–40 characters and includes letters, numbers, and symbols. The premium versions of Dashlane offer "Dark Web Monitoring" which monitors up to five email addresses for password breaches [45]. Dashlane claims to have regular security audits but there is very less publicly available information about them [46]. They have a public bug bounty program hosted on Hackerone that covers autofill/autologin, the website, API endpoints, client applications, and standalone extensions. In the past, Dashlane had a vulnerability related to DLL hijacking [47], but it has since been resolved. Dashlane is proprietary software, but parts of its code can be accessed through its active Github repository.

LogMeOnce is a paid password manager that claims to use AES-256 encryption to secure its users' passwords. However, detailed information about its encryption method is not readily available and is only available in the administrator's package. LogMeOnce offers a range of multi-factor authentication options, including voice call, Selfie 2FA, TOTP, SMS, email, X.509 certificate, USB flash drive, and security key. It also has the option for biometric authentication via fingerprint scanning. The password generator creates passwords with a length ranging from six to fifty characters, composed of letters, numbers, and symbols. LogMeOnce's password generator includes an estimation of the time needed to crack the password after hashing, giving users a clear idea of its strength [48]. Breach alerts such as monitoring for leaked passwords, dark web monitoring, and anti-theft protection are available but can only be purchased. LogMeOnce has a public bug bounty program with lower reward amounts compared to other password managers, with a maximum reward of fifty dollars [49]. The account can be frozen and access can be blocked from other IPs [50], and storage modes can be changed between local and cloud storage.

RoboForm uses AES-256 encryption to secure passwords and data, and the encryption keys are generated from the master password and managed by RoboForm [51]. This password manager requires a paid subscription for access to its features. RoboForm offers MFA options through text message, email, or Google Authenticator, and biometric authentication through its mobile app and Windows Hello. The password generator in RoboForm can generate passwords with a variable length, controlled by a text box, and includes letters, numbers, and symbols [52]. No information was found regarding breach alerts or a public bug bounty program, and researchers have found vulnerabilities in RoboForm, including PIN bruteforcing and clipboard data theft [53]. RoboForm also offers IP whitelisting in its business version and allows local-only storage in the desktop and mobile applications [54].

Samsung Pass protects sensitive data with Samsung Knox's encryption standard that's referred to as "military-grade." This system uses "Dual Data-at-Rest" to encrypt the data twice, utilizing two different keys, one being AES-256 encryption [55]. The inner encryption layer can be customized with a third-party cryptographic module. The National Information Assurance Partnership (NIAP) has certified Samsung Knox's encryption framework [56]. Samsung has a bug bounty program for its mobile devices, including Knox, but some vulnerabilities are excluded and the definition of "low probability of exploitation" isn't specified. In 2019, an exploit was found that allowed an attacker with physical access to retrieve sensitive data from the Samsung Knox secure folder [57].

Background

Working Methodology of Existing Solutions

It is not just the features that are important in a password manager, but also how it operates in the background. The process of storing and transferring user credentials is crucial in ensuring security and protecting against potential threats. During the transfer of data to the cloud, a Man-In-The-Middle attack can occur, and the lack of encryption in plaintext storage leaves user credentials open to theft. Additionally, unsecured apps and Java scripts on devices can be used to access password manager databases.

To better understand the security measures used by password managers, we will delve deeper into the workings of some of the most reliable password managers that have established their reliability over time.

BitWarden

BitWarden utilizes PBKDF2 to enhance the security of a user's Master Password by combining it with a salt derived from their email address to generate a 256-bit Master Key. This key undergoes additional strengthening to 512 bits through HKDF. Unlike other password managers, BitWarden does not store or send the Master Key to its servers. Instead, it transmits an AES-256 encrypted Protected Symmetric Key using the Stretched Master Key and an Initialization Vector. Additionally, when creating an organization, a RSA key pair and an encrypted Organization Symmetric Key with the user's Symmetric Key are generated. For authentication, a hash of the Master Password is sent to the servers during account creation and login [10].

When logging in, BitWarden requires the user to input their Email Address and Master Password as shown in Fig. 4.1. The latter is then transformed with PBKDF2 and the salt of the email to create the 256-bit Master Key. The hash of this key is sent to the server to authenticate the user, and then further strengthened to 512 bits with HKDF. All decryption and Vault Item retrieval takes place on the BitWarden client using the decrypted Protected Symmetric Key and the Symmetric Key, ensuring the Master Password or Stretched Master Key is never stored or transmitted to BitWarden's servers [10].

LastPass

LastPass employs advanced security measures to protect its users' data. When logging into the password manager, one must enter their username and a Master Password which is utilized to verify the account and unlock saved credentials. The data stored in the vault is encrypted locally and not on LastPass servers, and sensitive information is transmitted securely to prevent unauthorized access as shown in Fig. 4.2. The encryption utilized by LastPass is AES 256-bit, which is a military-grade encryption in Cipher Block Chaining (CBC) mode and is generated with a key created from each user's Master Password. The Master Password is transformed into a hash using PBKDF2-SHA256 and Scrypt and is then sent to the server for verification, but not in its original form. The encryption key and Master Password never leave the user's device, making it impossible for LastPass to access or reverse them [24].

The encryption key and login hash are created on the user's device through thousands of rounds of PBKDF2 SHA-256, making it extremely difficult for a computer to hack the Master Password. The login hash is then transmitted to the server for verification. LastPass implements PBKDF2 server-side as well to ensure maximum protection of both the locally stored and server-stored data [24].

In summary, LastPass employs robust encryption techniques and numerous rounds of PBKDF2 SHA-256 to create a secure login hash, making it almost impossible for

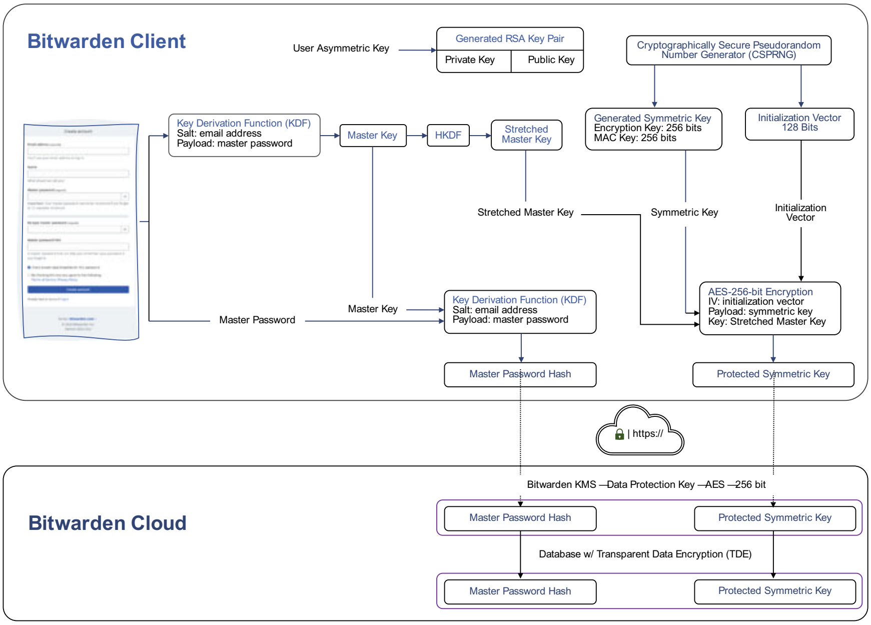


Fig. 4.1 BitWarden encryption and decryption procedure

anyone to access the sensitive information stored in the user's account. However, the high number of rounds utilized for encryption may cause slower login times for users using outdated browsers such as Internet Explorer.

1*Password*

The security of 1Password lies not only in the encryption algorithms and protocols used but also in the proper generation, management, and protection of the keys used for encryption. 1Password secures its encryption by using strong key management practices. The encryption algorithm used is RSA-OAEP, with a 2048-bit modulus and a public exponent of 65,537. To generate the keys, the client employs Cryptographically Secure Pseudo-Random Number Generators (CSPRNGs). These keys are further secured through two-secret key derivation (2SKD), which helps prevent brute-force attacks [5].

The keys are derived from the user's account password and a Secret Key. This process starts by normalizing the account password and preparing a salt using the lowercase version of the user's email address. The password is then processed through the PBKDF2-HMAC-SHA256 hash function, which has been chosen for its efficiency across various clients. However, the slow performance in JavaScript in web browsers limits the use of more advanced password hashing schemes [5].

The Secret Key is combined with the account password to generate an intermediate key, and then the authentication key is derived in a similar manner but with a different

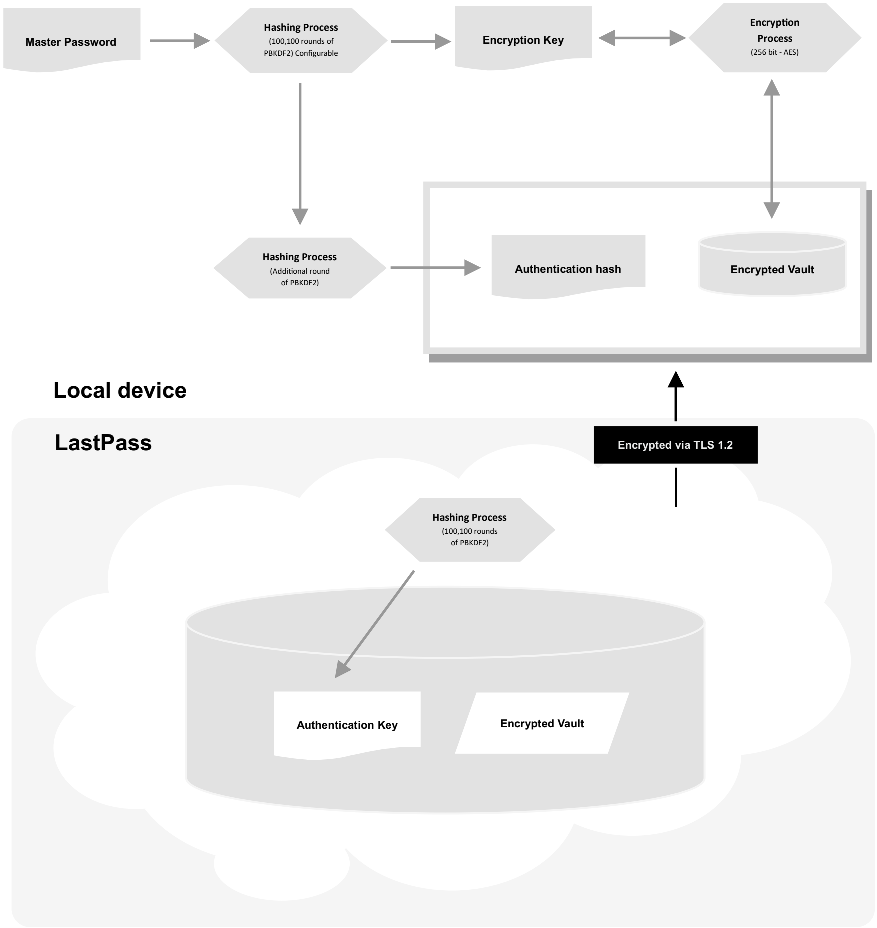


Fig. 4.2 LastPass password storage and synchronization procedure

salt for the PBKDF2 rounds. The resulting 32-byte key is converted for use with the SRP protocol [5].

To create a secure authentication process, a separate salt is utilized for the PBKDF2 rounds when deriving the authentication key, which differs from the method used for generating the Account Unlock Key (AUK). This results in a 32-byte key that is transformed into a BigNum format for use with SRP. Different tools are utilized depending on the platform—JSBN library in browsers and OpenSSL on other platforms. To derive the keys, 200,000 rounds of PBKDF2 are required, while an attacker only needs 100,000 rounds per attempt, leading to a minor advantage. Nonetheless, the sequence is infrequently performed, and the SRP-x is often stored locally or encrypted with the AUK. The client only needs to go through the derivation process during the initial sign-up or when enrolling a new client [5].

During the process of adding a new device, the user supplies the new device with both the “add-device” link and their account password. The “add-device” link is generated by an already enrolled device and contains information such as the team domain name, user’s email, and a secret key. The link uses the “onepassword:” format, with fields for email, server, and key included in the query string. The new device does not have its own unique salts or key derivation parameters, so it must request them from the server. Upon successful authentication, the device obtains the encrypted personal key set, including a private key, public key, and symmetric key used to encrypt the private key, all encrypted using the AUK and specific parameters and a salt [5].

NordPass

NordPass Business utilizes encryption technology to guarantee the protection of user data as shown in Fig. 4.3. All data belongs to the organization, and if an employee leaves, the data remains within the organization through the use of public-key cryptography. Each user has a unique key pair, and their private key is encrypted using a secret key on their device. The encryption process utilizes Argon2id for the derivation of the Master Key, XChaCha20-Poly1305-IETF for secret-key cryptography, and X25519-XSalsa20-Poly1305 for public-key cryptography [16].

The private key is only stored in plain text on the user’s device temporarily and is encrypted using the Master Key, which is generated from the Master Password and a unique salt. The app stores the unencrypted private key in secure memory while the app is unlocked and deletes it when the app is locked. Every item in the app has both metadata and secret data, allowing for more precise control of permissions [16].

Every item can be accessed through two methods: Direct access flow and the common flow. The Direct access flow is utilized when an item is shared with a user. The user is asked to enter their Master Password, which, together with the unique per-user cryptographic 16-byte salt, is used to derive the Master Key through the Argon2id key derivation function. The Master Key is then used to decrypt the user’s private key [16].

The user’s encrypted private key is then decrypted locally on their device using the XChaCha20-Poly1305-IETF algorithm and the Master Key as the decryption key. Since every item has both metadata and secret data, the user’s private key is used to decrypt either the item’s metadata private key or secret data private key, depending on the permissions granted to the user [16].

The private key of the item’s asymmetric key pair is then used to decrypt the item’s symmetric key through the xSalsa20 algorithm. Finally, the symmetric key is used to decrypt either the item’s metadata or secret data using the XChaCha20-Poly1305-IETF algorithm [16].

RoboForm

RoboForm requires new users to establish a Master Password, which serves as the solitary password required to access saved information both locally and online as shown in Fig. 4.4. To secure the data, the company implements two distinct cryptographic functions to create the symmetrical key used for local encryption/

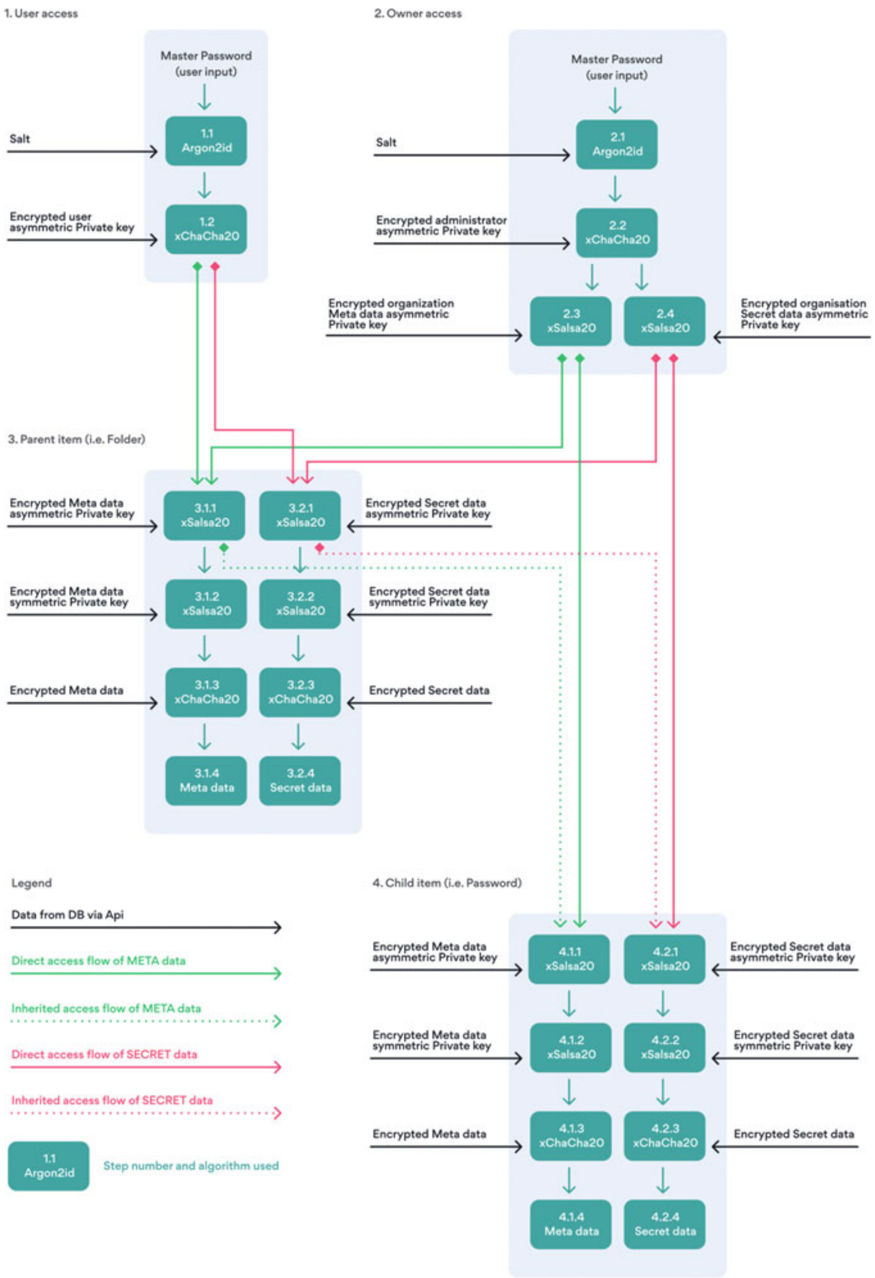

1. User access
Master Password (user input)
Salt
1.1 Argon2id
Encrypted user asymmetric Private key
1.2 xChaCha20
2. Owner access
Master Password (user input)
Salt
2.1 Argon2id
Encrypted administrator asymmetric Private key
2.2 xChaCha20
Encrypted organization Meta data asymmetric Private key
2.3 xSalsa20
2.4 xSalsa20
Encrypted organisation Secret data asymmetric Private key
3. Parent item (i.e. Folder)
Encrypted Meta data asymmetric Private key
3.1.1 xSalsa20
3.2.1 xSalsa20
Encrypted Secret data asymmetric Private key
Encrypted Meta data symmetric Private key
3.1.2 xSalsa20
3.2.2 xSalsa20
Encrypted Secret data symmetric Private key
Encrypted Meta data
3.1.3 xChaCha20
3.2.3 xChaCha20
Encrypted Secret data
3.1.4 Meta data
3.2.4 Secret data
Legend
Data from DB via Api
Direct access flow of META data
Inherited access flow of META data
Direct access flow of SECRET data
Inherited access flow of SECRET data
1.1 Argon2id
Step number and algorithm used
4. Child item (i.e. Password)
Encrypted Meta data asymmetric Private key
4.1.1 xSalsa20
4.2.1 xSalsa20
Encrypted Secret data asymmetric Private key
Encrypted Meta data symmetric Private key
4.1.2 xSalsa20
4.2.2 xSalsa20
Encrypted Secret data symmetric Private key
Encrypted Meta data
4.1.3 xChaCha20
4.2.3 xChaCha20
Encrypted Secret data
4.1.4 Meta data
4.2.4 Secret data

Fig. 4.3 NordPass password storage and synchronization procedure

decryption and server-side password protection. These functions each utilize a different, user-specific “salt” that is randomly generated [51].

To produce the AES encryption key, RoboForm employs the PBKDF2 algorithm and the SHA-256 hash function, as well as a long random salt of 32 bytes. This process only occurs on the user’s device, as the company does not perform any encryption or decryption on the server. Furthermore, user information is never transmitted to the server in an unencrypted format, with all communication between RoboForm clients and the server conducted through secure channels [51].

The Master Password created during the account creation process is the only password that the user is required to remember, and it serves as the key to unlocking their RoboForm data. To enhance security, two separate cryptographic functions are used to generate the symmetrical key for local encryption/decryption and server-side password protection, each with its own unique user-specific “salt” [51].

The AES encryption key is generated by utilizing the PBKDF2 (Password-Based Key Derivation Function 2) algorithm and the SHA-256 hash function, along with a long random salt. The number of iterations within PBKDF2 offers protection against brute force and dictionary attacks; however, it can also slow down the algorithm, particularly on slower devices or applications. To counteract this, it is recommended to increase the length of the password instead of the number of iterations [51].

Only the password hash derived from the Master Password is shared with the RoboForm server, and it is impossible to recover the Master Password or AES-256 key from this hash. This added level of security ensures that the user’s data remains confidential and protected [51].

Keeper

Keeper Security utilizes a Zero-Knowledge Architecture, a security system known for its superior privacy and protection features. It is based on several key principles to

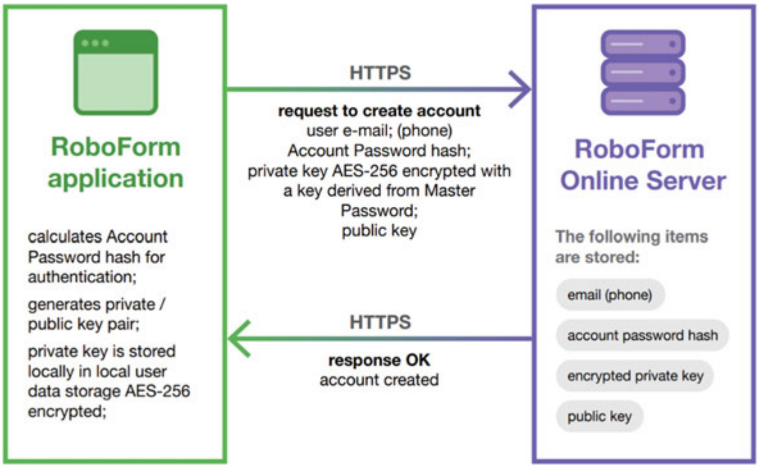

RoboForm application
calculates Account Password hash for authentication;
generates private / public key pair;
private key is stored locally in local user data storage AES-256 encrypted;
HTTPS
request to create account
user e-mail; (phone)
Account Password hash;
private key AES-256 encrypted with a key derived from Master Password;
public key
HTTPS
response OK
account created
RoboForm Online Server
The following items are stored:
email (phone)
account password hash
encrypted private key
public key

Fig. 4.4 RoboForm password storage and synchronization procedure

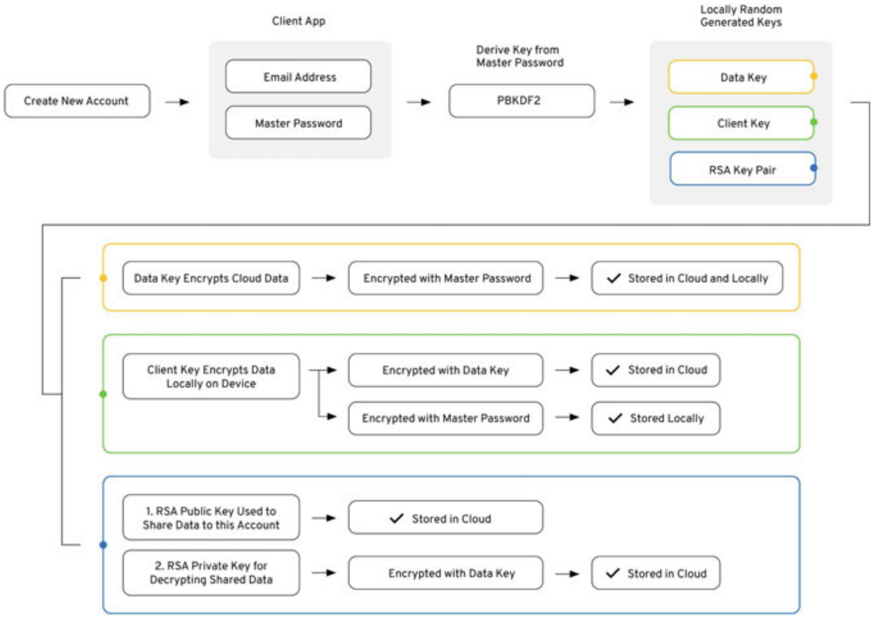

Create New Account
Client App
Email Address
Master Password
Derive Key from Master Password
PBKDF2
Locally Random Generated Keys
Data Key
Client Key
RSA Key Pair
Data Key Encrypts Cloud Data
Encrypted with Master Password
✓ Stored in Cloud and Locally
Client Key Encrypts Data Locally on Device
Encrypted with Data Key
✓ Stored in Cloud
Encrypted with Master Password
✓ Stored Locally
1. RSA Public Key Used to Share Data to this Account
✓ Stored in Cloud
2. RSA Private Key for Decrypting Shared Data
Encrypted with Data Key
✓ Stored in Cloud

Fig. 4.5 Keeper encryption model

maintain the security of user data. Firstly, the data is only decrypted and encrypted on the user's device and never on the server. This means that the application never saves the data in a readable format and the server never receives unencrypted information. Furthermore, neither Keeper employees nor any third parties have access to the unencrypted data as shown in Fig. 4.5. The keys used to encrypt and decrypt data are derived from the user's Master Password [58].

In addition, Keeper employs multiple layers of encryption, providing control access at the user, group, and administrator levels. Sharing of information is also secured with the use of Public Key Cryptography, which ensures safe key distribution. Keeper is committed to maintaining the highest levels of security and privacy and has received certifications such as SOC 2, ISO 27001, and is compliant with the EU-U.S. Privacy Shield program. Regular audits are also conducted to guarantee the continuous development of secure software. To use Keeper, a unique Master Password must be chosen by the user. The Zero-Knowledge Architecture guarantees that no one, including Keeper employees and administrators, has access to this password. The administrator can set guidelines for the Master Password and in the event of a lost password, the user can recover their account through a secure process that involves a security question, email verification, and two-factor authentication.

Keeper uses encryption to ensure the safety of user data, which is why it is recommended by the National Institute of Standards and Technology and the European Union's General Data Protection Regulations. The company implements symmetric encryption to store passwords in an encrypted form in a digital vault. The encryption key to access the vault is derived from the user's Master Password and all encryption keys, such as the Data Key, RSA Private Key, Record Keys, and Folder Keys, are unique to the user and encrypted for extra security.

To protect the data, Keeper employs the strongest forms of encryption, including 256-bit AES and PBKDF2 for key derivation. The application uses multiple layers of encryption, including at the record, folder, and team levels. This allows for records to be shared among authorized users without risking unauthorized access. The encrypted vault is stored in the cloud for synchronization and can also be used offline, but the Keeper Administrator can restrict offline access. Data in transit is protected with 256-bit TLS/SSL encryption, further secured by Key Pinning and encryption layers to prevent man-in-the-middle attacks [58].

Dashlane

Dashlane takes protecting user data very seriously and employs four distinct secrets to ensure the security of this information. The first of these secrets is the User Master Password, which is not stored on Dashlane servers or any of its affiliates, including hashes as shown in Fig. 4.6. By default, the Master Password is not saved on the device, but instead is used to encrypt and decrypt local data files. However, if the user wants to, they can opt to save the Master Password locally through the "Remember my Master Password" feature. The Master Password is never transmitted online [44].

In certain circumstances, an Intermediate Key encrypted with the Master Password is used for local storage. A unique User Device Key is generated for each device that the user enables, automatically. This key is then used for authentication purposes. A local secret key is also generated, which is used to secure communication between the Dashlane application and browser plugins. The key exchange is completed through local visual pairing and Diffie-Hellman [44].

Access to the user's personal information is only possible with the use of the User Master Password. This password generates a 256-bit AES key which is used to encrypt and decrypt the user's data on their device. The encryption process utilizes various libraries, such as OpenSSL or Web Crypto API, depending on the device being used. When the user inputs their Master Password into Dashlane, the data is decrypted and stored in memory. However, to maintain security, the data is encrypted using AES before being transmitted through named pipes or web sockets. The encryption process includes a random 16-byte initialization vector and a salt that is written in the AES file [44].

All communications between the Dashlane app and servers are protected by HTTPS and SSL/TLS. The HTTPS connections are implemented on the client side using OpenSSL and on the server side with a DigiCert High Assurance CA-3 certificate. The best cipher and hash algorithm are negotiated between the client and server, and the server sends a digital certificate that the client verifies with a Certificate Authority. A symmetric key is then generated to encrypt and decrypt data. Communication between the Dashlane browser plugin and application is also encrypted using AES 256 encryption with the OpenSSL library, which includes a 32-byte salt, a randomly chosen 16-byte IV, and the salt included in the encryption process [44].

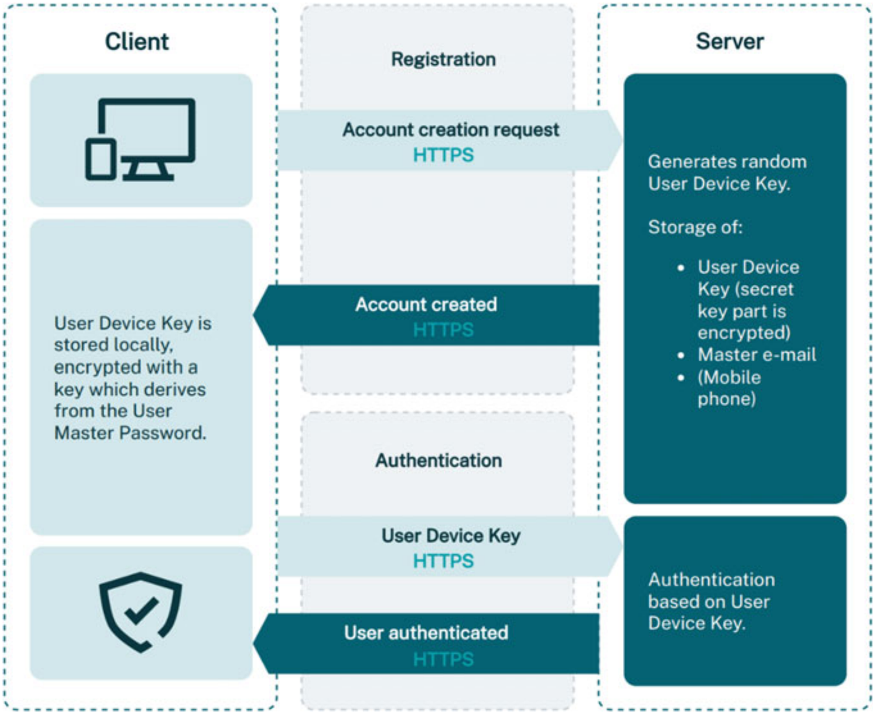

Client
Registration
Server
Account creation request
HTTPS
Generates random User Device Key.
Storage of:
• User Device Key (secret key part is encrypted)
• Master e-mail
• (Mobile phone)
User Device Key is stored locally, encrypted with a key which derives from the User Master Password.
Account created
HTTPS
Authentication
User Device Key
HTTPS
Authentication based on User Device Key.
User authenticated
HTTPS

Fig. 4.6 Dashlane authentication model

Zoho Vault

When establishing a Zoho Vault account, users must create a master password that will serve as their encryption key as shown in Fig. 4.7. The master password must be a minimum of 8 characters long, and users receive instant feedback on its strength. This password is kept confidential by the user and never stored on Zoho's servers. Zoho Vault uses the master password to generate the Key Encryption Key (KEK) through many iterations of PBKDF2 with HMAC-SHA256 Key Derivation Function, using a random salt value [21].

To ensure the security of sensitive user data, Zoho Vault employs a host-proof hosting method. This means that all data encryption and decryption take place within the user's browser on the client side. The data is encrypted using AES-256 encryption on the client side, transmitted securely over HTTPS, and stored on Zoho's servers in encrypted form. The master password set by the user acts as the encryption key, and it is never stored on Zoho's servers. When a user wants to access their data, the encrypted data is retrieved over HTTPS. When adding, deleting, or modifying data, Zoho Vault encrypts the data on the client side before sending it to Zoho's servers. Zoho's servers only hold encrypted data that can only be decrypted using the user's master password and a unique salt value. This means that even if someone were to

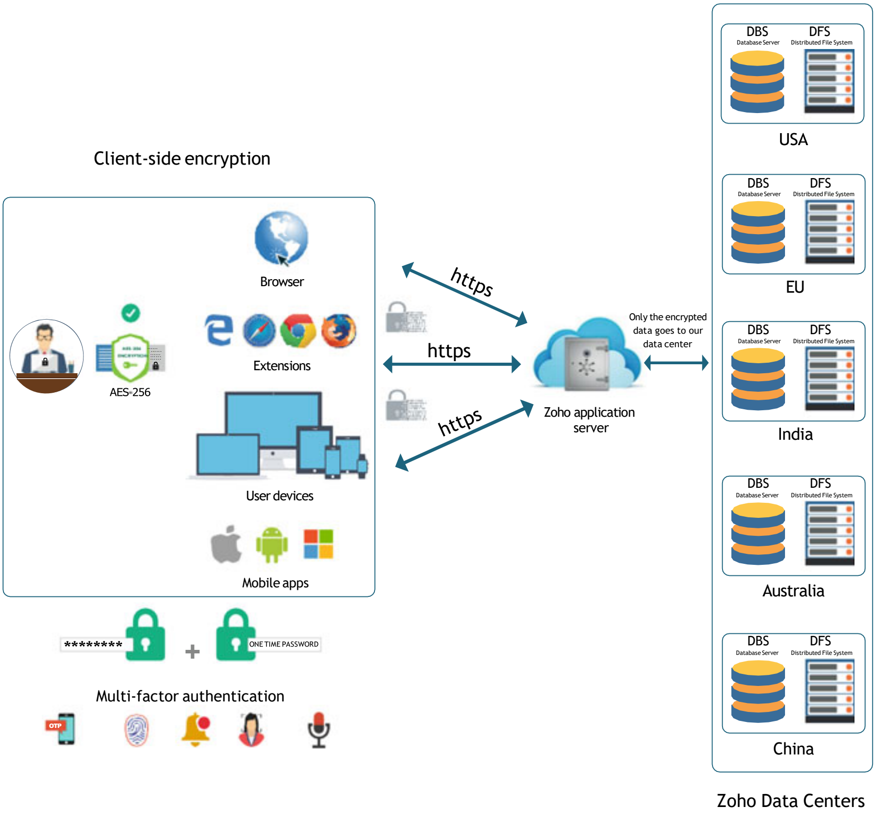


Fig. 4.7 Zoho encryption and decryption

access Zoho's servers, they would not be able to view the data in its original form [21].

The authentication process for each user involves several steps. Requests for access from web, browser extension, or mobile apps are directed to the Zoho Accounts login page. The login information is sent to Zoho's server for verification. If the user is authenticated, a cookie is set for the current browser session, and access to the Vault is granted. The Zoho Accounts agent on the application server then checks the cookie information with the Zoho Accounts server in the background [21].

Result and Analysis

Password managers are a vital resource for safeguarding online accounts from cyber threats. These tools generate strong, random passwords that enhance security against hacking attempts. Some password managers give users the choice of either retaining their current non-random passwords or generating new ones during the sign-up process. However, the efficacy of passwords decreases when the symbol set used is limited. Of the available options, KeePassXC provides the most extensive symbol set, including both normal and extended ASCII characters (excluding spaces). Other popular password managers like Dashlane, 1Password, etc. support only the standard ASCII symbol set, with a restricted number of characters.

Password storage is the second phase of the password manager lifecycle. Both app-based and extension-based password managers use AES-256 encryption to secure their databases. This advanced encryption technique provides robust protection for metadata stored in the password manager. All metadata is encrypted in all password managers, including KeePassXC.

Password managers employ various techniques to store and automatically fill in passwords. For instance, app-based password managers can copy cloud data locally and encrypt it using a master password. However, the autofill feature may pose a security risk if the password manager automatically fills in the password without seeking user interaction. To mitigate this risk, KeePassXC, Bitwarden, and RoboForm all require user interaction by default before autofill occurs. Autofilling passwords within iframes is also a potential security hazard, as attackers can acquire sensitive information through clickjacking or cross-domain iframe autofill.

The accepted method for storing password information is AES-256 encryption with PBKDF2 for transport to the cloud. All OS-based mobile autofill frameworks demand user interaction before autofill, providing a secure and accessible way to recall password data. iOS password autofill completely encrypts the autofill process for native UI components in apps, and local storage with master password encryption is considered the best approach for web extensions.

While many password managers offer unique features and advantages, they also come with limitations. LastPass is plagued with autofill issues, outdated apps, limited free version features, and a 2022 data breach that tarnished its reputation. Dashlane is limited in its password storage options, simultaneous device usage, and cloud storage. LogmeOnce, a great freemium option, is overloaded with features and has a cluttered user interface, making it overwhelming for common users. BitWarden, an open-source solution, lacks TOTP and password sharing features in its free version. KeePassXC, another open-source option, lacks password sharing and can be difficult to configure manually. 1Password, NordPass, and Keeper are leaders in the password manager space but lack free versions, making them inaccessible to those unwilling to pay for their services. RoboForm, a popular and relatively inexpensive option, lacks TOTP support for its mobile apps and cloud sync options.

Addressing the Challenges with an Innovative Solution

Our software solution aims to provide a comprehensive password management system with a graphical user interface, developed using Android Studio. This system enables users to securely store their online credentials, including usernames and passwords, along with other sensitive information, in an encrypted database. The encryption is managed using advanced algorithms such as Rijndael AES, Two-Fish, and ChaCha20. To access the system, users must login and the validity of the user is verified. Users can create and add to the database by using provided templates, and the list of accounts is organized into groups for easy access. Passwords are hidden by default but can be revealed upon selection. The software also includes a robust key-derivation function using Argon2/AES-KDF, making password handling secure.

In addition to storing passwords, our software also includes features such as zero-knowledge encryption, unlimited vault storage and synchronization, an open-source codebase, a secure password and passphrase generator, 2-factor authentication login, TOTP and HOTP support for stored credentials, and the ability to store notes, credit cards, and identities. The goal is to provide these advanced security features at a low cost to ensure that more people can benefit from proper management of their online credentials without compromising on security.

The system will be developed using Android Studio and programming languages such as Kotlin and Java, which are freely available online. The deployment platform will consist of Android devices, which are widely available, and laptops that meet the necessary requirements. The cloud server storage will be compatible with various solutions such as Google Drive, OneDrive, and Dropbox, as long as the corresponding app is installed on the user's smartphone. The only cost involved will be for internet access for downloading the required tools and for cloud storage synchronization.

Overall, this project is cost-effective as it utilizes freely available software and programming languages and only requires internet access for proper functionality. The goal is to keep the cost as low as possible to make the benefits of secure password management accessible to a wider audience.

Design Implications for Solving Problems

Zero-knowledge encryption is a method of encryption where the encrypted data is stored on the server, and the decryption key is stored locally, on the user's device. This way, the server or any third party cannot access the decryption key and therefore cannot access the original data. In other words, the server has no knowledge of the original data, hence the name "zero-knowledge." This architecture provides enhanced security for sensitive data, as the encrypted data cannot be decrypted by anyone other than the user who has the decryption key as shown in Fig. 4.8.

Our system implements comprehensive encryption, which encompasses all elements of the database, including passwords, usernames, URLs, notes, and other relevant information. The encryption technique adopted leverages AES (Rijndael), Two-Fish, and ChaCha20—widely recognized as secure encryption algorithms. The encryption is executed using the Cipher Block Chaining (CBC) mode, which conceals any patterns in the plaintext. Whenever the database is saved, a randomly generated Initialization Vector (IV) ensures the security of multiple databases encrypted with the same master key. The Encrypt-then-MAC approach confirms the authenticity

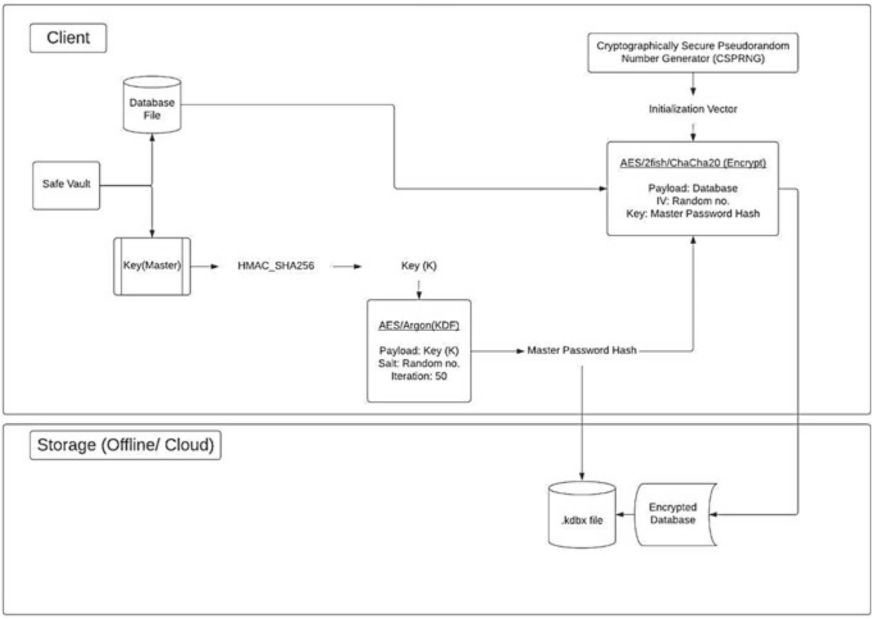


Fig. 4.8 Encryption and decryption model

and integrity of the data by creating a HMAC-SHA-256 hash of the ciphertext. This hash confirms the data's authenticity and integrity. To produce random bits for the high-level generation methods, we utilize a cryptographically secure pseudo-random number generator that is initialized using an entropy pool composed of various sources, such as the system cryptographic provider, current date/time, cursor position, operating system version, and more. The encryption master key is comprised of a combination of the master password, a key file, and/or a hardware key. The components of the master key are compressed using SHA-256, resulting in a 256-bit key (K). This key is transformed using a key derivation function with a random salt, making it difficult for attackers to carry out dictionary and guessing attacks. To further enhance security, the key derivation function (with a random salt) is utilized to prevent pre-computation of keys and make dictionary and guessing attacks more challenging. This helps to ensure the security of the encryption process.

Securing a database involves making it difficult for unauthorized users to access its contents. This is accomplished through the transformation of the user's master key into a secure key using a key derivation function that includes a random salt. The more complex the key derivation function, the more challenging it will be for an attacker to guess the key. Key hashing transforms passwords into hash values, while key derivation converts passwords into keys used for encryption and decryption. These processes defend against attempts to steal passwords by intercepting the hash or key. AES-KDF and Argon2 are examples of supported key derivation functions. The AES-KDF iterates AES and can be adjusted by the user for increased difficulty.

Argon2, the winner of the Password Hashing Competition, provides better protection against GPU/ASIC attacks, with variants Argon2d, Argon2id, and Argon2i. Argon2id is recommended for server applications as it provides better protection against side-channel attacks, while Argon2d is recommended for client devices as it offers better protection against GPU/ASIC attacks. The time required for key transformation may differ on different devices, so it's essential to ensure that all devices can load the database quickly. Aside from dictionary attacks, password managers can also be targeted by keyloggers and process memory breaches. Process memory breaches focus on attacking the memory of an application where temporary data is kept. If a password manager is compromised in this way, an attacker can access sensitive information like passwords. This can happen if the password manager is running on a device with a security flaw or if there is a vulnerability in the password manager itself. For security and efficiency, it's crucial that sensitive data like entry passwords and master keys be encrypted when stored in memory. To heighten security, the password manager will close the database and only retain the file path and certain view parameters when the workspace is locked.

Password managers typically have an auto-fill feature that sends fake keystrokes to other applications, making it challenging for the target app to differentiate between real and simulated keypresses. However, this feature can be vulnerable to keyloggers as well. Keyloggers are malicious software or hardware that track every keystroke made on a computer. If a user's password manager is breached by a keylogger, the attacker can obtain their login information and potentially access sensitive information. To address this issue, the password manager uses the device clipboard to transfer parts of the auto-typed text to the target app, making it challenging for keyloggers to monitor the process. While this adds another layer of security, it's not foolproof and can still be vulnerable to spyware specifically designed to log obfuscated auto-types. It's important to keep in mind that there is no perfect security solution and the best way to protect sensitive data is to regularly update the password manager software and follow good security practices.

Database synchronization refers to the process of keeping multiple copies of a database in sync. This is an important feature in password managers, as it allows the user to access their passwords from any device. The password manager synchronizes the encrypted database across devices, ensuring that the latest version of the database is always available on all devices. This feature provides convenience for users who need access to their passwords on multiple devices, as well as increased security, as the encrypted database is stored in the cloud and can be retrieved in case the user loses their device as shown in Fig. 4.9.

The synchronization process of a password manager is accomplished through two main stages. Firstly, the data recovery from a shared source, which is overseen by a specific application. This application is responsible for transmitting and receiving files and making them available through a URI. The password manager itself acts merely as an editor, and any issues related to synchronization may stem from the cloud-based application used, such as difficulties with file conflicts or caching issues. The second stage involves the merging of updated information. Once new data is acquired through the URI, the password manager application is capable of combining

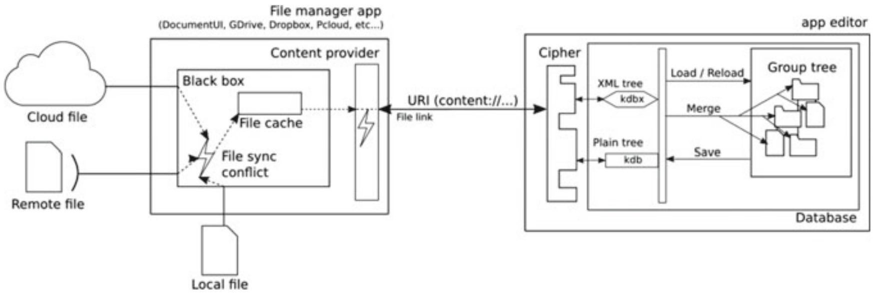


Fig. 4.9 Database synchronization model

it with the existing data. This allows the password manager to stay current and up-to-date, ensuring that users have the most accurate information at their disposal.

The password manager can provide several options for storing and synchronizing a user's password database. The database file can be stored locally on a device, on an external storage device, or on a remote file manager, such as a cloud application. The password manager acts as an editor for the database, but it is not responsible for managing data recovery or synchronization. This is typically handled by the cloud application used. The password manager can merge the retrieved data with the currently open data when a new version of the database is provided. This allows for greater scalability and configuration, but it also means that the user should be careful to choose a reliable cloud application for file management.

The main focus of the synchronization process is to decide which copy of an object is the latest one, mainly using the last modification time of the object. The synchronization process is performed at the entry level, ensuring that the combination of username and password is always consistent. In case of parallel updates and collisions, the password manager tries to store all information in an appropriate place. For example, if two users make changes to the same entry on two different devices and then try to synchronize, the password manager will consider the entry on the device with the latest modification time as the current version, while storing the changes made on the other device as a history entry. This helps to prevent loss of data, while ensuring that the user always has access to the latest information.

In order to ensure seamless data synchronization, the password manager would provide users with several options for triggering a synchronization process. Manual synchronization is the most straightforward and simple way to keep data in sync. With manual synchronization, the user must actively initiate the process every time they wish to synchronize their data. This might be a good option for users who do not require real-time data synchronization or who are concerned about data privacy and security. Another way to trigger synchronization is through the use of the "Save" command. With this option, a user can save the changes they have made to their password manager database to a remote server or cloud storage. This provides a convenient way to ensure that all data changes are automatically backed up and synced with other devices. Triggers are another way to automatically initiate synchronization. A trigger is an event that occurs within the password manager that automatically initiates a synchronization process. For example, a trigger might be set to occur every time a password is added or updated, or every time the password manager is closed or reopened. This provides an easy and efficient way to keep data in sync without requiring manual intervention. Finally, scripting can also be used to initiate synchronization. Scripting allows users to automate various tasks within the password manager, including synchronization. This is particularly useful for advanced users who require more complex data synchronization processes, or for those who wish to automate the synchronization process as part of a larger workflow.

Conclusion

Password managers serve a vital purpose in the safekeeping of our sensitive information that we manage online. They are software programs that allow individuals to create, store, and manage passwords securely. With the need to remember numerous unique passwords for various accounts and services, it can be challenging to ensure the security of this information. The occurrence of data breaches emphasizes the significance of having robust and original passwords that are difficult to crack.

There are various password managers available to users with varying features and security measures. Utilizing a password manager offers secure password storage, auto-generated passwords, password sharing options, and multi-factor authentication. The security of password managers also depends on the encryption methods, database storage techniques, and security protocols they employ.

To improve on the existing password managers, it is necessary to focus on strengthening the encryption mechanisms and database storing mechanisms. The encryption mechanism should use state-of-the-art encryption algorithms supporting post-quantum cryptography and use encryption keys that are stored locally on the device. The database storing mechanism should also be designed in a way that minimizes the risk of unauthorized access to sensitive information. Additionally, multi-factor authentication should be implemented to provide an additional layer of security. Furthermore, it is important to ensure that security methodologies such as IP whitelisting, bug bounty programs, and regular independent audits are in place to minimize the risk of security breaches.

In essence, password managers play a vital role in securing sensitive information and protecting against data breaches. Improving the existing password managers would require a focus on encryption mechanisms, database storing mechanisms, and security methodologies, to provide the best possible security to users. With the right focus and resources, it is possible to create a secure and user-friendly password manager that provides peace of mind and protects against the growing threat of cyber-attacks.

Acknowledgements We would like to express our gratitude to Jain (Deemed-to-be) University, Bangalore, for providing us with the resources and support necessary to complete this review paper. Their contribution has been invaluable, and we are grateful for the opportunities and experiences that have been provided to us.

References

1. Gallagher EA (2019) Choosing the right password manager. Ser Rev 45(1–2):84–87. https://doi.org/10.1080/00987913.2019.1611310
2. Shinde SK, Deshpande MV (2022) A study for an ideal password management system, vol 10. https://doi.org/10.22214/ijraset.2022.39970
3. Silver D, Jana S, Boneh D, Chen E, Jackson C (2023) Password managers: attacks and defenses. https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/silver. Last Accessed 05 Feb 2023. (Online)
4. Karole A, Saxena N, Christin N (2011) A comparative usability evaluation of traditional password managers. In: Lecture notes in computer science (Including subseries lecture notes in Artificial Intelligence and lecture notes in bioinformatics), vol 6829, pp 233–251. LNCS. https://doi.org/10.1007/978-3-642-24209-0_16/COVER
5. Password security design 1Password memberships (2021). https://1passwordstatic.com/files/security/1password-white-paper.pdf. Last Accessed 05 Feb 2023. (Online)
6. 1Password Watchtower. https://watchtower.1password.com/. Last Accessed 05 Feb 2023
7. Security audits of 1Password. https://support.1password.com/security-assessments/. Last Accessed 05 Feb 2023
8. CVE-2021-26905 for all versions of the 1Password SCIM bridge released prior to February 8, 2021. https://support.1password.com/kb/202102/. Last Accessed 05 Feb 2023
9. CVE-2020-10256 for all beta versions of the 1Password command-line tool and SCIM bridge released prior to December 24, 2018. https://support.1password.com/kb/202010/. Last Accessed 05 Feb 2023
10. 8bit Solutions LLC (2018) BitWarden security assessment report. ISSUE SUMMARIES, IMPACT ANALYSIS, AND RESOLUTION 8BIT SOLUTIONS LLC. https://cdn.bitwarden.net/misc/Bitwarden%20Security%20Assessment%20Report.pdf. Last Accessed 05 Feb 2023. (Online)
11. Have you been pwned? BitWarden Blog. https://bitwarden.com/blog/have-you-been-pwned/. Last Accessed 05 Feb 2023
12. Compliance, audits, and certifications. BitWarden Help Center. https://bitwarden.com/help/is-bitwarden-audited/. Last Accessed 05 Feb 2023
13. I. BitWarden (2022) Bitwarden network security assessment report. ISSUE SUMMARIES, IMPACT ANALYSIS, AND RESOLUTION, BITWARDEN, INC. https://bitwarden.com/images/resources/2022-bitwarden-network-security-assessment-report.pdf. Last Accessed 05 Feb 2023. (Online)
14. Security: BitWarden desktop app grants RCE to BitWarden developers. Issue #552 bitwarden/desktop. https://github.com/bitwarden/desktop/issues/552. Last Accessed 05 Feb 2023
15. XChaCha20 encryption. NordPass. https://nordpass.com/features/xchacha20-encryption/. Last Accessed 05 Feb 2023
16. NordVPN S.A. (2020) NordPass business whitepaper. NordPass. https://nordpass.com/nordpass-business-whitepaper.pdf. Last Accessed 05 Feb 2023. (Online)
17. Data breach scanner: check your password safety. NordPass. https://nordpass.com/features/password-breach-report/. Last Accessed 05 Feb 2023
18. NordPass completes a comprehensive security audit. NordPass. https://nordpass.com/blog/nordpass-security-audit-2020/. Last Accessed 05 Feb 2023

19. NordVPN-6.31.13.0-'NordVPN-service' unquoted service path. Windows Local Exploit. https://www.exploit-db.com/exploits/48790. Last Accessed 05 Feb 2023
20. The ultimate security for your passwords. Zoho Vault. https://www.zoho.com/vault/security.html. Last Accessed 05 Feb 2023
21. Zoho Corporation Private Limited, Zoho Vault security specifications. https://www.zoho.com/sites/default/files/zoho-vault-security-specifications.pdf. Last Accessed 05 Feb 2023. (Online)
22. Complete insights with our visual reports. Zoho Vault. https://www.zoho.com/vault/vault-security.html. Last Accessed 05 Feb 2023
23. IP restriction. Zoho Vault. https://help.zoho.com/portal/en/kb/vault/admin-guide/articles/vault-configure-enable-ip-restriction. Last Accessed 05 Feb 2023
24. LastPass technical whitepaper. https://assests.cdngetgo.com/da/ce/d211c1074dea84e06cad6f2c8b8e/lastpasstechnical-whitepaper.pdf. Last Accessed 05 Feb 2023. (Online)
25. How do I set up biometrics and mobile account recovery on Android for LastPass? LastPass support. https://support.lastpass.com/help/how-do-i-set-up-and-use-mobile-account-recovery-on-android-lp010120. Last Accessed 05 Feb 2023
26. Dark web monitoring & alerts. LastPass. https://www.lastpass.com/features/dark-web-monitoring. Last Accessed 05 Feb 2023
27. P. Berba (2016) How to decrypt a LastPass vault. This is a medium-sized extract from a … by Pepe Berba. Medium. https://medium.com/@pberba/how-lastpass-decrypts-your-vault-279153350930. Last Accessed 05 Feb 2023. (Online)
28. CVE-search results. https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=LastPass. Last Accessed 05 Feb 2023
29. Reichl D (2015) Security-KeePass. https://doi.org/10.6028/NIST.FIPS.180-4
30. KeePass gets full biometrics support in the latest Keepass2Android update. Android: gadget hacks. https://android.gadgethacks.com/how-to/keepass-gets-full-biometrics-support-latest-keepass2android-update-0331354/. Last Accessed 05 Feb 2023
31. EU to fund bug bounty programs for 14 open source projects starting January 2019. ZDNET. https://www.zdnet.com/article/eu-to-fund-bug-bounty-programs-for-14-open-source-projects-starting-january-2019/. Last Accessed 05 Feb 2023
32. CVE-search results. https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=keepass. Last Accessed 05 Feb 2023
33. Keeper encryption model-enterprise guide. https://docs.keeper.io/enterprise-guide/keeper-encryption-model. Last Accessed 05 Feb 2023
34. Login to keeper on macOS with touch ID-user guides. https://docs.keeper.io/user-guides/tips-and-tricks/login-to-keeper-on-macos-with-touch-id. Last Accessed 05 Feb 2023
35. Dark web monitoring for business-keeper security. https://www.keepersecurity.com/breachwatch.html. Last Accessed 05 Feb 2023
36. Data Security: Safeguard Sensitive Data with Enpass. https://www.enpass.io/security/ (accessed Feb. 05, 2023).
37. Quick unlock—EnPass security whitepaper documentation. https://support.enpass.io/docs/security-whitepaper-enpass/quick_unlock.html. Last Accessed 05 Feb 2023
38. Leo G, Consultant S, Watson F (2018) EnPass apps-security assessment. https://dl.enpass.io/docs/EnpassSecurityAssessmentReport.pdf. Last Accessed 05 Feb 2023
39. Vulnerability-reporting-EnPass. https://www.enpass.io/vulnerability-reporting/. Last Accessed 05 Feb 2023
40. Keychain data protection-Apple support (CA). https://support.apple.com/en-ca/guide/security/secb0694df1a/web. Last Accessed 05 Feb 2023
41. Secure iCloud keychain recovery-Apple support (CA). https://support.apple.com/en-ca/guide/security/secdeb202947/web. Last Accessed 05 Feb 2023
42. Accessing keychain items with face ID or touch ID. Apple developer documentation. https://developer.apple.com/documentation/localauthentication/accessing_keychain_items_with_face_id_or_touch_id. Last Accessed 05 Feb 2023
43. CVE-search results. https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=Apple+Keychain. Last Accessed 05 Feb 2023

44. I. Dashlane (2022) Dashlane's security principles & architecture. https://www.dashlane.com/download/whitepaper-en.pdf. Last Accessed 05 Feb 2023. (Online)
45. Security alerts and dark web monitoring in Dashlane–Dashlane. https://support.dashlane.com/hc/en-us/articles/360000038180-What-are-security-alerts-and-Dark-Web-Alerts-and-what-to-do-when-I-get-one. Last Accessed 05 Feb 2023
46. Gentili P, Shader S, Yip R, Zeng B (2016) Security analysis of Dashlane. https://courses.csail.mit.edu/6.857/2016/files/25.pdf. Last Accessed 05 Feb 2023. (Online)
47. Dashlane-DLL Hijacking-windows local exploit. https://www.exploit-db.com/exploits/44066. Last Accessed 05 Feb 2023
48. Why use LogMeOnce-LogMeOnce. https://www.logmeonce.com/why-use-logmeonce/. Last Accessed 05 Feb 2023
49. Vulnerability disclosure policy-LogMeOnce. https://www.logmeonce.com/vulnerability-disclosure-policy/. Last Accessed 05 Feb 2023
50. Password Shock-LogMeOnce. https://www.logmeonce.com/password-shock/. Last Accessed 05 Feb 2023
51. RoboForm Security Overview Whitepaper (2019), https://www.roboform.com/pdf/RoboForm_Security_White_Paper.pdf. Last Accessed 05 Feb 2023 (Online)
52. Key features. https://www.roboform.com/key-features. Last Accessed 05 Feb 2023
53. Zero-Day research. Fixes pending. FortiGuard. https://www.fortiguard.com/zeroday?type=zd&vendor=Roboform. Last Accessed 05 Feb 2023
54. RoboForm manual android. https://www.roboform.com/manual-android. Last Accessed 05 Feb 2023
55. DualDAR encryption. Knox platform for enterprise white paper. https://docs.samsungknox.com/admin/whitepaper/kpe/DualDAR.htm. Last Accessed 05 Feb 2023
56. General questions and information about Samsung Pass. https://www.samsung.com/us/support/answer/ANS00066601/. Last Accessed 05 Feb 2023
57. ZDI-19-515. Zero day initiative. https://www.zerodayinitiative.com/advisories/ZDI-19-515/. Last Accessed 05 Feb 2023
58. I. Keeper Security (2019) Keeper MSP technical whitepaper. https://www.keepersecurity.com/assets/pdf/Keeper-Managed-Service-Provider-Tech-WhitePaper.pdf. Last Accessed 05 Feb 2023. (Online)

Chapter 5
Review on Wi-Fi Attacks and Detection Methods

R. Harish and K. Praveen

Abstract This work summarizes various attacks performed on Wi-Fi networks and their impacts of it with mitigations. Attacks are classified based on WPA2 and WPA3. Different detection methods, including signature and anomaly-based, are used to prevent Wi-Fi attacks. Attack and detection methods are examined with a Wi-Fi Intrusion Detection system for various possible scenarios. A Wi-Fi auditing framework that can perform all Wi-Fi network-based attacks for testing the network and its features are investigated. Finally, future trends in Wi-Fi Intrusion Detection systems are studied and discussed.

Keywords Wi-Fi · WIDS · WPA2 · WPA3 · Wi-Fi attacks · 802.11

Abbreviations

Wi-Fi	Wireless Fidelity
AP	Access Point
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
WIDS	Wi-Fi Intrusion Detection System
SDN	Software Define Network
SAE	Simultaneous Authentication of Equals
WPA	Wireless Protected Access
TCP	Transmission Control Protocol
IP	Internet Protocol
MAC	Medium access control

R. Harish · K. Praveen (✉)
TIFAC-CORE in Cyber Security, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India
e-mail: k_praveen@cb.amrita.edu

R. Harish
e-mail: r_harish@cb.students.amrita.edu

D. Gountia et al. (eds.), *Information Technology Security*, Springer Tracts in Electrical and Electronics Engineering, https://doi.org/10.1007/978-981-97-0407-1_5

WLAN	Wireless Local Area Network
SYN	TCP Synchronization
ACK	Acknowledgement
ARP	Address Resolution Protocol
SSID	Service Set Identifier
DOS	Denial of Service
DDOS	Distributed Denial of Service
IOT	Internet of Things
EAPOL	Extensible Authentication Protocol over LAN

Introduction

Wi-Fi has become an integral part of our technological society. It enables us to connect to the internet and communicate with other devices. It used radio waves for sending and receiving data from other wireless devices. Intermediary devices like hubs, routers, and APs establish Wi-Fi connections. With the proper matching of the security parameters, devices can connect to the APs and the routers. Suppose a device wants to connect to the internet, it sends data to the router through the AP, and the data will be delivered to the internet (see Fig. 5.1). These processes need to be protected to maintain the CIA triad of security [1, 2]. The header information of Wi-Fi MAC contains all the required frames for establishing a connection with stations and APs. The first three fields and the last field are present in all frames with their subtypes. Other fields are optional according to its requirement. Four layer two addresses are used in the frame header: source, destination, transmitter, and receiver (see Fig. 5.2).

This work is further summarized with various attacks that can affect the regular usage of Wi-Fi networks based on the WPA2 and WPA3 security standards with the mitigations. Then the Wi-Fi Intrusion Detection System (IDS) is introduced with its need to add security over the existing methods. Various attacks and detection methods based on the TCP/IP layer are categorized and studied. Then, a Wi-Fi auditing framework is analyzed with Wi-Fi Pineapple. Finally, future trends in intrusion detection methods are discussed.

Wi-Fi Attacks on WPA2

WPA2 is the most used security standard for Wi-Fi. Though WPA3 has been released, not all Wi-Fi devices can support it due to hardware requirements. WPA2—AES/TKIP remains the most common and secured encryption standard in WPA2. WPA2 can be categorized into personal and enterprise. Personal is used in home Wi-Fi networks and small network setups. It contains an AP that provides the security

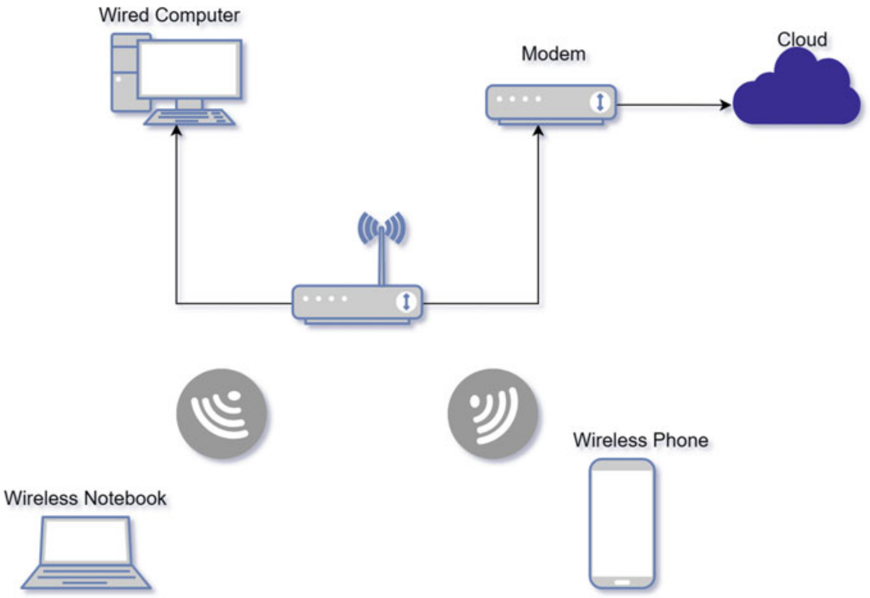


Fig. 5.1 Wi-Fi connections architecture

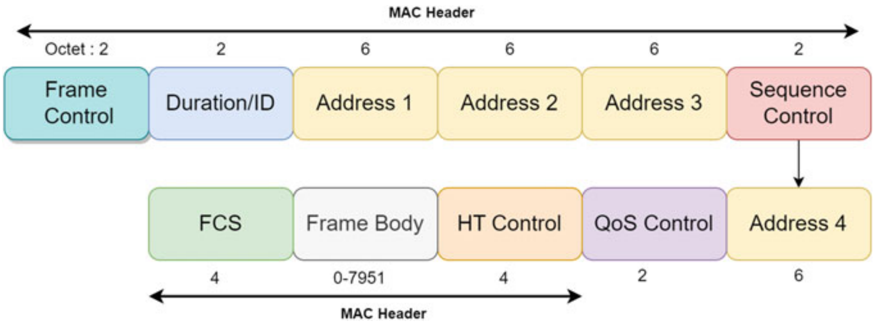


Fig. 5.2 802.11 frame header

configuration for stations connecting to the internet. A typical password is used for the new stations connecting to the WLAN. This provides ease of connection for the other station without compromising the WLAN security. Malicious users can join the network and sniff the traffic for further attacks. The following procedures will be performed to prevent the attackers from listening to the traffic.

After the establishment of Probe communications authentication request will be sent to the AP to verify the current authentication configuration of the stations. Once the AP proves it, i.e., AP supports those security configurations the station's share, the association packets will be exchanged for connection configurations. Finally, the

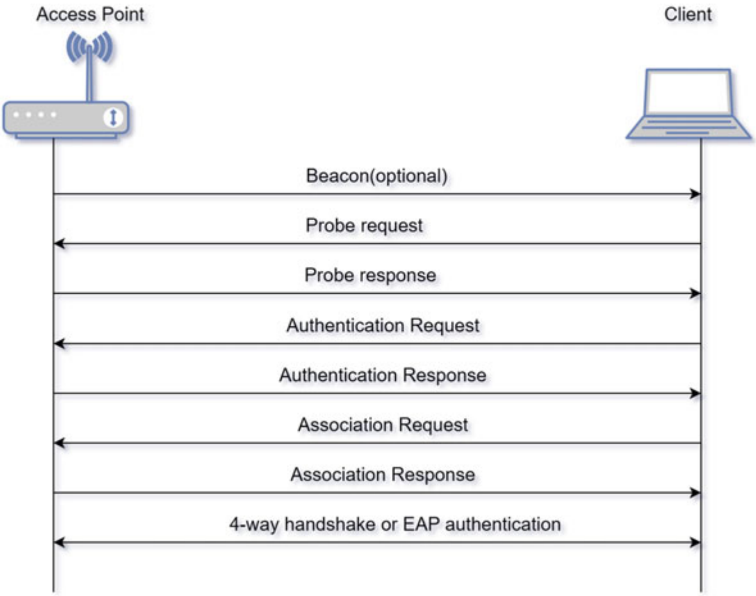


Fig. 5.3 WPA2 working

password will be converted into keys and transferred to other parties securely to prevent on-path attacks on keys. This process is called a 4-way handshake. After this process, all the connections are established with WPA2—Personal standard, and the data will be encrypted with the keys generated in a 4-way handshake at transit (see Fig. 5.3).

WPA2 Enterprise is used for organizations and massive networks. The key difference is using a radius server for generating and managing passwords. Here, all the stations in the same WLAN will use unique passwords for connecting with AP, which the administrator will define. This prevents the attacker from listening to other system traffic from the compromised system. And other security features like certificate verification can be integrated with the enterprise model.

Reconnaissance on Beacons

Beacons are short messages transmitted from the APs to inform the stations of the Wi-Fi services and their configurations within the range of APs. Stations use this data to connect to the corresponding APs manually or automatically.

Attack: Since APs transmit these beacons within their range, any malicious actors can listen to them. These pieces of information are used to discover and analyze the attack targets by understanding the protocols and security configurations. With these details, an attacker can identify the SSID, MAC address, signal range, protocol information, security standards, and the geographical location of that AP. This facilitates the attacker to prepare for further development with the attack scope. Similarly, the station transmits probe requests as a management frame containing the connectivity details the station supports to identify respective APs for scanning. Then the APs with appropriate parameters can send a probe response as a reply to the request from the station, which defines the connectivity. Access to these details by the attackers leads to replay attacks, Beacon frame spoofing [3], Denial of service, and De-authentication attacks, which are discussed in further sections. This reconnaissance defines the weak target the attacker will utilize for the mentioned attacks [4].

Mitigations:

1. Disabling SSID Broadcasting.
2. Hiding SSID on stations.
3. Enable Protection from Wi-Fi Probe requests on stations.

Fragile Configuration of Security Standard

Most attacks are performed due to misconfiguration or lack of device configurations. In Wi-Fi, it is evident that the APs and routers need proper configuration to fulfill the productivity of the devices. But the security configurations are often excluded due to a lack of awareness of their impact on the network. Attackers can use this information to launch advanced attacks, letting them monitor the traffic in stealth and control the traffic by launching on-path attacks.

Mitigations:

1. Changing default passwords to complex passwords.
2. Precise access control.
3. Configuring updated security configurations and standards for encryption.
4. Enabling SSID Protection.
5. Enabling a Firewall, if available.
6. Maintenance of Updates of APs or router software.
7. Update firmware periodically.

De-Authentication Attacks

The authentication frames are used for verifying the security configuration of the connecting stations and for authenticating the station to connect to the AP. De-authentication frames are used to either disconnect the station from the AP or the AP from the station. The de-authentication frames are unencrypted messages transmitted on the AP range. A sniffer in this range can easily capture the frames and decode them. Though these management frames don't carry any user data, these frames can be misused to perform attacks, such as Denial of service. An attacker can disconnect a station from the AP and sniff other packets transmitted while reconnecting; this can be used for further attacks. Though the latest version, 802.11w, provides some security for management frames, not all devices are supported to use it [5]. A similar attack can be performed using association frames.

Mitigations:

1. Management frames from spoofed MAC addresses can be blocked.
2. Using updated security standards like 802.11w.

Password Cracking Attacks

WPA2 is the most used security standard of Wi-Fi. It used a 4-way handshake to establish the connection between the station and the AP after verifying the correctness of the password provided by the station. If the password provided by the station is complex as per the standards, then cracking will eventually become impossible. In-Order to capture the handshake, the attacker will sniff the Wi-Fi traffic. Suppose any new stations join the AP. In that case, it will initiate the transfer of management frames followed by the handshake to establish the connection, which will be captured successfully. But, if the targeted stations are connected to the AP prior, the attacker can disconnect the stations using de-authentication or dis-association attacks. The station will attempt to rejoin with the AP, and the handshake can be captured [6].

After capturing the EAPOL 4-way handshake, an attacker can access the keys to maintain confidentiality in the Wi-Fi network. One of that messages will contain the Pairwise Master key generated by the station (see Fig. 5.4). An existing dictionary file will be used, or a manual customized dictionary file can be created to compare with the key's hash value. If dictionary terms are used as passwords, they will be easily cracked. Configuring a standard password will decrease the possibility of cracking. With the passwords, an attacker can join the Wi-Fi network, which increases the scope of the attack on the WLAN network.

Mitigation:

1. Configuration of complex passwords.
2. Prevention of DOS attacks in Wi-Fi networks.
3. Enabling security for management frames.

Rouge AP Attacks

It is one of the most effective attacks on Wi-Fi networks as other attacks depend on the keys and require precise time of capturing frames to complete the attacks. Evil-Twin or Rouge AP attacks can be performed without dependencies on other devices.

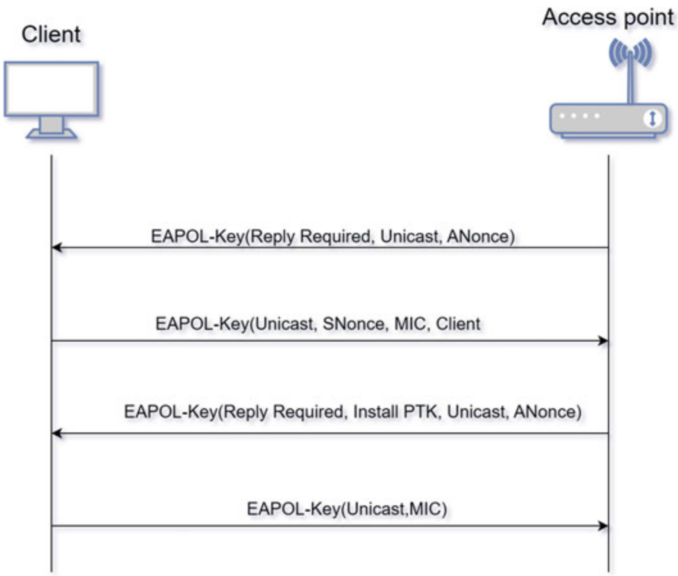

Client
Access point
EAPOL-Key(Reply Required, Unicast, ANonce)
EAPOL-Key(Unicast, SNonce, MIC, Client
EAPOL-Key(Reply Required, Install PTK, Unicast, ANonce)
EAPOL-Key(Unicast,MIC)

Fig. 5.4 Working of WPA2 4-way handshake

Attack: The attacker exploits the beacons transmitted by the AP to impersonate the legitimate AP. Attacker can sniff the SSID and MAC address from the beacons and spoof the information to create a rouge AP. Then the beacons with the same information as the legitimate ones will be transmitted. Stations looking for legitimate stations will identify Rouge AP as a legitimate AP and get connected. Without the involvement of passwords, the attacker can be able to control the Wi-Fi network. If a new client is looking for a connection to a legitimate AP, it will be connected to a rouge AP. Suppose a station is already linked to a legitimate AP. In that case, the attacker will perform a Denial of service to disconnect from the station, and then the stations will connect to rouge AP [7].

Mitigations:

1. Do not connect to unknown open Wi-Fi.
2. Verify the warning notifications while connecting.
3. Avoid Auto Connect features.
4. Use Multi-factor authentication.
5. Do not provide sensitive details in public Wi-Fi.
6. VPN can be used.

Wi-Fi Frame Injection

Though Wi-Fi encrypts user data, injecting malformed packets can compromise the Wi-Fi network. The Wi-Fi stack is configured to accept and process some plain text frames without verifying the sender's authenticity. This can be exploited to perform multiple attacks through frame injection. Broadcast frames can be injected with appropriate frame details during transmission to compromise the AP for further attacks. This attack can be implemented during the regular communication between the station and the AP. This leads to modifying any packet header details and may result in the decryption of the Wi-Fi traffic by compromising the encryption keys [8], leading to frame aggregation and fragmentation attacks [9].

Mitigations:

1. Updating the Wi-Fi Drivers.
2. Using 802.11w.

KRACK Attacks

A flaw in WPA2 leads to the decryption of all the Wi-Fi traffic, which leaks sensitive information to the attacker on any Linux-based system. This attack targets a 4-way handshake when it generates new encryption keys to encrypt the traffic and leads to using existing keys for the encryption, which the attacker can easily decrypt. This can be performed by manipulating the handshake packets (see Fig. 5.5). This can allow the attacker to overtake the TCP connections on the Wi-Fi network without the need for the credentials used by the station or by creating an Evil Twin [10].

Mitigations:

1. Updating the Wi-Fi Drivers.
2. Using WPA3.

Wi-Fi Attacks—WPA3

It is the latest secured version of the Wi-Fi security standard. It contains WPA3 personal and enterprise versions for practical usage. Ciphers suites are upgraded to provide complex encryption schemes which mitigate most of the issues from WPA2. It has an optional feature to connect and communicate with IOT devices faster and more securely. Even open hotspots without encryption are protected using Diffie Hellman key exchange techniques. It provides confidentiality to the user's data though the traffic is in Open Wi-Fi, unlike WPA2, in which the data will be transmitted from station to AP in plain text (see Fig. 5.6).

Personal contains unique keys for encryption though the stations are in the same WLAN. This overcomes the on-path attacks by which an attacker can listen to other's

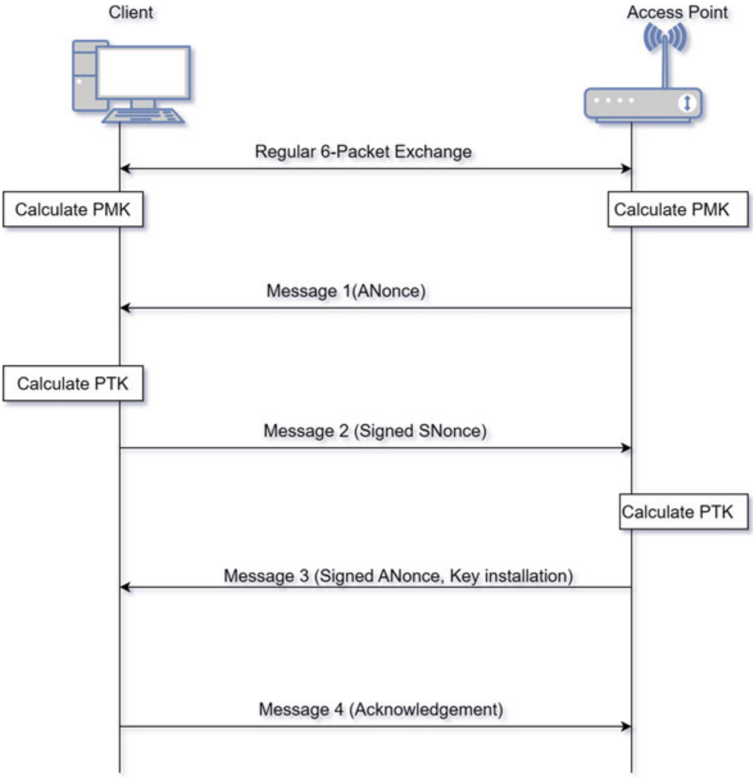

Client
Access Point
Regular 6-Packet Exchange
Calculate PMK
Calculate PMK
Message 1(ANonce)
Calculate PTK
Message 2 (Signed SNonce)
Calculate PTK
Message 3 (Signed ANonce, Key installation)
Message 4 (Acknowledgement)

Fig. 5.5 KRACK attack

traffic by compromising one system in the same WLAN. It uses the SAE protocol to provide better password encryption. This prevents attackers from password-cracking attacks.

Enterprise is designed to provide better security with the feature of 192-bit encryption. It requires a significant upgrade in the radius server's configuration. Down gradation from WPA3 to WPA2 is possible for those devices that do not match the hardware requirements.

Dragon Blood Attack

The Dragon fly handshake in SAE protocols makes the brute password force attacks impossible on WPA3. There was a minor loophole in working this protocol, which led to the compromise of the password that was patched later. A flaw in the design of this protocol will lead to the downgradation of the WPA3 to WPA2 while the device is configured with WPA3. Then, all the devices are vulnerable to all the attacks

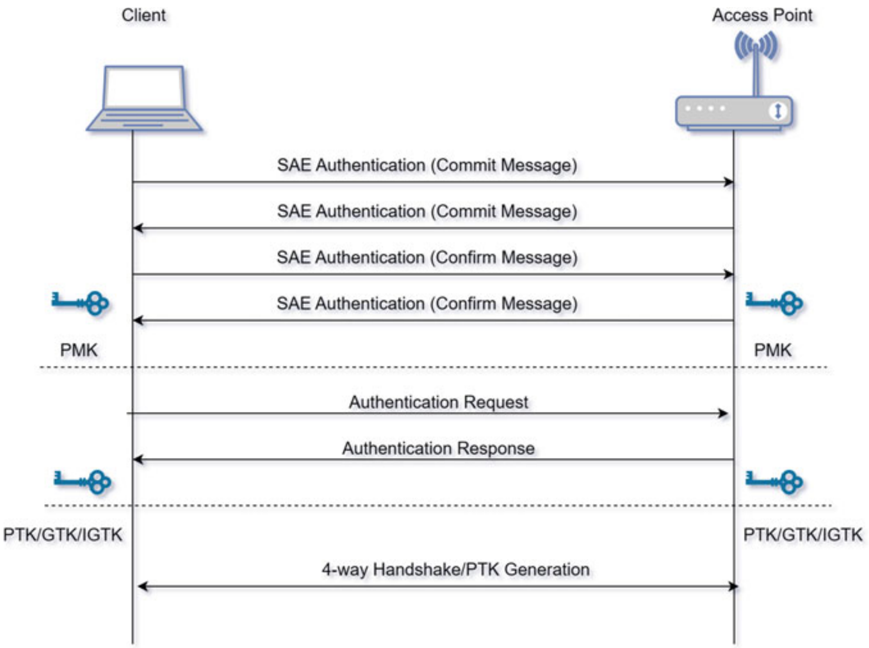

Client
Access Point
SAE Authentication (Commit Message)
SAE Authentication (Commit Message)
SAE Authentication (Confirm Message)
SAE Authentication (Confirm Message)
PMK
PMK
Authentication Request
Authentication Response
PTK/GTK/IGTK
PTK/GTK/IGTK
4-way Handshake/PTK Generation

Fig. 5.6 Working of SAE protocol

committed with WPA2. The identified weakness in the SAE protocol can be abused to recover the password of the Wi-Fi network, launch resource consumption attacks, and force devices into using weaker security groups [11].

On-Path Attack Without Evil Twin

This abuses the flaw in cross-layer interactions between WPA and ICMP protocols by exploiting the ICMP redirect messages. An attacker manually creates an ICMP redirect packet with spoofed details of legitimate AP. This traffic won't be visible to legitimate AP because it will not read the packet. With the help of this meal-crafted packet, the client can be convinced to connect with any other device without even creating one. Then the vulnerability in the network processing unit is discussed, which is responsible for not blocking such spoofed ICMP error packets. This attack works only if the ICMP redirect feature is enabled in the AP. This affects WPA2/ WPA3 Wi-Fi standards. Later it is patched from the NPU level [12].

Wireless Intrusion Detection/Prevention System

What Is WIDS?

Wired networks can be easily defined with the devices' borders, which can allow the administrator to determine the placement of the IDS/ Intrusion Prevention System accordingly. We cannot describe the borders in Wi-Fi since the range of the Wi-Fi routers or APs differs according to the scenario it is used. An attacker can be in any Wi-Fi device range and perform reconnaissance or sniffing-based attacks. To identify and prevent such attacks, Wi-Fi IDS plays a crucial role. It can perform all the features of a regular IDS with extra support for Wi-Fi Functionalities. Though a few station-based Wi-Fi attacks can be challenging to detect and prevent using WIDS, they can control the Wi-Fi network from an AP.

Unlike regular IDS, which works from Layer 3—The network layer, WIDS works from Layer 1 to 7—Physical to Application Layer. With this data transparency, WIDS can restrict or identify the un-legitimate traffic loading the Wi-Fi network. It controls both the AP and the stations in the same network. The signature and anomaly based WIDS features can be incorporated into detecting and preventing Wi-Fi attacks [13, 14].

Detection of Wi-Fi Attacks Based on TCP/IP Layers:

Physical Layer: Reconnaissance and Sniffing of the Wi-Fi traffic can be performed. Though the traffic is encrypted, details about channels, spectrum, modulation schemes, hardware specifications, encoding techniques, and Key generation schemes can be identified, and this information is crucial to an attacker for executing the attacks. Detecting an attacker sniffing the traffic within Wi-Fi is challenging because anyone can listen to the radio signals in its range. But the attacker's presence can be detected by analyzing the anomalies in the channel. Any binary bit injection or alteration of any bits while transmission can be detected with the WIDS. WIPS can maintain access control by avoiding unauthorized connectivity to the Wi-Fi. It can be able to identify and filter the traffic based on the nonlegitimate source of transmission. Verification of security standards and cipher suits can be performed to avoid any degradation of the security. Beacons and Probes can be monitored for attack detection and prevention.

Data-Link Layer: Spoofing can be performed. Since MAC addresses are valid in the LAN, LAN-based spoofing attacks are possible under this layer. ARP spoofing is a standard attack that can be performed inside WLAN. Since the Wi-Fi packets are encrypted from this layer, the insider will initiate this attack, which won't affect the encryption used by the Wi-Fi. The packets will be processed without detection since the APs receive the duplicate encryption keys as the initiation. With the presence of WIDS, these broadcasted spoofed packets can be monitored and analyzed to detect and prevent such spoofing attacks.

After source verification, MAC flooding can be detected by allowing the network's legitimate MAC addresses and permits. This provides better control of WLAN traffic. Similarly, multicast brute force can be detected and prevented by monitoring the irrelevant multicast traffic mapped with its services [15].

Internet Layer: Router poisoning, IP spoofing, Malicious Forwarding of traffic, and On-Path attacks can be performed in this layer. Network-based DOS and DDOS can be detected and prevented with the proper configuration of WIDS with specific rules. An attack can be detected if an enormous amount of traffic is transmitted from the same IP address, and the traffic behaviors will be analyzed with all the IP header parameters for detecting anomalies. WIDS should be configured in a specific path to read the incoming and outgoing traffic in monitor mode to get extra transparency of the WLAN. This adds additional security for the WLAN and reduces the load on the regular firewall and router. WIDS can actively scan the network to detect any anomaly in the Wi-Fi traffic. It can detect evil twin attacks, Rouge APs, and other On-path attacks with the appropriate logic.

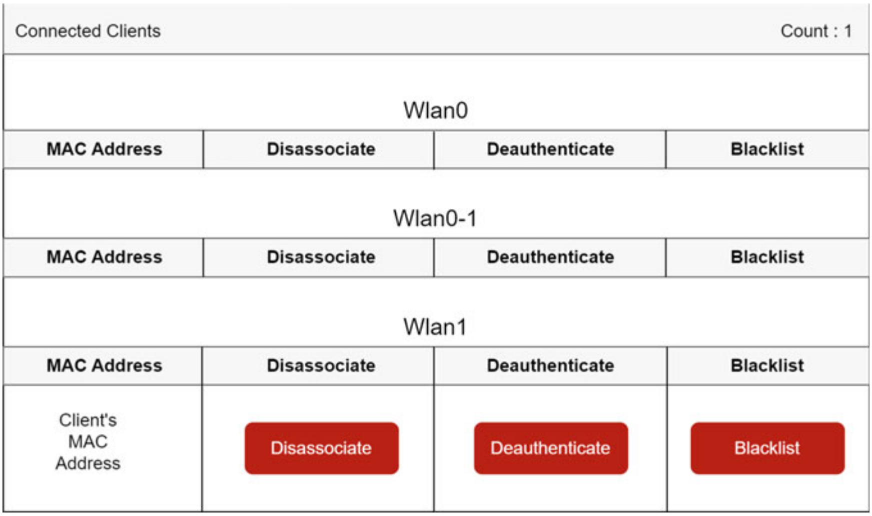
Transport Layer: Port scanning and connection-based flooding attacks can be performed on this layer. WIDS can be able to monitor stateful connections of ports and related associations. For example, an SYN flood can lead to a general DOS attack in a Wi-Fi network. It can be detected using the IP address, but the more effective method is to identify the packet type. Many SYN messages will be transmitted to the destination to perform the attack. Instead of depending on the IP address for detection, the WIDS will check the packet type. In this case, SYN is sent so that the WIDS will expect legitimate traffic like SYN-ACK, then any other packets other than this will be monitored and detected as suspicious and then blocked. Similarly, WIDS can detect station-based and AP-based traffic generations [16].

Application Layer: Detection in this layer works similarly to the regular IDS. Sessions will be monitored for hijacking, and phishing enumerations will be detected. Application-based malicious activities will be detected with the full functionality of a normal IDS. Though with the additional functionality of Wi-Fi, WIDS can equally perform detection of frequent attacks. This provides an additional benefit for WIDS over the regular one.

Wi-Fi Auditing Using Wi-Fi Pineapple

Wi-Fi Pineapple is a Wi-Fi penetration tool that helps to analyze and create various attack scenarios for security testing. It has all the built-in hardware components to perform robust testing. The features are modules that can be downloaded from their cloud console. Most Wi-Fi attacks discussed can be performed using this tool with a proper Graphical user interface.

Clients: This tool can act like a legitimate AP and can list the client's MAC address connected to it based on the associated interface. Accordingly, it facilitates



Connected Clients Count : 1

Wlan0

MAC Address	Disassociate	Deauthenticate	Blacklist

Wlan0-1

MAC Address	Disassociate	Deauthenticate	Blacklist

Wlan1

MAC Address	Disassociate	Deauthenticate	Blacklist
Client's MAC Address	Disassociate	Deauthenticate	Blacklist

Fig. 5.7 Wi-Fi pineapple client information

performing disassociation and de-authentication attacks to test the client's behavior. A group of clients can be allowed or blocked to follow specific rules (see Fig. 5.7).

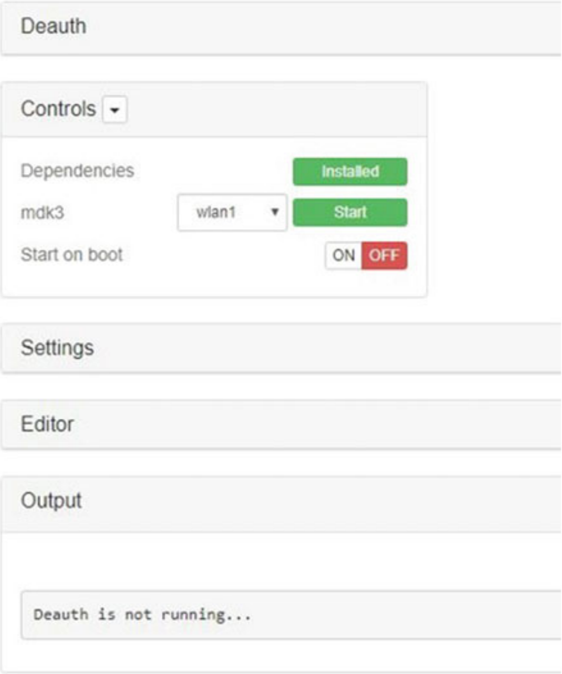
Deauth: It performs deauthentication using mdk3, a proof-of-concept tool for the 802.11 protocol. From the settings, monitor mode will be enabled so that the interface can listen to all the traffic not intended for the system. Then the corresponding interface will be selected to perform the attack. After the successful completion of the attack, the status of the victim can be noticed from the output tab (see Fig. 5.8).

Dwall: This module scans the client's traffic in its WLAN and sniffs the unencrypted web traffic. This provides the IP address, URL visited, cookies, if any, captured, and the images from that web traffic. This provides the consequences of open Wi-Fi and how any malicious actor can listen to other traffic. An attacker can be able to compromise the WPA2/WPA3 encryption with different attacks and can use this feature to sniff these details (see Fig. 5.9).

Evil-Portal: This can be a simple and efficient attack that can compromise clients easily. Once this feature is turned on, an attacker can choose the portal web page that needs twined. Here is an example Google logins page is taken. This page will be cloned and published as a legitimate web page in the browser when the clients connect to this Wi-Fi. The attacker will capture the credentials mentioned by the user on this page. Any advanced phishing attacks can facilitate it. This attack can be implemented after the attacker compromises the encryption of the Wi-Fi network (see Fig. 5.10).

PineAP: This module creates rouge APs by spoofing the existing, legitimate AP. It automatically takes the target AP's name and MAC to perform this attack. Then the

Fig. 5.8 Wi-Fi pineapple de-authentication attack



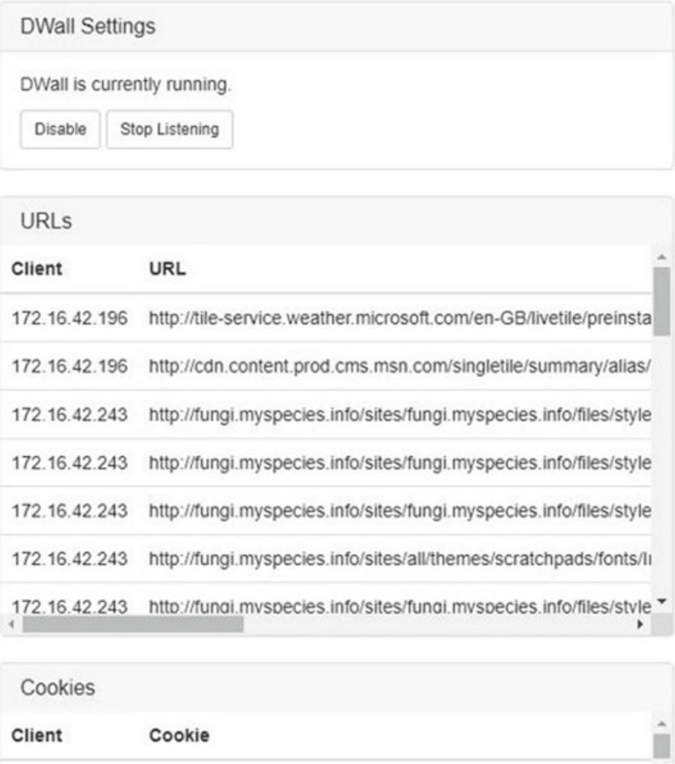
```
Deauth is not running...
```

rouge AP will be projected with higher signal strength with the powerful antennas present in Wi-Fi pineapple. It can also serve deauthentication to the targeted clients and AP to reinitiate the connection by which the rouge AP tricks the clients into getting connected without the need for passwords. Also, this module can create multiple random APs to flood the channel or confuse the victim in connecting with the correct AP. Then once the victim is connected, the Evil portal will be used to perform further attacks.

Network Scans: It performs network scans to perform reconnaissance in the WLAN. This will provide the attacker with several active devices, gateway, and open and closed ports with its services. Nmap is an open-source, powerful network scanning tool integrated with this device (see Fig. 5.11).

Wi-Fi Scan: This module provides complete details about clients' Wi-Fi parameters. SSID (even if hidden), MAC address, Wi-Fi encryption standard, cipher suite details, authentication method, Wi-Fi channel, and Frequency with signal strength will be listed here. This provides an attacker with a vast advantage in targeting a Wi-Fi network. It is used to perform wardriving. The main advantage of this tool is that it also provides a feature to perform a death attack to capture the handshake for further password cracking (see Fig. 5.12).

Other modules can help verify the Wi-Fi network's security configurations by performing advanced on-path attacks and password cracking to compromise the



DWall Settings

DWall is currently running.

URLs

Client	URL
172.16.42.196	http://tile-service.weather.microsoft.com/en-GB/livetile/preinsta
172.16.42.196	http://cdn.content.prod.cms.msn.com/singletile/summary/alias/
172.16.42.243	http://fungi.myspecies.info/sites/fungi.myspecies.info/files/style
172.16.42.243	http://fungi.myspecies.info/sites/fungi.myspecies.info/files/style
172.16.42.243	http://fungi.myspecies.info/sites/fungi.myspecies.info/files/style
172.16.42.243	http://fungi.myspecies.info/sites/all/themes/scratchpads/fonts/li
172.16.42.243	http://fungi.myspecies.info/sites/fungi.myspecies.info/files/style

Cookies

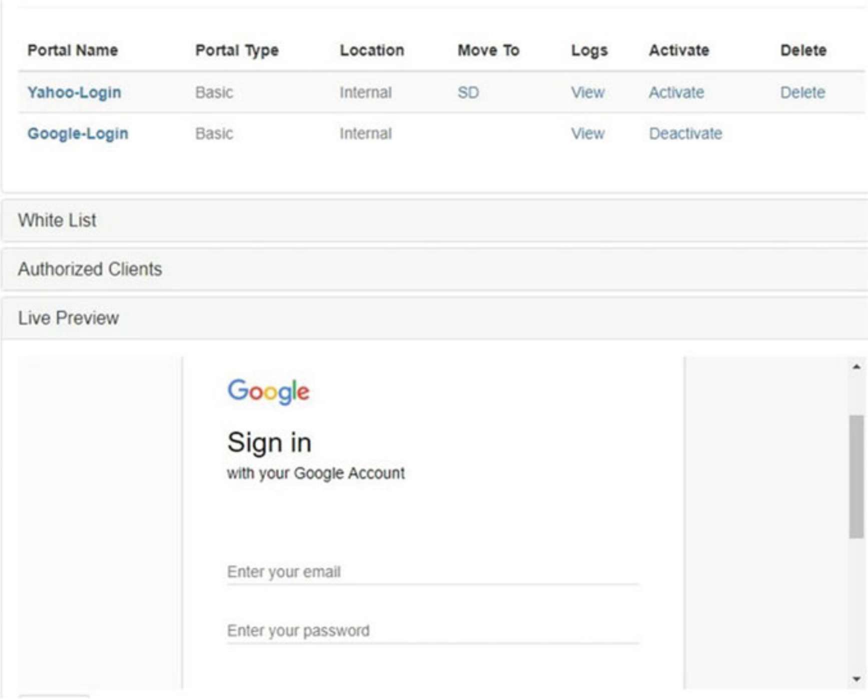
Client	Cookie

Fig. 5.9 Wi-Fi pineapple DWALL

network. It can be incorporated to advance detection and prevention techniques for WIDS further.

Conclusion and Future Work

This work discusses the importance of Wi-Fi usage in current technological trends and its security problems by analyzing various attacks on Wi-Fi networks under multiple security standards. The impact of these attacks with the mitigations is studied. Wi-Fi intrusion detection system is introduced and explained with its importance in securing Wi-Fi networks. TCP/IP layer-based attacks and methods of detection are investigated. With the capability of WIDS in Wi-Fi networks, it can be extended with advanced anomaly-based detection strategies to prevent attacks [17]. This can be further expanded as a Wi-Fi Intrusion detection network that contains multiple WIDS with SDN functionalities and zero trust security to maintain adversary-aware IDS technologies [18, 19].



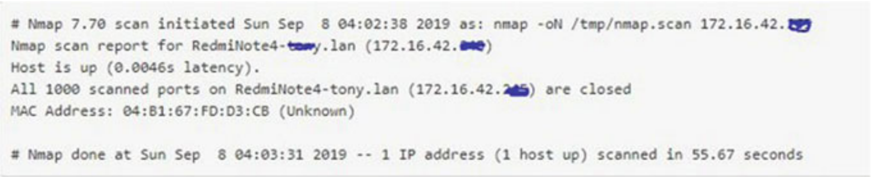
Portal Name	Portal Type	Location	Move To	Logs	Activate	Delete
Yahoo-Login	Basic	Internal	SD	View	Activate	Delete
Google-Login	Basic	Internal		View	Deactivate	

White List

Authorized Clients

Live Preview

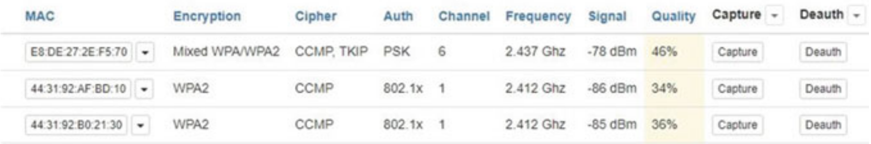
Fig. 5.10 Wi-Fi pineapple evil portal



```
# Nmap 7.70 scan initiated Sun Sep  8 04:02:38 2019 as: nmap -oN /tmp/nmap.scan 172.16.42.[redacted]
Nmap scan report for RedmiNote4-[redacted]y.lan (172.16.42.[redacted])
Host is up (0.0046s latency).
All 1000 scanned ports on RedmiNote4-tony.lan (172.16.42.2[redacted]) are closed
MAC Address: 04:B1:67:FD:D3:CB (Unknown)

# Nmap done at Sun Sep  8 04:03:31 2019 -- 1 IP address (1 host up) scanned in 55.67 seconds
```

Fig. 5.11 Wi-Fi pineapple NMAP



MAC	Encryption	Cipher	Auth	Channel	Frequency	Signal	Quality	Capture	Deauth
E8:DE:27:2E:F5:70	Mixed WPA/WPA2	CCMP, TKIP	PSK	6	2.437 Ghz	-78 dBm	46%	Capture	Deauth
44:31:92:AF:BD:10	WPA2	CCMP	802.1x	1	2.412 Ghz	-86 dBm	34%	Capture	Deauth
44:31:92:B0:21:30	WPA2	CCMP	802.1x	1	2.412 Ghz	-85 dBm	36%	Capture	Deauth

Fig. 5.12 Wi-Fi pineapple wardriving

References

1. https://datatracker.ietf.org/doc/html/rfc7494, last accessed 18 June 2023
2. https://datatracker.ietf.org/doc/rfc5416/, last accessed 18 June 2023
3. Martínez A et al (2008) Beacon frame spoofing attack detection in IEEE 802.11 networks. In: 2008 third international conference on availability, reliability, and security. IEEE
4. Thomas AM et al (2021) Evaluation of wireless AP security and best practices for mitigation. In: 2021 5th international conference on electrical, electronics, communication, computer technologies and optimization techniques (ICEECCOT). IEEE
5. Al-Nuaimi MAS, Ibrahim AA (2023) Analyzing and detecting the de-authentication attack by creating an automated scanner using Scapy
6. Sudar C, Arjun SK, Deepthi LR (2017) Time-based one-time password for Wi-Fi authentication and security. In: 2017 international conference on advances in computing, communications and informatics (ICACCI). IEEE
7. Shrivastava P, Jamal MS, Kataoka K (2020) EvilScout: detection and mitigation of evil twin attack in SDN enabled WiFi. IEEE Trans Netw Service Manag 17(1):89–102
8. Vanhoef M et al (2023) Testing and improving the correctness of Wi-Fi frame injection. In: Proceedings of the 16th ACM conference on security and privacy in wireless and mobile networks. ACM
9. Vanhoef M (2021) Fragment and forge: breaking Wi-Fi through frame aggregation and fragmentation. In: 30th USENIX security symposium (USENIX security 21), USENIX Association, pp 161–178. https://www.usenix.org/conference/usenixsecurity21/presentation/vanhoef
10. Vanhoef M, Piessens F (2017) Key reinstallation attacks: forcing nonce reuse in WPA2. In: Proceedings of the 24th ACM conference on computer and communications security (CCS). ACM
11. Vanhoef M, Ronen E (2020) Dragonblood: analyzing the dragonfly handshake of WPA3 and EAP-pwd. In: IEEE symposium on security and privacy (SP). IEEE.
12. Feng X et al (2022) Man-in-the-middle attacks without rogue AP: when WPAs meet ICMP redirects. In: 2023 IEEE symposium on security and privacy (SP). IEEE Computer Society
13. Fung CJ, Boutaba R (2013) Design and management of collaborative intrusion detection networks. In: 2013 IFIP/IEEE international symposium on integrated network management (IM 2013). IEEE
14. Pleskonjic D (2003) Wireless intrusion detection systems (WIDS). In: 19th annual computer security applications conference
15. Baharudin N et al (2015) Wireless intruder detection system (WIDS) in detecting de-authentication and disassociation attacks in IEEE 802.11. In: 2015 5th international conference on IT convergence and security (ICITCS). IEEE
16. Satam P, Hariri S (2021) WIDS: an anomaly based intrusion detection system for Wi-Fi (IEEE 802.11) protocol. IEEE Trans Netw Serv Manag 18(1):1077–1091. https://doi.org/10.1109/TNSM.2020.3036138
17. Satam P, Hariri S (2020) WIDS: an anomaly based intrusion detection system for Wi-Fi (IEEE 802.11) protocol. IEEE Trans Netw Serv Manag 18(1):1077–1091
18. Abilesh Raja G et al (2022) Detection of Log4j attack in SDN environment. In: International conference on information and communication technology for competitive strategies. Springer Nature Singapore, Singapore
19. Abhiram D, Harish R, Praveen K (2022) Zero-trust security implementation using SDP over VPN. In: Inventive communication and computational technologies: proceedings of ICICCT 2021. Springer, Singapore

Chapter 6
Privacy Preserving Spatio-Temporal Attribute-Based Access Control for Collaborative E-Healthcare

Kasturi Routray and Padmalochan Bera

Abstract Collaborative e-healthcare leverages the benefits of the cloud to facilitate seamless collaboration and information sharing among healthcare professionals. However, ensuring the security and privacy of data in cloud environments is one of the major challenges. This chapter presents a privacy-preserving spatio-temporal attribute-based access control for collaborative e-healthcare. Our proposed cryptosystem leverages ciphertext policy attribute-based encryption to protect patient data privacy. Trapdoors corresponding to dynamic time and location attributes are added to the access policy structure, enabling verification based on dynamic parameters. These trapdoors eliminate the requirement of frequent key revocations and reassignments when the context changes. Geo-hashes of the locations are used as attributes to support location constraints. Additionally, hiding the access policy structure prevents leakage of sensitive attribute information associated with the access policy. Our proposed scheme also provides security against key escrow attacks. For practical deployment of our scheme, we introduce fog servers near the data access area that generate time and location tokens. These servers also aid in the decryption process for resource-constrained users. Security and performance analysis shows the efficacy of our proposed scheme for practical applications in e-healthcare systems, enabling precise control over sensitive electronic medical health records.

Keywords Time and location constraint · Hidden access policy · Computation outsourcing · CP-ABE · Key escrow attacks

K. Routray (✉) · P. Bera
Indian Institute of Technology Bhubaneswar, Bhubaneswar, India
e-mail: s21ee09001@iitbbs.ac.in

P. Bera
e-mail: plb@iitbbs.ac.in

D. Gountia et al. (eds.), *Information Technology Security*, Springer Tracts in Electrical and Electronics Engineering, https://doi.org/10.1007/978-981-97-0407-1_6

Introduction

Advancements in technology have paved the way for collaborative e-healthcare, allowing healthcare providers to collaborate remotely, share patient information, and make intelligent and timely decisions collectively [1]. This collaborative approach brings numerous benefits, such as improved patient outcomes, reduced medical errors, and enhanced efficiency in healthcare delivery. The ability to store and access patient data in the cloud is a fundamental pillar of this collaborative model, enabling seamless information exchange and real-time decision-making.

The general architecture of a collaborative e-healthcare system hosted in the cloud is illustrated in Fig. 6.1, which involves utilizing cloud computing infrastructure as the central storage and processing platform. Patient data, including medical records, diagnostic reports, and treatment plans, are securely stored in the cloud. Healthcare professionals, including doctors, nurses, and specialists, can access and analyze this data remotely, regardless of location. This architecture promotes real-time collaboration and communication among healthcare professionals by breaking geographical barriers and enabling timely interventions. Additionally, cloud-based solutions provide scalability and flexibility, accommodating the growing volume of healthcare data and supporting the integration of emerging technologies such as artificial intelligence and machine learning.

While collaborative e-healthcare has several advantages, storing sensitive patient data on untrusted cloud environments raises concerns about data privacy. Patients rely on healthcare providers to handle their confidential health information with utmost care and privacy. However, due to the multi-tenancy model of cloud computing, there are potential risks of unauthorized access, data breaches, and privacy violations, necessitating robust measures to protect patient confidentiality and comply with stringent regulatory requirements. Healthcare institutions must make data privacy preservation a top priority to guarantee the security and integrity of patient data

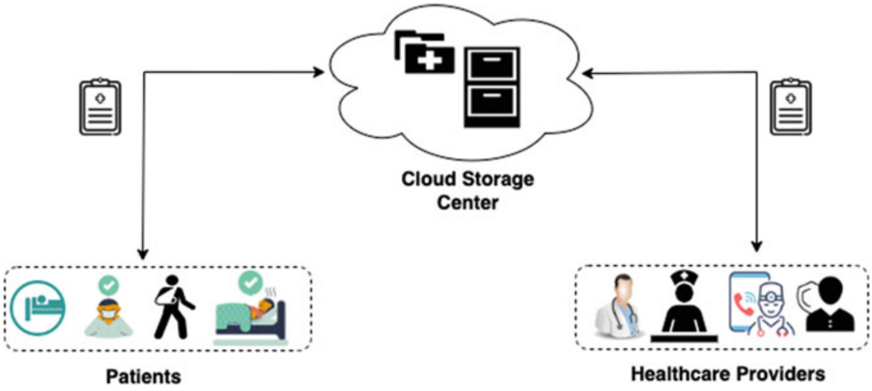


Fig. 6.1 General architecture of E-healthcare system

at all stages of its existence. Failure to safeguard data privacy can lead to breaches of trust, legal ramifications, reputation damage, and compromised patient care. To mitigate these risks, cryptographic solutions play a crucial role. Existing cryptographic techniques, such as symmetric and asymmetric encryption, have been employed to protect data confidentiality and integrity. Conventional encryption methods such as Advanced Encryption Standard (AES) [2] and Data Encryption Standard (DES) [3] are employed to guarantee secure access control for data stored in the cloud. Nevertheless, they lack the needed granularity and flexibility for fine-grained access control based on user attributes. Also, these methods make systems more complex and add significant overhead for managing keys, especially for dynamic group-based access control.

Ciphertext-policy-based encryption (CP-ABE) [4] schemes offer a promising secure data access control solution in collaborative applications. CP-ABE empowers data owners to establish access policies (in terms of attributes) to be embedded with the ciphertext to maintain greater control over their data while allowing it to be processed and analyzed by cloud providers and other third-party applications. This attribute-based approach enables personalized and flexible access controls, ensuring that only authorized individuals with specific attributes can access sensitive healthcare data. Besides, it improves the efficiency and scalability of cloud platforms by reducing the need for complex and centralized access control mechanisms.

Nonetheless, a primary challenge associated with CP-ABE schemes is that access to the encrypted data is defined based on static attributes of the users so it fails to effectively accommodate dynamic attributes. As dynamic attributes change value frequently, a new decryption key must be given for the updated attribute value, increasing the system overhead. This prohibits defining the access policy using attributes related to time and location. Hence, to achieve precise access control, it becomes essential to specify that particular data can solely be decrypted during designated time intervals and from specified locations, in addition to considering normal attributes.

A Motivating Use-Case: We present a practical scenario to describe the necessity of effective time and location-aware access control mechanisms in collaborative e-healthcare systems. Figure 6.2 shows a pictorial representation of the use case. Suppose Peter suffers from heart disease, hypertension, and arthritis. Doctors owing to his deteriorating heart condition, advise Peter to undergo heart transplant surgery in Hospital B. Peter is undergoing treatment under Doctor S_B, who works in Hospital B and can access Peter's electronic medical records to examine his past medical records. Moreover, Peter received an early referral to Hospital A for arthritis therapy. Doctor S_A in Hospital A, who handled Peter's treatment, suggested him to join a hospital-sponsored program for weight management and physiotherapy. Peter receives a smart health sensor-based device that tracks and logs his daily physical activities and locations. Due to Peter's deteriorating health, he wants to provide complete access to his electronic health records (EHRs) to family members and occasionally to emergency services. Furthermore, Peter's insurance provider guarantees to lower his rate if his heart condition significantly improves. In the above-presented example scenario, the data owner is patient Peter. Staff members of Hospital A and B,

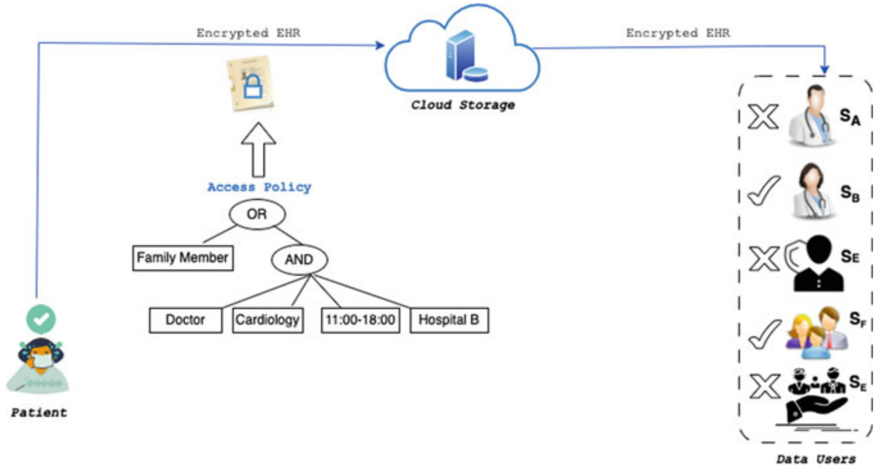


Fig. 6.2 A motivating use case

the insurance provider, the emergency service personnel, and Peter's family members are the data users. Users are intended to have restricted access to the stored EHR because of privacy issues. Without a suitable access control system, some users might unintentionally or purposely breach their limitations and get access to more resources than they should.

In this scenario, Peter may want to encrypt his EHR so that only users who are authorized to have access can examine the EHR. For instance, the EHR can be encrypted for online storage to the cloud with access to the doctors of Hospital B's cardiology division during office hours. It can be achieved with an access policy read as: $\{(location = Hospital\ B)\ AND\ (department = Cardiology)\ AND\ (profession = Doctor)\ AND\ (11:00 < time < 18:00)\}\ OR\ (relation = Family\ Member)$. In that case, users attempting to access the ciphertext must satisfy the access policy defined by Peter. Doctor S_B, who works in the cardiac department at Hospital B, will satisfy the access policy and can decrypt the encrypted EHR during *11:00 AM–6:00 PM*. The family members of the Peters also can access his EHR. Doctor S_A, who works for Hospital A, emergency personnel, and the insurance provider, cannot decrypt encrypted EHR containing surgery-related information as their attributes cannot satisfy the access policy. In this use case scenario, the spatio-temporal-based access control is crucial in enabling emergency response teams to access Peter's EHR quickly and securely, ensuring accurate and timely medical interventions. Thus, by enforcing context-aware access policies, healthcare organizations can maintain data privacy and confidentiality while facilitating collaborative decision-making in context-sensitive e-healthcare emergencies.

Besides supporting time and location-based access control attributes, concealing the attributes of access policy [5], preventing collusion attacks by key generation authorities [6], and resource limitation [7] of data user devices are other significant challenges in establishing secure storage within third-party cloud data centers. In CP-ABE cryptosystems, the access policy attached to the ciphertext can leak sensitive encryptor-specified access policy attributes to unauthorized entities. Another important issue with CP-ABE is the key generation authority issues all user's decryption keys and can decrypt ciphertext by calculating the corresponding secret key. Therefore, it becomes essential to counteract the possible collusion attack between the data storage center and key generation authority. Furthermore, within the context of conventional cloud infrastructure, the CP-ABE cryptosystem is impractical for end users due to limitations in resources, high communication costs, and extended transmission times.

Our Contribution

This chapter presents a novel and practical approach for location and time-aware CP-ABE that uses fog-computing architecture toward effective implementations of different computation-intensive operations. We introduce an efficient privacy-preserving spatio-temporal ciphertext policy attribute-based encryption solution designed for cloud storage. Data owners have the ability to incorporate time and location-based limitations into the access policy using both normal (static attributes) and temporal and spatial (dynamic) attributes. User decryption keys are associated with normal attributes, whereas access policy is embedded with trapdoors defining the constraints on dynamic time and location attributes. Our scheme utilizes geohash [8] code to define location constraints specifying the valid area attached to the access control policy. It is also used to model user location attributes. To achieve access policy obfuscation, vectorized access policy [9] is embedded in ciphertext which ensures the specific attributes and conditions required for accessing the data are hidden from unauthorized parties. The proposed scheme utilizes the key secrecy protocol [10] between the storage center and data owners to enhance security against key escrow attacks. Furthermore, we have incorporated fog servers into the design, strategically positioned across geographic locations to enhance location awareness and reduce communication overhead for end-user devices. To streamline computational efficiency, a substantial portion of computation is delegated to proxy servers based on fog computing. The primary contributions of our proposed cryptographic system can be outlined as follows:

- We introduce a CP-ABE scheme designed to effectively accommodate both static and contextual attributes for dynamic access control. We provide an application scenario from the e-health domain to demonstrate the usability of our framework for spatio-temporal-based access control for cloud storage.
- Our scheme provides improved security as the access policy in the ciphertext is converted into vector form before being sent to the storage center. Additionally, the ciphertext is obscured with a shared secret key between the data owner and the cloud which prevents key escrow attacks.

- Our scheme partially outsources computationally expensive calculations involving bilinear maps and group operations to fog servers, effectively diminishing the computational load on the user's end. Additionally, it establishes a safeguard to ensure that both the fog server and the cloud remain unable to deduce the original message from the partially decrypted ciphertext.
- A thorough analysis of the security properties and comparative study of performance in terms of computation complexity and experimentation has been performed to show the effectiveness and efficiency of the proposed scheme compared to the existing CP-ABE cryptosystem.

Chapter Organization

The subsequent sections of this chapter are organized as follows: In Sect. 6.2, we delve into existing schemes within this domain. Section 6.3 outlines the mathematical foundations. In Sect. 6.4, the system model and workflow of the scheme are presented. Section 6.5 gives the mathematical construction of our proposed CP-ABE cryptosystem. The security properties of the scheme are assessed in Sect. 6.6. Section 6.7 presents the complexity and experimentation analysis. Finally, we conclude our work in Sect. 6.8.

Related Work

There have been researches to develop fine-grained access control models in CP-ABE-based schemes, [4, 11–21]. These works do not possess a structured mechanism for taking into account the user's context when rendering access decisions. In the basic CP-ABE scheme [4], there is support for numerical attributes. Temporal and spatial attributes can be added to the policy and decryption key using integer attributes. This scheme uses a bit-matching technique for comparing attributes, resulting in large sizes of user decryption keys and costlier encryption and decryption operations. Further, the ubiquitous partial order relations do not align with strict equality operations. Weber [11] suggests adding a separate subtree for dynamic attributes to ABE access policy using AND gate. However, this lowers the expressiveness of access policy specification since the dynamic attribute subtree must be evaluated to true, or the decryption will fail. Weber [14] scheme suggests XORed location information with a normal key for encryption and decryption operations of the scheme. This scheme requires all the users requesting a particular resource to be present at a specified location. Comparison-Based Encryption (CBE) [12] first proposed a flexible scheme to handle range attribute comparison utilizing the notion of forward and backward functions. It employs integer comparison to generate designated cryptographic values for a given integer range and uses a hierarchical attribute access tree. As a result, when the dynamic attributes of access policy increase, it significantly increases the computation and transmission cost of the scheme. Balani and Ruj [16] propose the incorporation of a revocation capability into CBE by introducing additional blinding to attribute key components, specifically tailored for a dynamic cloud environment. Denisow et al. [19] cryptosystem allows dynamic attribute key components to be attached to an existing decryption key. In this approach, only the added attribute's key components need to be computed, and the existing key can be used without any changes. However, in time-critical applications, where in fractions of seconds, the verification of dynamic attributes must be done to decrypt the data, this scheme is unsuitable. CCP-CABE [18], which stands for Constant-size Ciphertext-Policy Comparative Attribute-Based Encryption, employs an innovative range derivation function to compare attributes within a range and leverages a proxy server to partially encrypt and decrypt data. This method handles negative attributes and wild cards but doesn't support location-based and static user attributes. Many proposed works have been put forth to manage dynamic location information as a component of access policy attributes. A location-based access control technique is presented in [21], which additionally takes into account access time information. Its primary emphasis revolves around safeguarding user's location privacy by employing coarse location data, represented in terms of ranges, as an attribute. In Ref. [20], an approach to a location-aware cryptosystem is proposed, wherein location constraints are formulated as trapdoors within an access tree structure. However, this method does not take into account time constraints. In [22], a spatio-temporal based CP-ABE is discussed, which utilizes a modified access structure and employs a multi-dimensional range derivation function. Nevertheless, it does not tackle the issue of user revocation. Privacy Preserving Location-Based Access Control (PPLBAC) [23] proposes a location as contextual information and location range as a policy and users can get the location and time key components from the attribute authorities before making access request. In TAFC [24], authors present a time-sensitive data publishing scheme using time released encryption scheme. Using the trapdoor mechanism, this scheme allows data owners to flexibly provide time privileges inside the data access policy. However, issuing time tokens at each predefined time point introduces heavy communication and computation overheads.

The schemes [25–29] have proposed solutions for fine-grained access control in a collaborative e-health care system. In [25], the scheme proposed to encrypt patient health data based on healthcare providers' attributes or credentials, and to decrypt EHRs, the data user's attribute must match the access policy. In [26], the scheme addresses the issue of user attribute revocation in a decentralized setup. In [27], the author proposes the utilization of attribute-based encryption for the purpose of selective access authorization and employs cryptographic secret sharing to distribute electronic health records among multiple cloud centers. Meanwhile, scheme [28] uses a multi-authority CP-ABE setup to achieve data security and user privacy in an e-healthcare environment. Additionally, it divides the data users into personal and private networks for increased efficiency. In scheme [29], addresses the problem of frequent user attribute update problem in outsourced CP-ABE schemes, but this scheme lacks access control based on contextual attributes.

Apart from the above-mentioned problems in the schemes, collusion attacks by key generation authorities, access policy privacy, and resource constraints of user devices represent significant hurdles in achieving a secure and workable cloud-based storage solution. The CP-ABE cryptosystem should prioritize preserving the privacy of access policy attributes and should prevent decryption key attacks by key generation authority. The existing CP-ABE schemes are also unsuitable for end users in traditional cloud infrastructure due to resource limitations of user devices, expensive communication, and transmission costs. Thus, the essential prerequisites for cryptosystems employed in cloud-based applications encompass spatio-temporal attributes, concealed access policies, thwarting key generation authority attacks, and the delegation of computations. To our current knowledge, none of the existing research considers the integration of time, location, concealed access policies, and the outsourcing of computations into a unified CP-ABE cryptosystem.

Mathematical Background

In this section, we briefly discuss the mathematical preliminaries.

Bilinear Map

The basis of our cryptographic system relies on prime order cyclic multiplicative groups, denoted as $\mathbb{G}_0$ and $\mathbb{G}_1$. These groups are associated with a bilinear map [30] $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$, which exhibits the following characteristics:

- Bilinearity: The equation $e(g^m, g^n) = e(g, g)^{mn}$ holds true for any integers $m, n \in \mathbb{Z}_p$ and an element $g \in \mathbb{G}_0$.
- Computability: For any integers $m, n \in \mathbb{Z}_p$ and element $g \in \mathbb{G}_0$, there exists a polynomial-time algorithm to calculate $e(g, g)^{mn}$.
- Non-degeneracy: The inequality $e(g, g) \neq 1$ holds for all generator elements $g \in \mathbb{G}_0$, where 1 denotes the identity element of $\mathbb{G}_1$.

Ciphertext-Policy Attribute-Based Encryption

CP-ABE [4] is the most widely used variant among attribute-based encryption techniques [31]. It empowers the data owner to establish access constraints that determine data access permissions. Figure 6.3 illustrates a general workflow of CP-ABE. During the encryption process, the access policy is embedded in ciphertext, enabling the data owner to selectively authorize individuals who can retrieve the plaintext. This

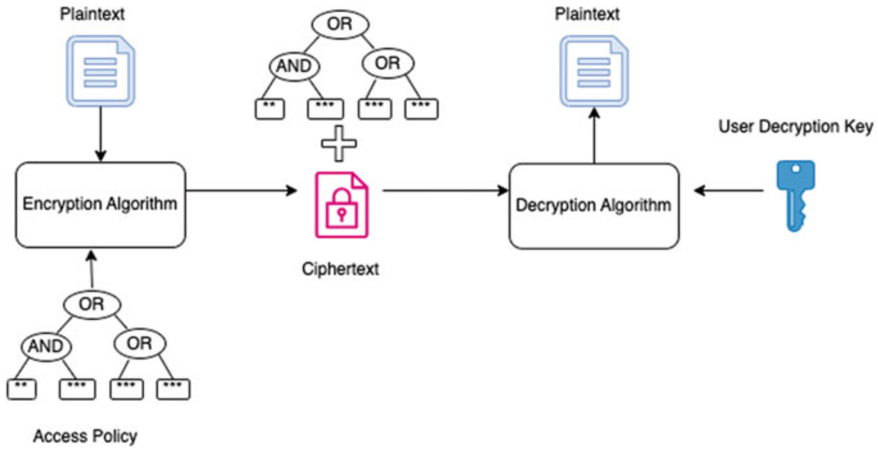


Fig. 6.3 CP-ABE workflow

provides the data owner with greater control over their outsourced data. Valid users who possess the required attribute set to satisfy the access policy can successfully decrypt the encrypted data. Consequently, CP-ABE offers a significant advantage by allowing sensitive data to be securely stored on an untrusted server without the need for authentication checks during data access.

The CP-ABE scheme encompasses four key components: the key generation center, data owner, data user, and cloud server. To initiate the system, a trusted authority establishes public parameters and a master secret key. The data owner proceeds to encrypt their data using symmetric encryption methods like AES, DES, etc. Following this, an access policy is formulated by the data owner, outlining authorized entities, and the associated symmetric key is encrypted under the access policy using CP-ABE. User receive their decryption keys from the key generation authority, based on their attribute sets, thereby enabling data decryption. In a CP-ABE scheme, the workflow involves four key algorithms to ensure secure and flexible data access control:

- **Setup()**: In the setup phase, a trusted authority is responsible for generating the master secret and public parameters. The master secret key is used for generating the decryption keys of users, while the public parameters are disseminated to both users and data owners.
- **Generate_Key()**: During key generation, a user's decryption key is generated based on their attributes. The user specifies their attributes and the key generation center creates a decryption key linked to those specified attributes. This decryption key allows the user to decrypt the data that satisfies the access policy defined by those attributes.
- **Encrypt()**: In the process of encrypting data, the owner defines an access policy that dictates the required attributes for data decryption. Using the public parame-

ters, the data is subsequently encrypted, and the access policy is embedded within the ciphertext. This ensures that only users possessing the specified attributes are able to successfully decrypt the data.

- **Decrypt()**: When users request access to encrypted data, they furnish their decryption key along with the ciphertext to the decryption algorithm. Upon receipt of the ciphertext, the decryption algorithm initiates a validation procedure to verify whether the user's attributes align with the access policy encoded within the ciphertext. If the user's attributes conform to the policy, access is granted, enabling the user to decrypt the data and gain entry to it.

By employing the CP-ABE scheme, fine-grained access control is achieved, allowing data encryption with access policies that depend on multiple relevant attributes. This enables personalized and attribute-based access control, promoting data security, confidentiality, and privacy in an untrusted cloud environment.

Access Policy Structure

The access policy structure in CP-ABE can be represented using a logical expression or a tree-like structure [32]. It defines the rules and conditions that determine which users are granted access to encrypted data. It specifies the attributes required by a user to decrypt a specific ciphertext. Access policy defined using a tree structure consists of leaves representing the attributes and logical operators such as AND, OR, and NOT as internal zor non-leaf nodes. Every non-leaf node n is associated with two values: n_c and n_t, where n_c defines the number of child nodes, and n_t presents threshold value denoting the number of child nodes of n satisfying specified condition. For instance, if $(n_t = 1)$, the leaf node represents an *OR* gate, whereas, if $(n_t = n_c)$ leaf node is an AND gate.

Satisfying Access Tree

The function $Sat(\mathbb{A}, \mathbb{T}_x) = 1$ indicates that an attribute set $\mathbb{A}$ fulfills the access control policy defined by the subtree $\mathbb{T}_x$ rooted at x. The evaluation of $Sat(\mathbb{A}, \mathbb{T}_x)$ can be done iteratively according to the following rules:

- If x is a leaf node: Check if the attribute $att(x)$ is present in $\mathbb{A}$. If it is, then set $Sat(\mathbb{A}, \mathbb{T}_x) = 1$. Otherwise, set $Sat(\mathbb{A}, \mathbb{T}_x) = 0$.
- If x is an internal node with m child nodes $x_1, \dots, x_{\mathrm{num}_m}$: Check if there exists a set I of indices $\{1, \dots, \mathrm{num}_m\}$ such that the cardinality of I, denoted as $|I|$, satisfies $|I| \geq k_x$ and $\forall j \in I$, we have $Sat(\mathbb{A}, \mathbb{T}_{x_j}) = 1$. If such a subset I exists, set $Sat(\mathbb{A}, \mathbb{T}_x) = 1$; otherwise, set $Sat(\mathbb{A}, \mathbb{T}_x) = 0$.

Secret Share Distribution

To distribute the secret s based on an access tree $\mathbb{T}$ and a set of leaf nodes $\mathbb{Y}$ is defined as follows:

$$\{q_y(0) \mid y \in \mathbb{Y}\} \leftarrow \text{DistributeShare}(\mathbb{T}, s).$$

This algorithm operates in a top-down manner, generating a polynomial q_x of degree $k_x - 1$ for each node $x \in \mathbb{T}$:

- x is root node: set $q_x(0) = s$ and randomly choose $k_x - 1$ coefficients for the polynomial q_x.
- x is a leaf node: set $\{q_x(0) \mid x \in \mathbb{Y}\} \leftarrow \text{DistributeShare}(\mathbb{T}, s)$.
- x is an internal node: set $q_x(0) = q_{\text{parent}(x)}(\text{index}(x))$ and randomly choose $k_x - 1$ coefficients for the polynomial q_x.

Upon completion of the algorithm, each leaf node x is assigned a value $q_x(0)$, representing the secret share of s at node x.

Reconstructing the Root Secret

Given an access tree $\mathbb{T}$ and a set of values $\{F_{y_1}, \ldots, F_{y_m}\}$, corresponding to the leaf nodes $y_1, \ldots, y_m$ of $\mathbb{T}$. Additionally, it is specified that $Sat(\{\text{att}(y_1), \ldots, \text{att}(y_m)\}, \text{T}) = 1$ and $F_{y_\eta} = e(g, g)^{q_{y_j}(0)}$. Set of values $\{q_{y_1}(0), \ldots, q_{y_m}(0)$ are secret shares of $s\}$ according to $\mathbb{T}$. The algorithm denoted by

$$e(g, g)^s \leftarrow \text{ReconstructSecret}(\text{T}, \{F_{y_1}, \ldots, F_{y_m}\}),$$

is used to reconstruct $q_{y_1}(0), \ldots, q_{y_m}(0)$.

This algorithm operates in a bottom-up manner, following these steps with respect to node x according to the structure of $\mathbb{T}$:

- $Sat(\{\text{att}(y_1), \ldots, \text{att}(y_n)\}, \mathbb{T}_x) = 0$: then F_x for node x is set null.
- $Sat(\{\text{att}(y_1), \ldots, \text{att}(y_n)\}, \mathbb{T}_x) = 1$: the following steps are executed:
    - x is a leaf node: $F_x = F_{y_j}(0) = e(g, g)^{q_{y_j}(0)}$ is set, where $x = y_j$ for some j.
    - x is an internal node with x_n child nodes: a set of indices S exists such that $|S| = k_x$, and $j \in S$ where $F(\{\text{att}(y_1), \ldots, \text{att}(y_n)\}, \text{T}_{x_j}) = 1$. In this case, F_x is set as follows:

$$F_x = \prod_{j \in S} F_{x_j}^{\Delta_{x_j}} = \prod_{j \in S} (e(g, g)^{q_{x_j}(0)})^{\Delta_{x_j}} = e(g, g)^{q_x(0)},$$

where $\Delta_{x_j} = \prod_{l \in S, l \neq j} \left(\frac{-j}{l-j}\right)$.

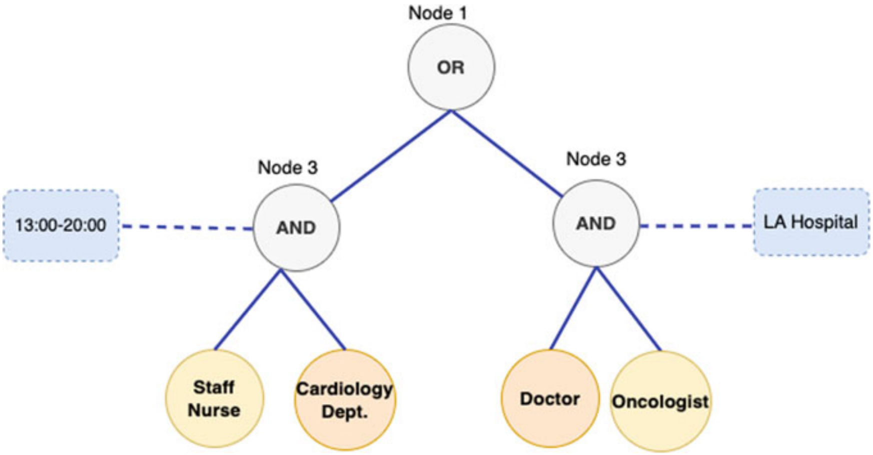


Fig. 6.4 Access Tree Structure

Upon completion of the algorithm, the root of T is associated with $F_{\text{root}} = e(g, h)^{q_{\text{root}}(0)} = e(g, g)^s$.

Access Policy Tree with Spatio-Temporal Constraints

A modified access tree $\mathbb{T}$ comprises policy tree nodes, attributes of user, and contextual trapdoors as depicted in Fig. 6.4. The user's normal attributes are present as leaf nodes and dynamic time and location attributes are present as contextual trapdoors which can be placed on any arbitrary node of the tree. Non-leaf nodes are threshold gates (NOT, AND, OR, etc.). Here, contextual constraints to limit the user's access rights based on the dynamic attribute (time and location) are present in the form of trapdoors. A data owner specifies the set of contextual constraints in the policy before encrypting the data, and access tokens are provided to the data user's requests to satisfy the constraint. In the shown access policy tree, a staff nurse from the cardiology department will be able to decrypt the ciphertext only during the 13:00–20:00 time interval.

Overview of Our CP-ABE Cryptosystem

Notation Used

We have introduced the notations used in the mathematical construction of the proposed cryptosystem in Table 6.1.

Table 6.1 Notation used

Variables	Description
λ	Security parameter
$\mathbb{G}_{\nvdash}, \mathbb{G}_1$	Cyclic multiplicative groups
p, g	Order and the generator element of group $\mathbb{G}$
$e : \mathbb{G}_{\nvdash} \times \mathbb{G}_{\nvdash} \rightarrow \mathbb{G}_T$	Bilinear map
H_1, H_2	One way hash functions
$\mathbb{F}_{GH}, \mathbb{F}_T$	Geohash location and access time format
LS, TS	Location and time secret
id_k, ID_a	Unique identifier of user and data owner respectively
PP, SP	Public and secret parameter
$\mathcal{K}_{se}$	Symmetric encryption key
$\mathcal{A}, \mathbb{T}$	Access policy and corresponding access tree structure
LL_y, TL_y	Location and time trapdoor appended to node y
$\mathcal{K}_{id_k}, \widetilde{\mathcal{K}_{id_k}}$	Decryption and transformed decryption key of user
LS'_x, TS'_x	Location and time secrets
$\mathbb{K}_{se}$	Shared secret key between the data owner and cloud storage
$\mathbb{A}_{id_k}$	User id_k attribute set

System Model

The system model of our proposed CP-ABE cryptosystem is depicted in Fig. 6.5. This framework consists of five fundamental entities, which are data owners, users, cloud data centers, fog servers, and the key generation authority.

1. Cloud Data Center: Cloud data centers provide storage infrastructure for encrypted EHRs. Additionally, it manages user authentication and authorization and directs the user request to the appropriate fog server for subsequent handling.
2. Fog Servers: The proximity of the fog server to the user devices allows for quicker processing and response times, enhancing the real-time nature of context verification. Along with this, these nodes help resource-constrained user devices by offloading resource-intensive computations.
3. Data Owners: Data owners are the patients and hospitals registered with e-healthcare systems. Data owners specify the access policy constraints that should be satisfied by the user to be able to access the data.

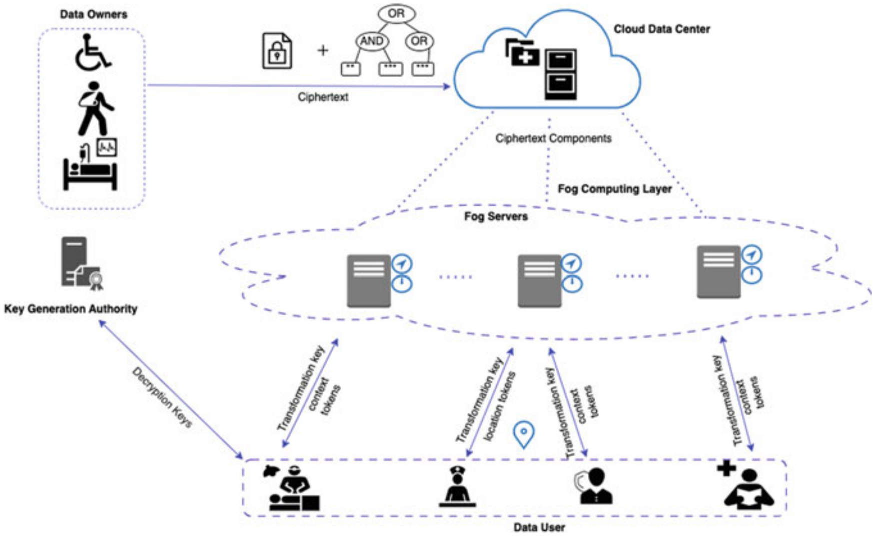


Fig. 6.5 System model of our proposed CP-ABE scheme

4. Key Generation Authority (KGA): KGA is a semi-trusted entity that generates user decryption keys based on the user attribute list.
5. Users: Users are the healthcare providers who want access to encrypted medical records. To be able to access the data user decryption key must satisfy the constraints of the access policy.

Security Assumptions

In our proposed cryptosystem, we consider the cloud data center and fog servers to exhibit semi-honest behavior, typically acting within legitimate boundaries but open to collusion with data users for unauthorized advantages. Due to the potential for unauthorized access, complete trust cannot be extended to data users, as collusion among them could lead to instances of unauthorized data retrieval. Key generation authority is treated as untrusted and anticipated to follow our prescribed protocol while aiming to extract as much information as possible.

Preventing Key Escrow Attacks

In [10] Chein et al. proposed an efficient pair-wise key scheme, which can be used by any two entities to compute a shared key between them without interactive communication. Let's consider two entities **A** and **B** which are operating under the same authority. **A** possesses the identity ID_A with public key as $PK_A = H(ID_A)$ and her private key is represented as $SK_A = PK_A^s = H(ID_A)^s$. **A** aims to maintain her anonymity while communicating with **B**, whose identity is denoted as ID_B and public key is $PK_B = H(ID_B)$. The static pair-wise key scheme works as follows:

A starts by selecting a random integer r from the set $\mathbb{Z}_p^*$ and generates the corresponding pseudonym P_A using its public key as $P_A = (PK_A)^r$. After that **A** shares its pseudonym P_A with **B**. Upon receiving the pseudonym from **A**, **B** computes the shared secret key using his private key as $K_{BA} = e(P_A, SK_B) = e(PK_A, PK_B)^{sr}$. Subsequently, **A** computes the shared secret using the pseudonym and public key of **B** as $K_{AB} = e(P_A, PK_B)^r = e(PK_A, PK_B)^{sr}$. This approach is very efficient in terms of communications and computations. This generated shared secret can be used to prevent key escrow attacks. The data owner and the cloud can establish a shared secret key between them to encrypt the data before storing it in the cloud. Consequently, the encrypted data stored over the cloud remains secure against decryption by the key generation authority since the shared secret key conceals the ciphertext.

Access Policy Obfuscation

In CP-ABE schemes, the access policy attached with ciphertext can reveal sensitive information, to prevent privacy leakages, attributes present in the access policy are masked. In CP-ABE cryptosystems based on a small universe, complete access policy obfuscation can be achieved through inner product predicate encryption, as discussed in [9]. In these schemes, the access structure $\mathcal{A}$ and the user attribute set are encoded as vectors with a total of $(n+1)^t$ elements, where t corresponds to the attribute types, and n indicates the attribute value count. The access policy, represented as a tree structure, designates attributes as leaf nodes, with non-leaf nodes denoting threshold gates. The predicate OR and AND with attributes A and B can be encoded as polynomials as $OR_{A,B}$: $P(x, y) = (x - A) \cdot (y - B)$ and $AND_{A,B}$: $P(x, y) = (x - A) + r(y - B),\ \ r \in Z_N$. The coefficient of the polynomials represents the access policy vector $\mathbf{v}$. Similarly, polynomials and corresponding coefficient vector $\mathbf{a}$ can be generated for the user attribute set. Here, the access policy $\mathcal{A}$ cannot be derived from the vector $\mathbf{v}$ hence ensures access policy privacy, whereas the $\mathbf{a} \cdot \mathbf{v} = 0$ can be used to check whether the attribute set satisfies the access policy.

Handling Spatio—Temporal Attributes

Before sending a request to the cloud for data access, the user converts his/her location geo-coordinates into a geohash [8] encoding and splits it into location prefix and suffix. Geohash strings serve as distinctive identifiers that represent a collection of latitude and longitude coordinates, offering efficiency in their handling. These strings exhibit a characteristic where geohash codes corresponding to proximate locations share common prefixes. The prefix of geohash encoding determines the rectangular region of its location, while the suffix represents the relative location within the rectangular region. When sending an access request to cloud, the data user shares the geohash of his/her location, where the suffix part is encrypted with a secret key. Cloud forwards the request to the nearest fog server based on location suffix. After that, the assigned fog server gets the secret key from the user to decrypt the location suffix part and then verify the user's location using suffix matching. We employ two distinct types of location attributes: GeoPoint and Geo-Area. Geo-Points store the user's specific location, while Geo-Area outlines the location constraints within the access policy. In our scheme, we make the assumption that user requests originate from a total of L locations. For each of these locations, a designated fog server is tasked with verifying whether the user's location falls within the defined region outlined by the Geo-Area constraint. To sum it up, we can conduct a streamlined containment check by examining the prefixes and suffixes. When a data requester seeks access to the encrypted data, the fog server verifies the location attributes and time of the requester to ensure they are within the authorized geographic areas and time specified in the access policies.

During the data encryption, the data owner specifies the spatio-temporal constraints corresponding to which time and location trapdoor components are added to ciphertext. These time and location trapdoor components are generated by the data owner using the public keys of the fog server. During the user's context verification process, the fog server utilizes its secret keys associated with both time and location to generate tokens for time and location, which are subsequently used to release the trapdoors.

Mathematical Construction

In this section, we offer an in-depth mathematical elaboration of the construction of our CP-ABE cryptosystem. The proposed system consists of five phases: System Setup, Key Generation, Data Encryption, and Data Decryption.

System Setup

The system takes security parameter 1^λ as an input and outputs the size of two cyclic multiplicative groups $\mathbb{G}_0$ and $\mathbb{G}_1$. Let p be the prime order and g be the generator element of group $\mathbb{G}_0$. Additionally, let $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$ denote a non-degenerate prime order bilinear map [30]. The scheme additionally uses two hash functions: $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}^*$ and $H_2 : \mathbb{G}_T^* \rightarrow \mathbb{Z}_p^*$.

KGA Initialization

KGA randomly chooses two integers $\alpha, \beta \in \mathbb{Z}_p^*$. It sets $h = g^\beta$. Subsequently, it generates the public and secret parameters as follows: $PP_{KGA} = (h, e(g, g)^\alpha)$ and $SP_{KGA} = (\beta, g^\alpha)$.

Cloud Server Initialization

Let ID_{CS} be the identity of the cloud storage center. Cloud server randomly chooses an integer $\gamma \in \mathbb{Z}_p^*$ to compute its public and secret parameters as $PP_{CS} = g^\gamma$ and $SP_{CS} = H_1(ID_{CS}^\gamma)$.

Fog Server Initialization

Assuming there are L locations supported by the system, then corresponding to each location, a fog server FS_m $(m \in 1, 2, \ldots, L)$ is deployed. Every FS_m randomly chooses γ_m^l as the location secret, and γ_m^t is its time secret. The public parameters PP_{FS_m} of a fog server FS_m are: $\{g^{\gamma_m^l}, g^{\gamma_m^t}\}$.

$\mathbb{F}_{GH}$ and $\mathbb{F}_T$ are the format description of geohash location string and time, respectively, supported by the fog server. Additionally, each data user is assigned a unique ID id_k.

Data Encryption

Encryption

The data owner with identity ID_a chooses a random key $\mathbb{K}_{se} \in \mathbb{G}_1$ to encrypt data $\mathbb{M}$ using a symmetric encryption scheme. Then, it defines an access policy $\mathcal{A}$ to encrypt the symmetric key $\mathbb{K}_{se}$. Suppose $\mathbb{T}$ is the access tree structure corresponding to $\mathcal{A}$.

The encryption algorithm is defined as follows: The encryption algorithm is defined as follows:

For the root node R, the algorithm chooses a random $s \in \mathbb{Z}_p$ as the root secret of the access tree. This secret is distributed among all the nodes in a top-down manner similar to approach [4]. Each node $x \in \mathbb{T}$ is associated with secret share q_x^0 whose value is computed using three parameters: q_x^1, s_x^τ and $s_x^\rho \in \mathbb{Z}_p$ where q_x^1 is the one shared from its parent node. s_x^ρ and s_x^τ are location and time-related secret parameters. For each node x with q_x^0, its values are defined based on the presence of time and location trapdoor as follows:

$$q_x^0 = \begin{cases} s_x^\rho \cdot s_x^\tau \cdot q_x^1, & \text{both location and time trapdoor} \\ s_x^\rho \cdot q_x^1, & \text{location trapdoor} \\ s_x^\tau \cdot q_x^1, & \text{time trapdoor} \\ q_x^1, & \text{otherwise} \end{cases}$$

For a location constraint $l \in \mathbb{F}_{\mathbb{GH}}$ defined for node y in the access tree, a location trapdoor LT_y is defined using a random secret $r_l \in \mathbb{Z}_p$ and public key of the nearest fog server FS_m where $LT_y = \{A_y^l = g^{r_l},\ B_y^l = s_y^\tau + H_2(e(H_1(l),\ g^{\gamma_m^l})^{r_l})\}$. Similarly for a time constraint $t \in \mathbb{F}_{\mathbb{T}}$ is defined as $TT_y = (A_y^t = g^{r_t},\ B_y^t = s_y^\tau + H_2(e(H_1(t),\ g^{\gamma_m^t})^{r_t})$, where, $r_t \in \mathbb{Z}_p$ is randomly chosen secret. Let y be a node from the leaf node set $\mathcal{Y}$. So for each node y the algorithm computes: $C_y = g^{q_y^1}$ and $C'_y = H_1(Att(y))^{q_y^1}$.

To prevent a key escrow attack, the data owner chooses a random value $a \in \mathbb{Z}_p$ and calculates a shared secret between him/her and cloud as $\mathbb{K}_{ss} = e((g^\gamma)^a, H_1(ID_{CS}))$. This $\mathbb{K}_{ss}$ is used to blind the ciphertext. Additionally, data owner generates a vector $\mathbf{v}$ corresponding to the access structure to obfuscate its attribute before storing the ciphertext on the cloud. Finally, it generates a ciphertext CT consisting of the following components,

$$\begin{aligned} CT = \{\mathbf{v}, \bar{C} = Enc_{\mathbb{K}_{se}}(\mathbb{M}),\ C = \mathbb{K}_{se} \cdot \mathbb{K}_{ss} \cdot e(g,g)^{\alpha s},\ C' = g^s, \forall y \in \mathcal{Y}:\ C_y,\ C'_y; \\ \forall LL_y \in \mathbb{T} : LL_y = (A_y^l,\ B_y^l); \\ \forall TL_y \in \mathbb{T} :\ TL_y = (A_y^t,\ B_y^t)\}\} \end{aligned}$$

The data owner uploads $(P = g^a, CT)$ to the cloud storage server.

Key Generation

A data user with id_k and attribute set $\mathbb{A}_{id_k}$ requests his decryption key from KGA. The key generation authority generates the user decryption key based on the static attributes of the user.

User Key Generation

In this phase, KGA randomly selects integer $r_k \in \mathbb{Z}_p$ for the user id_k. For each attribute $i \in \mathbb{A}_{id_k}$, it chooses random $r_i \in \mathbb{Z}_p$. Then, it computes a set of key components and outputs a personalized decryption key for a user as

$$\mathcal{K}_{id_k} = \{D = g^{\frac{(\alpha+r_k)}{\beta}}, \forall i \in \mathbb{A}_{id_k} : D_i = g^{r_k} \cdot H_1(i)^{r_i}, D_i' = g^{r_i}, D_i'' = H_1(i)^{\beta}\}.$$

Delegate Key Generation

To outsource partial decryption of ciphertext to the fog server, the user randomly selects an integer $b \in \mathbb{Z}_p$ and computes the transformed version of the decryption key $\mathcal{K}_{id_k}$ as:

$$\widetilde{\mathcal{K}_{id_k}} = \{\widetilde{D} = D^{1/b},\ \forall i \in \mathbb{A}_{id_k} : \widetilde{D_i} = D_i^{1/b}, \widetilde{D_i'} = D_i'^{1/b}\}$$

While requesting partial decryption, the user shares the delegation key $\widetilde{\mathcal{K}_{id_k}}$ with the fog server.

Data Decryption

When a data user id_k with a set of attributes $\mathbb{A}_{id_k}$ wants to access the ciphertext from the cloud storage, it generates a vector $\mathbf{a_k}$ using the polynomial corresponding to the attribute set $\mathbb{A}_{id_k}$. After receiving the user request cloud verifies whether the received attribute set vector $\mathbf{a_k}$ fulfills the (obfuscated) access policy vector $\mathbf{v}$ by computing $\mathbf{a_k} \cdot \mathbf{v}$. After successful verification, cloud computes $\mathcal{K}_{ss} = e(P, SP_{CS}) = e(g^a, H_1(ID^{\gamma}_{CS}))$ and updates the component C in CT as $C' = C/\mathbb{K}_{ss} = \mathbb{K}_{se} \cdot e(g, g)^{\alpha s}$ and forwards the updated CT and encrypted location suffix to the nearest fog server. On receiving the partial decryption request, the fog server verifies the time and location of the user to generate corresponding tokens.

Upon successful verification of the user request, the data user sends its delegate key $\widetilde{\mathcal{K}_{id_k}}$ to the nearest fog server FS_m.

Token Generation

Let $t_{id_k} \in \mathbb{F}_t$ be the DU's time of access and $l_{id_k} \in \mathbb{F}_{GH}$ as the location value. The fog server first confirms the access request time and performs the suffix matching to verify the user's context. After this, the FS_m generates the time and location token corresponding to the user context as $TT_{id} = (H_1(t_{id_k}))^{\gamma_m^t}$ and $LT_{id} = (H_1(l_{id_k}))^{\gamma_m^l}$ respectively. To retrieve the time and location secrets for node $x \in \mathbb{T}$, the FS_m uses the trapdoors defined in the CT as follows:

$$LS'_x = B_x^l - H_2(e(TT_{id},\ A_x)) = s_x^\rho$$

$$TS'_x = B_x^t - H_2(e(LT_{id},\ A_x)) = s_x^\tau$$

Fog Decryption

To perform partial decryption fog server defines a function DecryptNode(). If y is a leaf node, let $i = Attribute(y)$. Then, DecryptNode(y) function is defined as

$$F_y = (\frac{e(\widetilde{D_i}, C_y)}{e(\widetilde{D'_i}, C'_y)} = e(g,\ g)^{\frac{r_k \cdot q_y^1}{b}} = e(g,\ g)^{\frac{r_k \cdot q_y^0}{b}}$$

The leaf node with the attached location trapdoor is decrypted as

$$F_y = \frac{e(\widetilde{D_i}, C_y)}{e(\widetilde{D'_i}, C'_y)})^{s_y^\rho} = e(g,\ g)^{\frac{\beta_j q_y^1 s_y^\rho}{b}} = e(g,\ g)^{\frac{\beta_j q_y^0}{b}}$$

The leaf node with attached time trapdoor is decrypted as

$$F_y = \frac{e(\widetilde{D_i}, C_y)}{e(\widetilde{D'_i}, C'_y)})^{s_y^\tau} = e(g,\ g)^{\frac{\beta_j q_y^1 s_y^\tau}{b}} = e(g,\ g)^{\frac{\beta_j q_y^0}{b}}$$

The leaf node with the attached time and location trapdoor is decrypted as

$$F_y = \frac{e(\widetilde{D_i}, C_y)^{s_y^\rho s_y^\tau}}{e(\widetilde{D'_i}, C'_y)} = e(g,\ g)^{\frac{\beta_j q_y^1 s_y^\rho s_y^\tau}{b}} = e(g,\ g)^{\frac{\beta_j q_y^0}{b}}$$

Then, it recursively computes DecryptNode(x) for every node $x \in \mathbb{T}$ in a bottom-top manner similar to [4]. If $\mathbb{A}_{id_k}$ satisfies the access structure $\mathbb{T}$ then F_{Root} is calculated as: $e(g, g)^{r_k \cdot s/b}$ otherwise, it outputs null. Using F_{Root}, it computes the blinded value of decryption key as $\widetilde{\mathbb{K}_{se}} = \frac{e(C,\widetilde{D})}{F_{Root}} = \frac{e(g,g)^{\frac{(\alpha+r_k)s}{b}}}{e(g,g)^{r_k/b}} = e(g,\ g)^{\frac{\alpha s}{b}}$

User Decryption

Once the user receives the partially decrypted symmetric key, it computes the plaintext key by calculating $(\mathbb{K}_{se})^b$ and finally decrypts $\bar{C}$ to obtain the message $\mathbb{M}$.

Security Analysis

In this section, we present the security features of our proposed CP-ABE cryptosystem.

Security Against Users Collusion Attack

In the key generation process, each user is allocated a unique random secret denoted as r_k, which is employed to obfuscate the key components generated based on the user's attribute set. In scenarios where groups of users attempt to combine their attribute sets with the intention of gaining unauthorized access, the forging of decryption keys becomes unfeasible for the newly merged attribute set. This impossibility arises from the inherent randomization integrated into the key component generation procedure, rendering the combination of components from two distinct decryption keys invalid. Consequently, our cryptosystem effectively thwarts user-initiated key collusion attacks.

Safeguarding Privacy in the Context of Fog Servers

During the partial decryption process, the fog server computes the blinded value of the decryption key as $\widetilde{\mathbb{K}_{se}} = e(g,\ g)^{\frac{\alpha s}{b}}$. However, to decrypt the ciphertext the the secret parameter b is needed which is known to the valid user only. Thus, the ciphertext cannot be decrypted by the fog server even if he possesses the partially decrypted key.

Access Policy Privacy

When a data user transmits encrypted data to the cloud storage, the access policy is vectorized as $\mathbf{a_r}$. This obfuscation ensures that only authorized users possessing valid coefficient vector $\mathbf{v}$ can satisfy the equation $\mathbf{a_r} \cdot \mathbf{v} = 0$. Guessing $Attribute(y)$ from the polynomial coefficient vector is infeasible by unauthorized entities due to the randomness used while creating the polynomial.

Security Against Key Escrow Attack

The key generation authority is responsible for generating and distributing decryption keys to all users and possesses the capability to decrypt each ciphertext stored on the cloud storage by computing the relevant secret key. However, due to the key secrecy property of the pair-wise shared secret key protocol [10], the component C of CT is concealed by $\mathbb{K}_{ss}$, which can only be established between the data owner and the cloud server. In addition, during the partial decryption process, the fog server generates $e(g, g)^{r_k s/b}$, which is blinded by random secret b known only to the user, thus prevents the KGA from decrypting it.

Revocation for Unsatisfied Time and Location

A data user who can decrypt a ciphertext associated with location X cannot employ the same token to decrypt another ciphertext originating from the same location X. Furthermore, access privileges are automatically withdrawn when users change their locations. If a data user fails to satisfy time and location constraints, correct values of time and location tokens are not generated. Likewise, the time and location secrets s_y^{τ} and s_y^{ρ} cannot be calculated correctly by the fog server. Without these values, F_y cannot be calculated in the partial decryption algorithm.

Performance Analysis

Functional Properties

We present a detailed feature-wise comparison between our CP-ABE scheme and several existing access control cryptosystems, including [4, 17, 20, 22, 33], all of which tackle the challenge of managing dynamic user attributes. As Table 6.2 illustrates, none of the currently available schemes simultaneously accommodates both static and contextual attributes, prevents key escrow attacks, and conceals the sensi-

Table 6.2 Functional comparison

Features	Ref. [4]	Ref. [17]	Ref. [20]	Ref. [22]	Ref. [33]	Our scheme
Fine grained access control	✓	×	✓	✓	✓	✓
Time attribute	×	×	×	✓	✓	✓
Location attribute	×	✓	✓	✓	✓	✓
Revocation for unsatisfied Context	×	✓	✓	×	×	✓
Hidden access policy	×	×	×	×	×	✓
Preventing key escrow attack	×	×	×	×	✓	✓
Location privacy	×	✓	×	×	×	✓
Outsourcing decryption	×	×	×	✓	×	✓

tive access policy. Furthermore, our system ensures the confidentiality of data user's location information while eliminating the need for an extra mechanism to revoke the location and time attributes of users in case of contextual changes. Therefore, our scheme is usable in practical applications with different access control features.

Complexity Analysis

We conduct a computational complexity comparison between our cryptosystem and the fundamental CP-ABE scheme [4], as well as the scheme mentioned in [33]. The outcomes are presented in Table 6.3. Our complexity assessment considers bilinear pairing and exponentiation operations within prime order groups $\mathbb{G}$ and $\mathbb{G}_T$. Consider $C_{E_{\mathbb{G}}}$ and $C_{E_{\mathbb{G}_{\mathbb{T}}}}$ as the computational cost associated with exponentiation operations in groups $\mathbb{G}$ and $\mathbb{G}_T$, respectively. Similarly, C_B denotes the cost of bilinear pairing operations. Let N_A and N_U represent the collective count of attributes within the access policy and the user access privileges, respectively. We have assumed single location and time trapdoor are associated with the access policy. Our complexity comparison yields the following observations: (i) During the generation of the delegation key, the data user doesn't need to perform any bilinear pairing operations to generate

Table 6.3 Algorithmic complexity

Algorithms	Ref. [4]	Ref. [33]	Our cryptosystem
Encryption	$C_{E_{\mathbb{G}}}(N_A+1)+C_{E_{\mathbb{G}_{\mathbb{T}}}}$	$E_{\mathbb{G}}(2+N_A)+$ $2C_B+C_{E_{\mathbb{G}_{\mathbb{T}}}}$	$C_{E_{\mathbb{G}}}(2+N_A)+$ $2C_B+C_{E_{\mathbb{G}_{\mathbb{T}}}}$
Key escrow attack avoidance	–	$6E_{\mathbb{G}}$	$C_B+C_{E_{\mathbb{G}}}$
Policy obfuscation	–	–	$C_B N_A+C_{E_{\mathbb{G}}}$
Indices generation	–	–	$N_U(C_B+2C_{E_{\mathbb{G}}})$
Key delegation	–	–	$2C_{E_{\mathbb{G}}}N_U$
Partial decryption	–	–	$C_B(2N_U+1)+C_{E_{\mathbb{G}}}$
User decryption	$C_B(2N_U+1)+C_{E_{\mathbb{G}}}$	$N_U(C_B+C_{E_{\mathbb{G}}})$	$N_U C_{E_{\mathbb{G}}}$

a transformed version of the key for the fog server. (ii) Fog server needs to perform a single bilinear pairing operation and exponentiation operations (in multiples of N_U)to generate the partially decrypted ciphertext which ensures faster response time. (iii) During the final decryption phase, users require only exponentiation operations in $E_{\mathbb{G}}$ to decrypt the symmetric key and obtain the plaintext. Moreover, a substantial part of the decryption process, including the validation of user contextual data and the creation of partially decrypted ciphertext, is delegated to the fog server within our approach. This strategic allocation alleviates the computational burden on the user's side.

Experimentation Analysis

We have implemented our proposed cryptosystem using the Java Pairing-Based Cryptography (jPBC) library [34, 35]. We performed experiments on a system with an Intel quad-core processor, 16 GB RAM, and Ubuntu 14:04:5 LTS operating system platform. Python 3.4 and jPBC version 2.0.0 were used for implementation. Figure 6.6 shows the computation time with a varying number of user attributes, access policy attributes, and contextual trapdoors. We calculated the average computational time of algorithms across various phases of our proposed cryptosystem. Figure 6.6a shows that key generation depends linearly on the number of user attributes, whereas setup doesn't vary much as it generates the public and private parameters of key generation authority, cloud server, and fog server based on systems public parameters. Figure 6.6b shows that keeping the number of contextual constraints and attributes in the access policy constant, key generation time increases linearly with the number of attributes due to the increased number of key components. Figure 6.6c shows that keeping the number of user attributes and the access policy's static attribute size constant, encryption and decryption time increases linearly with the number of contextual constraints in the access policy. It can be observed that costly operations

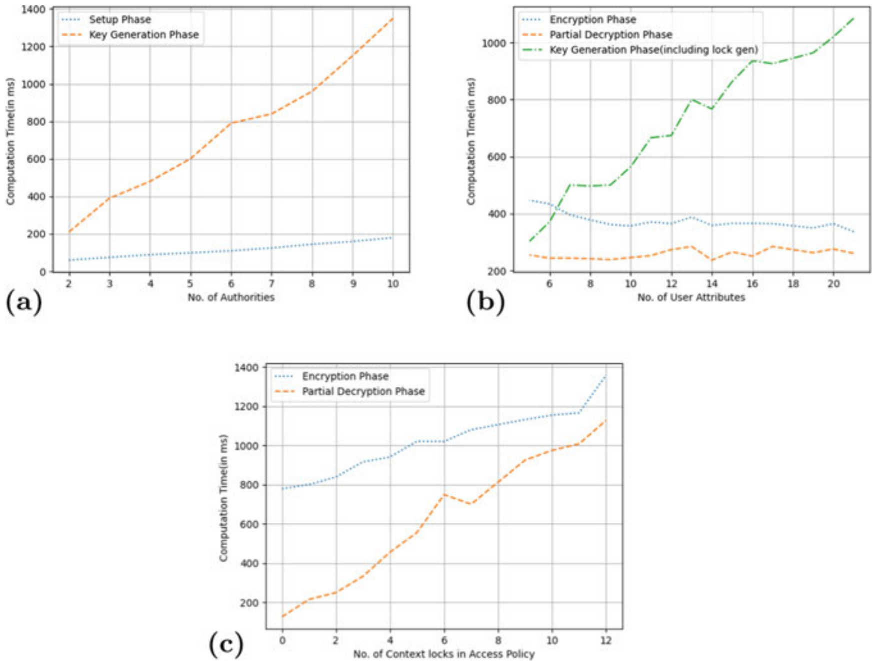


Fig. 6.6 Experimental results: **a** system setup and key generation time with varying number of user attributes, **b** data encryption, partial decryption and key generation time with varying number of user attributes (fixed: no. of the attribute in the policy = 5, time trapdoor = 1, location trapdoor = 1), **c** encryption, decryption time with varying number of contextual trapdoors (no. of static attributes = 20)

under the verification and decryption phase are outsourced from end users to the fog servers. This makes our scheme usable for different users with limited computing resources.

Conclusion

The rapid advancement of collaborative e-healthcare systems has brought significant challenges in the secure access and management of sensitive patient data stored on the cloud. We have presented a privacy-preserving spatio-temporal access control scheme that offers a robust and efficient access control solution for securing sensitive EHR data stored on the cloud. By considering both static parameters (user roles) and dynamic parameters (location and time), fine-grained access control ensures that users have appropriate access privileges aligned with their roles and the current operational context. Additionally, our systems prevent key escrow attacks and preserve the privacy of the access policy. This enhances system security and operational efficiency in the collaborative e-health environment. To aid user devices with limited resources, significant computations are delegated to fog servers. Our cryptosystem has been proven to meet diverse security criteria. The performance analysis showcases the efficiency and efficacy of our CP-ABE scheme, which outperforms the existing access control cryptosystem.

References

1. Abbas A, Khan SU (2014) A review on the state-of-the-art privacy-preserving approaches in the e-health clouds. IEEE J Biomed Health Inf 18(4):1431–1441. https://doi.org/10.1109/JBHI.2014.2300846
2. Announcing the advanced encryption standard (AES) (PDF). FIPS 197. NIST. November 26, 2001
3. Diffie W, Hellman ME (1977) Exhaustive cryptanalysis of the NBS data encryption standard. Computer 10(6):74–84
4. Bethencourt J, Sahai A, Waters B (2007) Ciphertext-policy attribute-based encryption. In: IEEE symposium on security and privacy, pp 321–334
5. Khuntia S, Kumar PS (2018) New hidden policy CP-ABE for big data access control with privacy-preserving policy in cloud computing. In: 2018 9th international conference on computing, communication and networking technologies (ICCCNT), Bengaluru, India, pp.1–7. https://doi.org/10.1109/ICCCNT.2018.8493698
6. Zhang X, Jin C, Wen Z, Shen Q, Fang Y, Wu Z (2015) Attribute-based encryption without key Escrow. In: Huang Z, Sun X, Luo J, Wang J (eds) Cloud computing and security. ICCCS 2015. Lecture notes in computer science, vol 9483. Springer, Cham. https://doi.org/10.1007/978-3-319-27051-7
7. Green M, Hohenberger S, Waters B (2011) Outsourcing the decryption of ABE ciphertexts. In: Proceedings of the 20th USENIX conference on security (SEC'11). USENIX Association, USA, 34
8. Niemeyer G (2008) Geohash. Website: https://geohash.org
9. Lai J, Deng RH, Li Y (2011) Fully secure cipertext-policy hiding CP-ABE. In: Bao F, Weng J (eds) Information security practice and experience. ISPEC 2011. Lecture notes in computer science, vol 6672. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-21031-0
10. Chien H-Y, Lin R-Y (2006) Identity-based key agreement protocol for mobile ad-hoc networks using bilinear pairing. In: IEEE international conference on sensor networks, ubiquitous, and trustworthy computing (SUTC'06), Taichung, Taiwan, pp 8. https://doi.org/10.1109/SUTC.2006.1636220
11. Weber SG (2009) Securing first response coordination with dynamic attribute-based encryption. In: Privacy, security, trust and the management of e-business, CONGRESS'09. World Congress on IEEE 2009, pp 58–69
12. Zhu Y, Hu H, Ahn G-J, Huang D, Wang S (2012) Towards temporal access control in cloud computing. In: Proceedings IEEE INFOCOM. Orlando, FL, USA, pp 2576–2580. https://doi.org/10.1109/INFCOM.2012.6195656
13. Zhu Y, Hu H, Ahn G-J, Yu M, Zhao H (2012) Comparison-based encryption for fine-grained access control in clouds. In: Proceedings of the second ACM conference on data and application security and privacy (CODASPY '12). Association for Computing Machinery, New York, NY, USA, pp 105–116. https://doi.org/10.1145/2133601.2133614
14. Weber SG (2012) A hybrid attribute-based encryption technique supporting expressive policies and dynamic attributes. Inf Secur J A Glob Perspect 21(6):297–305
15. Doshi N, Jinwala D (2013) Updating attribute in CP-ABE: a new approach. In: IJCA proceedings on international conference in distributed computing and internet technology, vol ICDCIT, pp 23–28

16. Balani N, Ruj N (2014) Temporal access control with user revocation for cloud data. In: IEEE 13th international conference on trust, security and privacy in computing and communications, pp 336–343
17. Shao J, Lu R, Lin X (2014) Fine: a fine-grained privacy-preserving location-based service framework for mobile devices. In: 33rd annual IEEE international conference on computer communications, pp 244–252
18. Wang Z, Huang D, Zhu Y, Li B, Chung C-J (2015) Efficient attribute-based comparable data access control. IEEE Trans Comput 64(12):3430–3443. https://doi.org/10.1109/TC.2015.2401033
19. Denisow I, Zickau S, Beierle F, Küpper A (2015) Dynamic location information in attribute-based encryption schemes. In: 2015 9th international conference on next generation mobile applications, services and technologies, Cambridge, UK, pp 240–247. https://doi.org/10.1109/NGMAST.2015.63
20. Xue Y, Hong J, Li W, Xue K, Hong P (2016) LABAC: a location-aware attribute-based access control scheme for cloud storage. In: IEEE GLOBECOM, pp 1–6
21. Baseri Y, Hafid A, Cherkaoui S (2016) K-anonymous location-based fine-grained access control for mobile cloud. In: IEEE annual consumer communications and networking conference (CCNC), pp 720–725
22. Liu Z, Jiang ZL, Wang X, Yiu SM, Zhang R, Wu Y (2018) A temporal and spatial constrained attribute-based access control scheme for cloud storage. In: IEEE (TrustCom/BigDataSE, pp 614–623
23. Baseri Y, Hafid A, Cherkaoui S (2018) Privacy-preserving fine-grained location-based access control for mobile cloud. In: Computers and security, vol 73, pp 249–265. ISSN 0167-4048. https://doi.org/10.1016/j.cose.2017.10.014
24. Hong J et al (2020) TAFC: time and attribute factors combined access control for time-sensitive data in public cloud. IEEE Trans Serv Comput 13(1):158–171
25. Alshehri S, Radziszowski SP, Raj RK (2012) Secure access for healthcare data in the cloud using ciphertext-policy attribute-based encryption. In: 2012 IEEE 28th international conference on data engineering workshops, Arlington, VA, USA, pp 143–146. https://doi.org/10.1109/ICDEW.2012.68
26. Edemacu K, Jang B, Kim JW (2021) CESCR: CP-ABE for efficient and secure sharing of data in collaborative ehealth with revocation and no dummy attribute. PLoS ONE 16(5): e0250992. https://doi.org/10.1371/journal.pone.0250992
27. Fabian B, Ermakova T, Junghanns P (2015) Collaborative and secure sharing of healthcare data in multi-clouds. Inf Syst 48:132–150
28. Fan K, Huang N, Wang Y, Li H, Yang Y (2015) Secure and efficient personal health record scheme using attribute-based encryption. In: 2015 IEEE 2nd international conference on cyber security and cloud computing, New York, NY, USA, pp 111–114. https://doi.org/10.1109/CSCloud.2015.40
29. Ghosh B, Parimi P, Rout RR (2020) Improved attribute-based encryption scheme in fog computing environment for healthcare systems. In: 2020 11th international conference on computing, communication and networking technologies (ICCCNT), Kharagpur, India, pp 1–6. https://doi.org/10.1109/ICCCNT49239.2020.9225606
30. Freeman DM (2010) Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In: Gilbert H (eds) Advances in cryptology - EUROCRYPT 2010. EUROCRYPT 2010. Lecture notes in computer science, vol 6110. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-13190-5
31. Sahai A, Waters B (May2005) Fuzzy identity-based encryption. In: Proceedings of the EUROCRYPT. LNCS, vol 3494, pp 457–473
32. Goyal V, Pandey O, Sahai A, Waters B (2006) Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM conference on computer and communications security. ACM, pp 89–98
33. Arfaoui A, Cherkaoui S, Kribeche A, Senouci SM (2020) Context-aware adaptive remote access for IoT applications. IEEE Internet Things J 7(1):786–799

34. De Caro A, Iovino V (2011) JPBC: Java pairing based cryptography. In: Proceedings of the 16th IEEE symposium on computers and communications, ISCC 2011, IEEE, Kerkyra, Corfu, Greece, June 28–July 1, pp 850–855
35. Lynn B (2007) PBC library: the pairing-based cryptography library. https://crypto.stanford.edu/pbc/

Chapter 7
Advances in Differential Privacy and Differentially Private Machine Learning

Saswat Das and Subhankar Mishra

Abstract There has been an explosion of research on differential privacy (DP) and its various applications in recent years, ranging from novel variants and accounting techniques in differential privacy to the thriving field of differentially private machine learning (DPML) to newer implementations in practice, like those by various companies and organisations such as census bureaus. Most recent surveys focus on the applications of differential privacy in particular contexts like data publishing, specific machine learning tasks, analysis of unstructured data, location privacy etc. This work thus seeks to fill the gap for a survey that primarily discusses recent developments in the theory of differential privacy along with newer DP variants, viz. Renyi DP and Concentrated DP, novel mechanisms and techniques, and the theoretical developments in differentially private machine learning in proper detail. In addition, this survey discusses its applications to privacy-preserving machine learning in practice and a few practical implementations of DP.

Keywords Differential privacy · Privacy-preserving machine learning · Trustworthy AI

Introduction

The explosion of popularity and adoption of fields like machine learning and big data, and powerful data processing machinery has meant that high quality data is considered to be among the most valuable, high utility commodities. This data, which often includes sensitive details about certain individuals and entities, helps demographers draw useful information about a population and socioeconomic distribution across

S. Das (✉) · S. Mishra
National Institute of Science Education and Research, an OCC of Homi Bhabha National Institute, Odisha, India
e-mail: saswat.das@niser.ac.in

S. Mishra
e-mail: smishra@niser.ac.in

D. Gountia et al. (eds.), *Information Technology Security*, Springer Tracts in Electrical and Electronics Engineering, https://doi.org/10.1007/978-981-97-0407-1_7

an area of land, helps tech companies analyse the usage habits of and issues faced by users to design updates to their products, and helps medical professionals to improve upon diagnostic systems and medical care, to understand diseases better, create medical data visualisations, etc. Companies like Netflix and YouTube often utilise data to provide personalised content recommendations for their users.

But as a corollary, this has enabled the extraction of certain, potentially sensitive, information about the individuals in databases (a.k.a. *data subjects*) unless protected in some form. This sensitive information can be used to the detriment of the concerned data subjects by entities like insurance companies that could use sensitive data on whether someone has a particular ailment or habit to increase their insurance premiums or deny them insurance and thus violate legislations like HIPAA that deal with such concerns about sensitive medical data, by other individuals or agencies to blackmail them or track their activities/movements, by governments or political agencies to gain sensitive data on citizens etc. This has naturally led to privacy concerns. Well known attacks like the linkage attack on the medical records released by the Massachusetts Group Insurance Commission [106], and that that on the Netflix Prize database [82] respectively compromised the medical records of government employees in the state of Massachusetts in the 1990s and the private content consumption data of Netflix viewers in 2006. Very prominently the reconstruction attack on the 2010 US Census data [45] was able to reconstruct the private microdata of a significant proportion of American citizens from deidentified and publicly available census data. Kasiviswanathan, Rudelson, and Smith [64] (2012) demonstrated that linear reconstruction attacks can be successful in various, including some seemingly “non-linear", settings, including when applied to a large class of ERM algorithm outputs, including linear regression and logistic regression. In addition to these well known attacks, more recent ones like that on Myki transportation records released by Public Transport Victoria in 2018 [19] alarmingly showed that it was possible to trace an individual movements on Melbourne’s transportation from these deidentified records by just using two randomly selected touch events/data points.

This naturally leads to questions such as what is individual privacy, how it can be breached, and how one can go about protecting it while still allowing analysts to draw useful conclusions. In the 1970s, statistical agencies and computer scientists proposed a set of privacy preserving techniques collectively known as *statistical disclosure limitation* (SDL); this called for “anonymising data via methods like top-coding, noise injection into certain attributes in the database, and swappng of attributes among rows. More recent techniques have been proposed, either in light of incidents like these, or in anticipation of certain privacy threats. For instance, the linkage attack on the Massachusetts Group Insurance Commission’s medical records led to k-anonymity[99] being introduced by Samarati and Sweeney to allow for a level of indistinguishability within a database. k-anonymity however happens to be vulnerable to certain linkage attacks anyway; and therefore sophisticated variants of k-anonymity were introduced subsequently to improve on it, viz. t-closeness, m-invariance, or l-diversity. Even so, these are still vulnerable to composition attacks [44].

However, no notion of privacy provided for an objective assessment of the privacy guarantee granted in the process, and any guarantee thus granted by previously existing techniques or definitions of privacy differed among various contexts. In addition, most of the aforementioned techniques suffered from certain vulnerabilities.

This state of affairs continued until a powerful class of attacks, known as reconstruction attacks, was introduced by Dinur and Nissim in 2003 [23]. Studying these attacks led directly to the introduction of *differential privacy* (often abbreviated as DP) by Dwork et al. in 2006 [28], often called the gold standard of privacy , which provided for a mathematically precise and quantifiable notion of privacy. This notion rests on the idea given by the Fundamental Theory of Information Recovery [30], which states that providing overly accurate responses to too many queries will inevitably lead to a privacy catastrophe. Thus, differential privacy entails the protection of individual privacy to a large extent by perturbing responses to queries made on a database while still allowing high accuracy of responses and subsequent analysis. The privacy guarantees themselves can be shown in an objective and mathematically rigorous manner.

Related Work

There is a rich trove of literature on differential privacy, various facets of which have been well studied and surveyed in the past. Table 7.1 compares different surveys with respect to ours.

There have been surveys and detailed tutorials that address differential privacy from a technical point of view, like ones by Dwork [26], Dwork and Roth [30], Vadhan [113] and [61] which focus on a rigorous treatment of differential privacy; in particular tutorials/textbooks like [30, 113] focus mostly on providing a rigorous introduction to the principles of differential privacy, and the basic results, ideas, mechanisms pertaining to it.

Sengupta et al. [102] presented a more recent survey in 2020 which discusses the theory and ideas behind differential privacy, learning with differential privacy and a few industrial deployments of the same. Wang et al. [114] present an in depth and specific survey on local differential privacy, the theory behind it, its mechanisms and models, variants etc. Xiong et al. [122] published a survey on local differential privacy in 2020, taking an general and fairly technical look at local differential privacy, pertinent mechanisms, and its applications to privacy-preserving statistical analysis, machine learning, practical deployments, etc. Fioretto et al. [40] (2022) discuss the interplay between differential privacy and fairness, including in the context of differentially private machine learning or decision-making using differential privacy, and how differential privacy affects fairness in these settings. Boulemtafes et al. [8] (2020) presented a survey on how to train and release deep learning models in a privacy preserving manner in general, and without specific focus on differential privacy.

Other related surveys address different facets of privacy, or that in certain contexts. Dwork et al. in [33] discuss attacks on databases, chiefly reconstruction attacks and tracing attacks, and further motivate the need for privacy-preserving data analysis. Sarwate and Chaudhuri's survey [100] focuses on differentially private techniques for continuous data for use in signal processing and machine learning. Cunha et al. [20] present a survey of privacy preserving mechanisms in general, and Liu et al. [70] discuss privacy in machine learning in general, but only these surveys only discuss differentially private methods briefly. Gong et al. [50] (2020) presented a high-level discussing differentially private machine learning exclusively, without delving into technical details rigorously. Zhang et al. [126] focus on discussing differential privacy in conjunction with game theory, emphasising on game-theoretic solutions to various problems and game-theoretic mechanism design. Zhao and Chen (2020) [127] discuss various applications of differential privacy for privacy preserving analysis of unstructured data, viz. images, video, audio etc.

Surveys by Jiang et al. [58] and Fung et al. [43] discuss location privacy and privacy-preserving data publishing respectively, and include brief discussions on differential privacy as a tool in said contexts.

Our Contribution

The works (viz. [30]) cited above include some seminal ones and remain a vital introduction to the study of differential privacy. But the recent surveys on differential privacy literature that the authors have encountered are either introductory in nature or focus only on a specific context of use or facet of differential privacy, or they treat differential privacy as a supplement for certain application, without paying much attention to the technical aspects of it. Surveys discussing the most novel methods, variants and developments as of the moment do not exist. In addition, newer privacy loss accounting techniques, variants of differential privacy, and several applications to fields like machine learning have been introduced in recent years. This marked a need for a newer, more general, and broader survey on some facets of the rapidly expanding and recent literature on differential privacy, in particular novel variants of DP, applications to machine learning and data analysis, tighter bounds and deployments, while bringing back some emphasis to central differential privacy. This is a need this survey seeks to fulfill.

Contributions of our survey:

- We review basic definitions and ideas central to the theory and study of differential privacy briefly for readers who might not be very acquainted with the subject matter.
- We then discuss basic mechanisms that are used to implement differential privacy, recent results and privacy analysis, and some variants of differential privacy, the motivation behind them and their salient features.

Table 7.1 Table comparing similar surveys with this work

Survey	NoVR	DPML	NoDP	CoUs	Remarks/focus
Dwork [26]	×	×	×	×	First survey on early advances in DP
Fung et al. [43]	×	×	×	✓	Privacy in Data-Publishing
Sarwate and Chaudhuri [100]	×	✓	×	✓	DP in Signal Processing and ML
Dwork and Roth [30]	×	✓	×	×	Seminal primer on DP discussing various aspects of it and relevant algorithms and applications
Vadhan [113]	×	×	✓	×	Theoretically rigorous
Dwork et al. [33]	×	×	×	✓	Privacy attacks on databases
Wang et al. [114]	✓	✓	✓	✓	LDP
Xiong et al. [122]	✓	✓	✓	✓	LDP
Sengupta et al. [101]	×	✓	✓	×	Learning with DP, Implementations
Gong et al. [50]	×	✓	×	✓	High-level discussion on applied DPML
Boulemfates et al. [8]	×	✓	×	×	General survey on privacy preserving deep learning
Jiang et al. [58]	×	×	×	✓	Privacy Preserving Mechanisms for Location Based Services
Liu et al. [70]	×	✓	×	✓	Privacy Preserving ML (not restricted to DPML)
Cunha et al. [20]	×	×	×	×	Privacy Preserving Mechanisms (including DP) in general
Zhang et al. [126]	×	×	×	✓	Game Theory with DP
Zhao and Chen [127]	×	×	×	✓	Analysis of Unstructured Data with DP
Fioretto et al. [40]	×	✓	×	✓	Intersection of Fairness and DP
Das and Mishra	✓	✓	✓	×	**Our Survey**

The adjective “novel” in the table refers to anything that has been introduced in or after 2015. A ✓ is awarded if a survey makes more than a brief mention of a topic and discusses it in some detail. NoVR—Novel Variants of Differential Privacy, DPML—Differential Privacy in Machine Learning, NoDP—Novel DP mechanisms and techniques, CoUs—Context or Usage specific survey

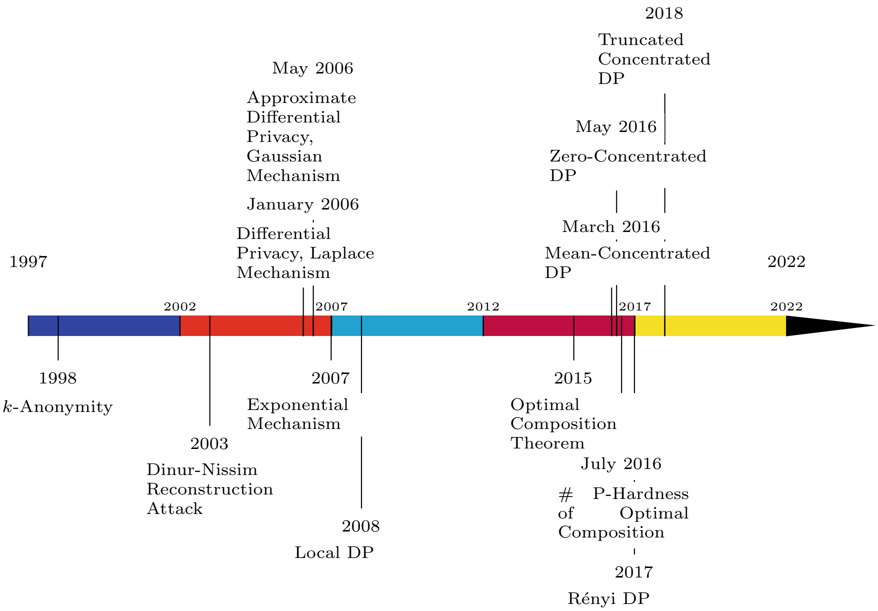

2018
Truncated Concentrated DP
May 2006
Approximate Differential Privacy, Gaussian Mechanism
January 2006
Differential Privacy, Laplace Mechanism
May 2016
Zero-Concentrated DP
March 2016
Mean-Concentrated DP
1997
2022
2002
2007
2012
2017
2022
1998
k-Anonymity
2003
Dinur-Nissim Reconstruction Attack
2007
Exponential Mechanism
2008
Local DP
2015
Optimal Composition Theorem
July 2016
# P-Hardness of Optimal Composition
2017
Rényi DP

Fig. 7.1 Timeline of important definitions and developments

- We discuss the theoretical foundations of differentially private machine learning and deep learning, and novel advances and algorithms in these fields, including in contexts like federated learning.
- We also take a look at some industrial/practical deployments of differential privacy to provide an idea of how some of the largest data-intensive companies/agencies use differentially private techniques to preserve privacy of their users' data.
- We perform bibliometric analysis of the papers that have been published in order to give an idea of the directions research in differential privacy is moving in.

In addition, readers are encouraged to refer to Fig. 7.1 for a concise timeline on some important developments in the study of differential privacy.

Definitions, Mechanisms, and Variants

To concretely discuss the subject matter of this paper, it is imperative to provide a quick and brief introduction to differential privacy and relevant definitions [30]. In particular, this section shall, in a brief technical manner, introduce differential privacy, its salient properties, and some basic tools used to implement. In addition, novel variants of differential privacy, the ideas behind them, and the properties associated with them shall be discussed.

Pure and Approximate Differential Privacy and Prerequisite Definitions

To start off, we shall define what differential privacy is concretely.

$$||x||_p = \left(\sum_{i=1}^{|\mathcal{X}|} |x_1|^p \right)^{\frac{1}{p}}.$$

Definition 1 ε- Differential Privacy
A randomised algorithm/mechanism $\mathcal{M}$ is said to be ε*-differentially private* or *purely differentially private* if $\forall\, S \subseteq \text{Range}(\mathcal{M})$ and $\forall\, x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $||x - y||_1 \leq 1$ (i.e. for *neighbouring databases* or $x \sim y$),

$$\ln \left(\frac{\Pr[\mathcal{M}(x) \in S]}{\Pr[\mathcal{M}(y) \in S]} \right) \leq \varepsilon$$

with the probability space being over the coin flips of the mechanism $\mathcal{M}$.

The above definition of differential privacy is also the earliest, and was given by Dwork, McSherry et al[28] in 2006. A relaxation of this was given by Dwork et al[27] shortly later in 2006 in order to provide comparable, albeit slightly weaker privacy guarantees, with addition of significantly less noise, and excusing events that have low probability (denoted by δ) of occurring.

Definition 2 (ε, δ)- Differential Privacy
A randomised algorithm $\mathcal{M}$ on the domain $\mathbb{N}^{|\mathcal{X}|}$ is said to be (ε, δ)-differentially private (or *approximately differentially private* if $\delta > 0$) if $\forall\, S \subseteq \text{Range}(\mathcal{M})$ and $\forall\, x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $||x - y||_1 \leq 1$ (i.e. for *neighbouring databases*),

$$\ln \left(\frac{\Pr[\mathcal{M}(x) \in S] - \delta}{\Pr[\mathcal{M}(y) \in S]} \right) \leq \varepsilon$$

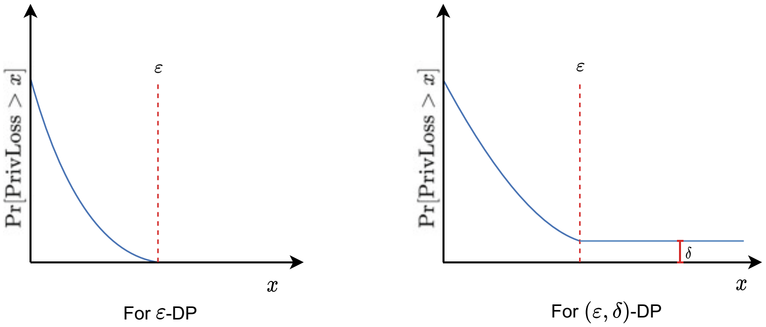
with the probability space being over the coin flips of the mechanism $\mathcal{M}$.

The privacy loss graphs for ε-DP and (ε, δ)-DP are given in Fig. 7.2.

Basic Properties of $(\boldsymbol{\varepsilon}, \boldsymbol{\delta})$-DP

Differential privacy possesses some very appealing properties that make modular design of differentially private algorithms and the analysis of their privacy properties possible, all of which are mentioned and proven in [30]. These properties are as follows.

Theorem 3 Post- Processing Invariance
Let $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \rightarrow R$ *be a randomised algorithm that is* $(\varepsilon, \delta)-$*differentially private.*



For ε-DP

For (ε, δ)-DP

Fig. 7.2 Privacy loss graphs for pure and approximate differential privacy

Let $f : R \rightarrow R'$ be an arbitrary randomised mapping. Then $f \circ \mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \rightarrow R'$ is $(\epsilon, \delta)-$differentially private.

In simpler terms, post-processing invariance simply means that an adversary cannot degrade the privacy of the output of a differentially private algorithm by post-processing it by any means without using the raw data itself [100].

For an algorithm satisfying ε-DP for databases that are at most k distance apart, *group privacy* provides a bound of $k\varepsilon$ on the privacy loss.

Theorem 4 GROUP PRIVACY
Let $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \rightarrow R$ (R being an appropriate codomain/range of the mechanism) be an $(\varepsilon, \delta)-$differentially private mechanism; for groups of size k, i.e. $\forall$ databases x, y such that $||x - y||_1 \leq k$ and all $T \subseteq R$,

$$\ln\left(\frac{\Pr[\mathcal{M}(x) \subseteq T] - \delta'}{\Pr[\mathcal{M}(y) \subseteq T]}\right) \leq k\varepsilon.$$

Where $\delta' = \frac{e^{k\cdot\varepsilon}-1}{e^{\varepsilon}-1} \cdot \delta$.

When $\delta = 0$, the mechanism satisfies ε-DP, and for groups of size k, the above reduces to

$$\ln\left(\frac{\Pr[\mathcal{M}(x) \subseteq T]}{\Pr[\mathcal{M}(y) \subseteq T]}\right) \leq k\varepsilon.$$

An often studied aspect of DP involves the composition of DP algorithms. This is based on the fact that DP allows graceful composition of various $(\varepsilon_i, \delta_i)$-DP algorithms to provide $(\sum_i \varepsilon_i, \sum_i \delta_i)$-DP.

Theorem 5 COMPOSITION OF DIFFERENTIALLY PRIVATE MECHANISMS
Let $\{\mathcal{M}_i\}_{i\in[k]}$, where $\mathcal{M}_i : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathcal{R}_i$, be a finite sequence of $(\varepsilon_i, \delta_i)-$differentially private algorithms. If

$$\mathcal{M}_{[k]} : \mathbb{N}^{|\mathcal{X}|} \rightarrow \prod_{i\in[k]} \mathcal{R}_i, \mathcal{M}_{[k]}(x) = (\mathcal{M}_1(x), \ldots, \mathcal{M}_k(x)),$$

then $\mathcal{M}_{[k]}$ *is* $\left(\sum_{i=1}^{k} \varepsilon_i, \sum_{i=1}^{k} \delta_i\right)$ *−differentially private.*

Tighter Composition Bounds

There have been attempts to further refine the composition bound. The advanced composition theorem, due to Dwork, Rothblum, and Vadhan [32], provides a tighter bound on composition under k-fold adaptive composition.

Definition 6 k- FOLD ADAPTIVE COMPOSITION
For $b \in \{0, 1\}$, family $\mathcal{F}$, and adversary A, for each $i \in [k]$, A produces two neighbouring databases x_i^0 and x_i^1, a mechanism $\mathcal{M}_i \in \mathcal{F}$, and parameters w_i, and is returned a randomly chosen $y \in \mathcal{M}_i(w_i, x_i^b)$.
The choice of b, once made is kept constant throughout the experiment (ergo giving us two different variants of the experiment).

Under k-fold adaptive composition, the advanced composition theorem is stated as follows.[1]

Theorem 7 ADVANCED COMPOSITION FOR DIFFERENTIALLY PRIVATE MECHANISMS
$\forall\, \varepsilon > 0, \delta, \delta' \in [0, 1]$, *and* $k \in \mathbb{N}$, *the class of* (ε, δ)*-differentially private mechanisms is* $(\varepsilon', k\delta + \delta')$ *differentially private under k-fold adaptive composition, where*

$$\varepsilon' = \sqrt{2k \ln(1/\delta')} \cdot \varepsilon + k\varepsilon(e^{\varepsilon} - 1).$$

In 2015, Kairouz, Oh and Viswanath [60] gave the optimal composition theorem, which provides a tighter composition bound than the advanced composition theorem itself under k-fold adaptive composition. The statement of the theorem in [60] is quite verbose. A simpler, but equivalent, restatement of the same was given by Murtagh and Vadhan [81], which is given below.

Theorem 8 OPTIMAL COMPOSITION THEOREM
For all $\varepsilon_i > 0$ *and* $\delta_i \in [0, 1)$, *where* $i \in [k]$, *and for any* $\delta' \in [0, 1)$, *the composition of the algorithms/mechanisms* $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$, *where* $\mathcal{M}_i$ *is* $(\varepsilon_i, \delta_i)$ *differentially private, yields* (ε', δ')*-differential privacy with the least value of* $\varepsilon' > 0$ *satisfying*

$$\frac{1}{\prod_{i \in [k]}(1 + e^{\varepsilon_i})} \sum_{S \subseteq [k]} \max \left\{ e^{\sum_{i \in S} \varepsilon_i} - e^{\varepsilon'} \cdot e^{\sum_{i \notin S} \varepsilon_i}, 0 \right\} + \frac{1 - \delta'}{\prod_{i \in [k]}(1 - \delta_i)} \leq 1.$$

Murtagh and Vadhan [81] showed that computing optimal composition is # P hard, even under simpler conditions like when only purely differentially private mechanisms are being composed. This makes it a computationally difficult problem.

[1] We shall forego a detailed discussion on k-fold adaptive composition and the advanced composition theorem for the sake of brevity. Details can be found in the cited paper and [30].

Also note that the composition of approximate differentially private algorithms seems to pay a penalty in factors of $\log(\frac{1}{\delta})$; each step of composition entails a $\sqrt{\log\left(\frac{1}{\delta}\right)}$ penalty on privacy loss.

A corollary of the above penalty meant that composition of (ε, δ)-differentially private algorithms is not associative, by which we mean that the composition of (ε, δ)-DP algorithms is not independent of the order and manner in which it is done.

These are certain issues that demanded attention, and were ultimately dealt with with the introduction and study of certain, more novel variants of differential privacy. Owing to size constraints, those are very briefly discussed here and in Table 7.2.

Concentrated differential privacy (CDP) was introduced and discussed by Dwork and Rothblum [31] (mCDP), and further refined by Bun and Steinke [12] (zero CDP or zCDP) in 2016 in order to address some of the aforementioned issues with existing definitions of DP and to gain sharper bounds on and associativity of composition. These are defined in terms of Rényi divergence [94] and the privacy loss random variable Z, and demand that Z is *concentrated* around a bound on the mean of the privacy loss μ and zero respectively. Interestingly, the optimal composition of zCDP can be computed in linear time, in contrast to that of (ε, δ)-DP, which is #P hard.

However, some tools like propose-test-release and amplification-via-subsampling are not supported by zCDP, which led Bun, Dwork et al. [10] to introduce a relaxation on zCDP, called truncated CDP in 2018. While CDP demands that the privacy loss is at least as concentrated as a Gaussian, tCDP relaxes that demand to having the privacy loss being concentrated like a Gaussian up until a certain amount of standard deviations (roughly ω) away.

Rényi differential privacy was introduced by Mironov [79] in 2017, which is also defined with respect to Rényi divergence. This seeks to improve on (ε, δ)-DP by modifying the relaxation condition from having a potential catastrophe occur with probability δ to weakening the DP guarantee in another fashion, and to provide better privacy loss accounting for the Gaussian mechanism. Moreover, Geumleuk, Song, and Chaudhuri [46] (2017) stated that many differentially private mechanisms that sample from distributions from exponential families, viz. posterior sampling, have closed-form Rényi DP guarantees available.

However, the aforementioned variants face issues regarding the composition of private algorithms and the analysis of techniques such as privacy amplification via subsampling [4]. To remedy that, Dong, Roth and Su [24] (2020) introduce a family of definitions of DP called f-differential privacy. This definition is motivated by differential privacy being formulated as a hypothesis testing problem for an adversary by [60, 117]. The authors also defined a specialisation of f-DP called Gaussian differential privacy or GDP.

Table 7.2 Table summarising different variants of (central) differential privacy

Variant	Notation	Canonical mechanism	Advantages or improvements	Disadvantages
Pure DP (Jan, 2006)	ε-DP	Laplace	Elegant composition, group privacy, post processing invariance	Too strict, requires significant noise addition at times
Approximate DP (May, 2006)	(ε, δ)-DP	Gaussian	Relaxes pure DP, requires less noise addition	Can lead to catastrophic failure with a small probability δ; Composition is not elegant or associative
Mean concentrated DP [31] (2016)	(μ, τ)-mCDP	Gaussian	Avoids catastrophic failure, tighter composition bounds, concentrates privacy loss around a bounded mean, associative and elegant composition	Not post-processing invariant
Zero concentrated DP [12] (2016)	(η, ρ)-zCDP	Gaussian	Preserves the benefits of mCDP, is post-processing invariant, avoids catastrophic failure, associative and elegant composition	Tools like propose-test-release and amplification via subsampling not supported
Truncated concentrated DP [10] (2018)	(ρ, ω)-tCDP	sinh-Normal	Relaxes zCDP, supports techniques like propose-test-release and amplification via subsampling	N/A
Rényi DP [78] (2017)	(α, ε)-RDP	Gaussian	No catastrophic failure, tighter error bounds, better privacy loss accounting for approximate DP, linearly additive composition	Tighter privacy bound on Gaussian is possible
Gaussian DP [24] (2020)	μ-GDP	Gaussian	Tightest possible privacy bound for the Gaussian mechanism, tight composition	N/A

Basic Mechanisms

Bringing differential privacy from the realm of theory to practice involved the introduction of various mechanisms used to achieve it in different ways and contexts. This subsection shall deal with some fundamental DP mechanisms. Most mechanisms endow DP guarantees on data releases by perturbing the data in some form or fashion, including by adding noise from an appropriate distribution. This perturbation can be done by perturbing the raw data prior to answering a query made on the dataset, or by perturbing the query response received on the raw data.

The earliest model of applying differential privacy involves having a trusted curator who holds all the data of the data subjects and is responsible for responding to queries made by analysts in a way that upholds differential privacy of the data. This is called *central differential privacy* (CDP). CDP is often enforced by the curator by the addition of noise to the data to perturb the values of true query responses by the use of privacy mechanisms.

There are some well known mechanisms used to add noise to query responses via different methods, the oldest of which is the global sensitivity method, which calibrates the noise added to responses to numeric queries with respect to a quantity called the *global sensitivity* of the query.

Definition 9 GLOBAL SENSITIVITY OF (A SET OF) QUERIES
The *global sensitivity of a query* f, or the ℓ_p sensitivity of f is given by

$$\Delta_p(f) = \max_{\|x-y\|_p \leq 1} |f(x) - f(y)|,$$

and that of a set of queries Q is

$$\varDelta_1(Q) = \sup_{\|x-y\|_p \leq 1} \left(\sum_{q \in Q} |q(x) - q(y)|^p \right)^{\frac{1}{p}}.$$

The Laplace mechanism from a paper by Dwork et al. [28] (which also provided the first known definition of what we now call pure differential privacy) calibrates the added noise with respect to the ℓ_1 sensitivity of the query being made, or the set of queries being made.

Definition 10 LAPLACE MECHANISM
The Laplace distribution, $\text{Lap}(\mu, b)$, is given by the pdf $p(z|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|z-\mu|}{b}\right)$. We denote $\text{Lap}(b) := \text{Lap}(0, b)$, pdf, $p(z|b) := p(z|\mu = 0, b) = \frac{1}{2b} \exp\left(\frac{-|z|}{b}\right)$. Given $\varepsilon > 0$, a set of queries, Q, and an input database x, the Laplace mechanism $\mathcal{M}_L$ returns noisy answers $\{q(x) + \text{Lap}(\varDelta_1(Q)/\varepsilon)\}_{q \in Q}$.

Dwork et al. [28] showed that the Laplace mechanism is ε-differentially private.

Later, the notion of approximate (i.e. (ε, δ)-) differential privacy was introduced in [27] along with the addition of Gaussian noise as a means of achieving it. The Gaussian mechanism, which is an (ε, δ)-differentially private mechanism, utilises the global sensitivity method, and adds Gaussian noise calibrated to the ℓ_2 sensitivity of queries made.

Definition 11 GAUSSIAN MECHANISM
Let $N(\mu, \sigma^2)$ denote the Gaussian distribution with mean μ and standard deviation

σ.
Given some $\varepsilon > 0$ and $\delta > 0$, a set of queries Q, and an input database the Gaussian mechanism $\mathcal{M}_G$ returns noisy answers $\{q(x) + N(0, \sigma^2)\}_{q \in Q}$, where $\sigma \geq c\frac{\Delta_2(Q)}{\varepsilon}$, and $c^2 > 2\ln\left(\frac{1.25}{\delta}\right)$.

It is common to take $\sigma^2 = 2\ln(\frac{1.25}{\delta})\frac{\Delta_2(Q)^2}{\varepsilon^2}$.

Additive noise addition is not suited for some contexts, and might even be counterproductive, like in some contexts involving choosing a best object from a database. For that reason McSherry and Talwar [76] introduced the exponential mechanism in 2007, which is ε-differentially private.

Definition 12 Exponential Mechanism
Given a database $x \in \mathbb{N}^{|\mathcal{X}|}$, a set of objects $\mathcal{H}$, a score function $s : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{H} \to \mathbb{R}$, and $\varepsilon > 0$, the exponential mechanism $\mathcal{M}_E$ outputs $h \in \mathcal{H}$ with probability $=$ $c\exp(\frac{\varepsilon s(x,h)}{2\Delta s})$, where $c \in \mathbb{R}_+$ is a suitable constant.

However, in certain contexts, having a trusted curator is not possible or is not desirable, and the data subjects could instead perturb their data locally and respond to queries. This brings us to the following definition given by Kasiviswanathan et al[63].

Definition 13 Local Differential Privacy (LDP)
A randomised mechanism $\mathcal{M}$ for $\varepsilon > 0$ is said to be ε-locally differentially private if for all pairs x, y of a user's private data, and for all possible outputs $z \in \text{Range}(\mathcal{M})$,

$$\ln\left(\frac{\Pr[M(x) = z]}{\Pr[M(y) = z]}\right) \leq \varepsilon$$

with the probability space being over the coin flips of $\mathcal{M}$.

The very first implementation of local differential privacy interestingly vastly predates the conception of differential privacy itself; in 1965, Warner [116] came up with the concept of *randomised response* for data collection about sensitive topics. A simple version, as mentioned in [30], of this involves the use of a fair coin; given a sensitive property $\mathcal{P}$, a respondent flips a coin. If they obtain a tails, then they answer truthfully, else they flip the coin again; in the latter event, if they obtain a heads, they answer yes, else they answer no. The coin tosses and the number of them remain private to the respondent. This is shown to give a close approximation of the expected number of people who possess the property $\mathcal{P}$, given by

$$\mathbb{E}[\text{Yes}] = \frac{3}{4}n(\text{has } \mathcal{P}) + \frac{1}{4}n(\text{does not have } \mathcal{P}).$$

In the context of differential privacy, it is defined for some $\varepsilon > 0$ as follows (and thus is ε-locally differentially private).

Definition 14 ε- Locally Differentially Private Randomised Response
Given $\varepsilon > 0$, for every private bit X in a piece of data, output

$$\mathcal{M}(X) = \begin{cases} X, \text{ with probability} = \frac{\exp(\varepsilon)}{1+\exp(\varepsilon)}; \\ 1 - X, \text{ with probability} = \frac{1}{1+\exp(\varepsilon)}. \end{cases}$$

Differentially Private Algorithms

While various definitions of differential privacy have been discussed along with some basic mechanisms, differential privacy can be implemented in ways that try to reduce or minimise error and privacy loss, while answering queries optimally, and sometimes only when they satisfy certain conditions. We discuss some prominent techniques, and some research done on and improvements made to them post their initial introduction.

Sparse Vector Technique (SVT)

The Sparse Vector Technique, or SVT, was introduced by Dwork et al. [29] in 2009 and later improved upon by Roth and Roughgarden [96] in 2009 and Hardt and Rothblum [53] in 2010. It serves the purpose of answering only a certain number (c) of queries from a sequence of k low sensitivity, and adaptively chosen queries with noise addition given that they lie above a certain threshold, T. Therefore the privacy loss would not increase as a factor of k (for pure DP) or $\sqrt{k}$ (as for approximate DP), but will depend on c, where $c \ll k$. In fact, the noise added by SVT scales as $\theta(\log k)$.

For an adaptively chosen sequence of $\frac{1}{n}$ sensitivity queries, the original SVT algorithm has $\delta = 0$, and noise from Lap(σ) is added to the threshold T to obtain a noised threshold $\hat{T}$. Laplace noise from Lap(2σ) is added to each query response and compared against the noised threshold, $\hat{T}$; only if the noised query response exceeds $\hat{T}$ does the algorithm output $\top$, else it outputs $\perp$, thus identifying “meaningful" queries the responses to which exceed the noised threshold. This satisfies ε-DP. Modified SVT follows the same paradigm as above but with $\sigma = \frac{\sqrt{32c \ln \frac{1}{\delta}}}{n\varepsilon}$, and this satisfies (ε, δ)-DP.

After obtaining the meaningful queries from SVT, one can get these queries answered using differentially private noise addition mechanisms. Another modification, called NumericSparse, augments Sparse to enable it to release query responses with Laplace noise addition for meaningful queries, and output $\perp$ for all others, with (ε, δ) differential privacy.

However, SVT as initially proposed remains practically infeasible. For instance in 2018, Papernot et al. [86] showed that SVT is often outperformed by the Gaussian mechanism answering all queries in model agnostic learning, owing to more concentrated noise addition and tighter composition via CDP or RDP based privacy accounting.

In 2020, Kaplan, Mansour, and Stemmer [62] devised an improvement to SVT by doing away with the SVT algorithm's reinitialisation after every meaningful query, and by deleting records from the dataset after they have contributed to some k many $\top$ answers. It is noted while SVT can answer c meaningful queries for a database of size= $\tilde{O}(\sqrt{c} \cdot \log m)$, Kaplan et al.'s technique can do the same for a database of size= $\tilde{O}(\sqrt{k^*} \cdot \log m)$, where k^* is the maximum number of times any record contributes to the response to a meaningful query, and is thus $\leq c$. This was also demonstrated to be useful for the shifting-heavy hitters problem[2].

Zhu and Wang [128] (2020) studied a generalised family of SVT that allows for the use of any noise-adding mechanisms, and introduced a variant of SVT that uses Gaussian noise instead of Laplace noise. They found tighter RDP bounds for SVT, improving upon previously known bounds by a constant factor.

Privacy Amplification via Subsampling

Privacy Amplification via Subsampling promises that applying (ε, δ)-differentially private mechanisms to random γ-subsets of records of a dataset yields a stronger privacy guarantee in the form of $(O(\gamma\varepsilon), \gamma\delta))$-DP [4]. The intuitive reason for this is that by picking a random subset/sample from the dataset, we reduce the probability of a data point that differs between that dataset and its neighbouring datasets from appearing in that sample.

The benefits of using this principle have been widely recognised, often providing significant improvements in terms of privacy loss wherever applied. A variant of the Gaussian mechanism called the Sampled Gaussian Mechanism (SGM) combines Privacy Amplification via Subsampling and the Gaussian mechanism, using which the privacy cost of a single evaluation diminishes quadratically with respect to the sampling rate.[3]

Definition 15 Sampled Gaussian Mechanism (SGM)
Let $f : P(S) \rightarrow \mathbb{R}^d$ be a function mapping subsets of a set S to d-dimensional real valued vectors. Then the Sampled Gaussian Mechanism is defined with respect to a sampling rate $q \in (0, 1]$ and $\sigma > 0$ as

$$SG_{q,\sigma}(S) := f(\{x : x \in S \text{ is sampled with probability } q\}) + \mathcal{N}(0, \sigma^2 \mathbb{I}_d),$$

Where each element of S is sampled independently at random with probability q without replacement, and $\mathcal{N}(0, \sigma^2 \mathbb{I}_d)$ is spherical d-dimensional Gaussian noise with per-coordinate variance σ^2.

[2] An element x is called a heavy hitter if it is current input for a large number of users.

[3] The following definition is from [79].

Mironov et al. [79] (2019) discuss SGM and provide a numerically stable method to calculate the Rényi DP of SGM precisely and provide nearly tight closed form bounds on the RDP of SGM.

Theorem 16 CLOSED FORM BOUND
If $q \leq \frac{1}{5}$*,* $\sigma \geq 4$*, and if along with α, these satisfy*

$$1 < \alpha \leq \frac{1}{2}\sigma^2 L - 2\ln\sigma;\, \alpha \leq \frac{\frac{1}{2}\sigma^2 L - 2\ln\sigma}{L + \ln(q\alpha) + \frac{1}{2\sigma^2}}$$

where $L := \ln\left(1 + \frac{1}{q(\alpha-1)}\right)$*, then SGM applied to a function of* ℓ_2*-sensitivity 1 satisfies* (α, ε)*-RDP where*

$$\varepsilon := \frac{2q^2\alpha}{\sigma^2}.$$

Privacy Amplification via Shuffling

Privacy Amplification via Subsampling [35] seeks to strengthen the privacy guarantees of a locally differentially private algorithm when viewed through the lens of central differential privacy; by privacy amplification via shuffling, a permutation invariant algorithm satisfying ε-LDP can be shown to satisfy $\left(O\left(\varepsilon\sqrt{\frac{\log(1/\delta)}{n}}\right), \delta\right)$ -DP.

Shuffling in itself is a powerful primitive. Techniques like BUDS by Sengupta et al. [101] primarily rely on shuffling techniques to provide differential privacy guarantees. BUDS in particular makes use of a technique the authors call iterative shuffling, which involves randomly choosing a shuffler from a set of shufflers in each iteration and shuffling a given lot of data using it until all of the data has been shuffled. A recent improvement to BUDS, titled BUDS+ [103], was introduced by the authors in 2022, which improves on the privacy utility tradeoff, longitudinal privacy guarantees, with memory efficiency and improved security guarantees.

Applications in Machine Learning

Differential privacy has found a rich potential for use in various realms of machine learning. Non-private machine learning carries certain risks, given that training an ML model uses a large volume of data, and non-private machine learning does not by default ignore specific facts provided in the training data, which can be used to compromise the privacy of the raw training data used to train a given model.

For example, in 2017, Shokri et al. [104] demonstrated a membership inference attack that, given a model, could infer whether an individual was included in the model's training dataset. Fredrikson et al. [42] (2014) and Fredrikson et al. [41] (2015) produced model inversion attacks that enabled an adversary to exploit confidence values of predictions to infer sensitive information about individuals in the training dataset. They demonstrated that these attacks were successful in compromising regression-based pharmacogenetic models, decision trees deployed in machine-learning-as-a-service systems, and neural networks for facial recognition. It has also been seen that convolutional neural networks can memorise arbitrary labelings of the training data (Zhang et al. [125]). More recently, Carlini et al. [13] (2021) produced an attack that could expose the training data of a model by passing possible training points into the model and seeing if there is a strong indication of membership, indicated by low log perplexity of a point/term.[4] That is, given a model, the log perplexities of possible training points are calculated and ranked in terms of how low each log perplexity value is. For example, in a not too large dataset, if an adversary wishes to find out if a particular sensitive value, like a social security number has been included, then the above ranking will indicate a lower log-perplexity for a sensitive value that is in the dataset over a random possible value of the sensitive feature. These attacks strongly demonstrate the necessity for private statistical learning (Table 7.3).

Differentially Private ERM

Differentially private machine learning, abbreviated as DPML, however began as a field of study as early as 2008 with the introduction of privacy-preserving empirical risk minimisation (ERM) achieved via *objective perturbation* by Chaudhuri and Monteleoni [15], which was further refined by Chaudhuri, Monteleoni, and Sarwate [16] along with the introduction of *output perturbation* for private regularised ERM as an application of a result from Dwork, McSherry et al. [28] (2006).

Table 7.3 Table summarising the different ways of performing differentially private stochastic gradient descent

Type	Introduced by	Year	Further Work
Output perturbation	[15]	2008	[16]
Objective perturbation	[15]	2008	[6, 16, 56, 66, 83]
Gradient perturbation	[119]	2010	[7, 105, 110, 115, 121]

[4] In this context, given a generative sequence model f_θ and a sequence $\{x_i\}_{i\in[k]}$, the log perplexity is given by

$$P_\theta(x_1, x_2, \ldots, x_k) = -\log_2 \Pr[x_1, \ldots, x_k | f_\theta] = \sum_{i\in[k]} \left(-\log_2 \Pr[x_i | f_\theta(x_1, \ldots, x_{i-1})]\right).$$

.

Consider the following setting: given a data space $\mathcal{X}$ and a label set $\mathcal{Y}$, training data $\mathcal{D} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} : i \in [n]\}$, and a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$, we wish to find a predictor $f : \mathcal{X} \to \mathcal{Y}$ that performs well. Regularised ERM attempts to find such a well performing predictor f by minimising the regularised empirical loss, given by

$$J(f, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) + \lambda N(f)$$

where $\lambda \in [0, \infty)$ is a parameter for regularisation and N is a real valued function that solely depends on f and is not dependent on any $(x_i, y_i) \in \mathcal{D}$ and is called the regularisation function that helps prevent overfitting.

In the above setting, Chaudhuri et al. [16] discuss a couple of ways of performing differentially private ERM for linear predictors f, $\mathcal{X} = \mathbb{R}^d$ by an abuse of notation, they denote f as a d-dimensional real vector and thus write $f(x) = f^T x$.

Output Perturbation

This involves adding noise calibrated to the sensitivity of the regularised ERM to the output of the same. More precisely, given training data $\mathcal{D}$, ERM outputs

$$f_{\text{priv}} = \arg\min_{f} J(f, \mathcal{D})$$

And then for a given $\varepsilon > 0$, a random noise vector b is taken with respect to the probability density function $v(b) = \frac{1}{\alpha} e^{-\beta \|b\|}$, where α is a normalisation parameter and $\beta = \frac{n\lambda\varepsilon}{2}$. Note that the pdf given here is that of the Gamma distribution resembles that of the Laplace distribution, and indeed, Chaudhuri et al. show that for a given λ, the L_2-sensitivity of regularised ERM is upper bounded by $\frac{2}{n\lambda}$, and that the following theorem holds.

Theorem 17 *If N is differentiable, 1-strongly convex, and ℓ is convex and differentiable, with $|\ell'(z)| < 1, \forall z$, then the aforementioned output perturbation method provides ε-differential privacy.*

Objective Perturbation

This method was initially introduced by Chaudhuri and Monteleoni [15] in 2008 in the context of logistic regression and reiterated in the general case of ERM in [16]. Objective perturbation as initially given in [15] involves minimising a perturbed objective function which is given by

$$J_{\text{priv}}(f, \mathcal{D}) = J(f, \mathcal{D}) + \frac{1}{n} \langle b, f \rangle \tag{7.4.1}$$

Where b is a random noise vector of dimension d sampled with respect to the probability density function v as given above but with $\beta = \frac{\varepsilon}{2}$, and $\langle b, f\rangle := b^T f$.

Chaudhuri et al. [16] presented a more sophisticated version of objective perturbation for ERM as given in algorithm 1, with noise being added with respect to $\varepsilon' := \varepsilon - \log\left(1 + \frac{2c}{n\lambda} + \frac{c^2}{n^2\lambda^2}\right)$ for a given privacy parameter $\varepsilon > 0$, and this choice of ε' is used by the authors to show that algorithm 1 is ε-differentially private as in the following theorem.

Algorithm 1 ERM with objective perturbation

```
Require: Training data $\mathcal{D} = \{z_i := (x_i, y_i)\}$, privacy parameter $\varepsilon$, regularisation parameter $\lambda$,
    parameter $c$
$\varepsilon' \leftarrow \varepsilon - \log\left(1 + \frac{2c}{n\lambda} + \frac{c^2}{n^2\lambda^2}\right)$.
if $\varepsilon' > 0$ then
    $\Delta \leftarrow 0$
else
    $\Delta \leftarrow \frac{c}{n(\exp(\frac{\varepsilon}{4})-1)} - \lambda$
    $\varepsilon' \leftarrow \frac{\varepsilon}{2}$
end if
Sample $b$ according to $v(b) = \frac{1}{\alpha}e^{-\frac{\varepsilon'}{2}\|b\|}$
Compute $f_{\text{priv}} = \arg\min J_{\text{priv}}(f, \mathcal{D}) + \frac{1}{2}\Delta\|f\|^2$.
```

Theorem 18 *If N is 1-strongly convex and doubly differentiable, and ℓ is convex and doubly differentiable, with $|\ell'(z)| \leq 1$ and $|\ell''(z)| \leq c, \forall\, z$, then algorithm 1 is ε-differentially private.*

Note that these results only hold given that the loss and regularisation functions satisfy certain mathematical conditions.

In 2012, Kifer, Smith, and Thakurta [66] provided a better analysis for and slightly modified output perturbation to allow for similar privacy guarantees as the original version. They did this while relaxing the requirement of differentiability for the regulariser and with the addition of less noise, and expanded its scope of application to problems involving hard constraints. Loosely speaking, this involves minimising the perturbed objective function Sect. 7.4.1 while taking random noise b from the Gamma distribution (similar to [16]) for ε-differential privacy and from the Gaussian distribution for (ε, δ)-differential privacy.

Kifer et al. also showed that where Chaudhuri et al.'s method provided an expected excess risk bound of $O\left(\frac{\zeta\|\hat{\theta}\|_2 p\log p}{\varepsilon\sqrt{n}}\right)$, where p is the dimension of the input data, their method provided a better bound of $O\left(\frac{\zeta\|\hat{\theta}\|_2\sqrt{p\log(1/\delta)}}{\varepsilon\sqrt{n}}\right)$. Later, Jain and Thakurta [57] (2014) provided techniques to perform output and objective perturbation by drawing noise only from the Gaussian distribution, and not from the Gamma distribution as in [16], that have expected excess risk bounds independent of the dimension

p. More precisely, Jain and Thakurta's technique achieved an excess risk bound of $O\left(\frac{(\log^2 n)(\zeta)^2\|\theta\|_2\sqrt{\log(1/\delta)+\varepsilon}}{\varepsilon\sqrt{n}}\right)$. Duchi et al. [25] (2013) provided a formal minimax risk based framework for local differential privacy on statistical estimators, and provided tight bounds on the expected excess risk in locally differentially private convex risk minimisation, and a gradient perturbed and locally differentially private version of stochastic gradient descent that achieves these bounds.

However these approaches impose the requirement that an exact optimum is arrived at for these guarantees to hold. This is often not possible in a practical setting due to various issues, such as those involving numerical precision in computers and the iterative nature of most optimisers in practice. This leaves these algorithms open to attacks in a practical setting. A prominent illustration is given in [77] by Ilya Mironov showing that practical implementations of something as basic as the Laplace mechanism is vulnerable to attacks due to irregularities of floating-point implementations of the mechanism, and this vulnerability is inevitably carried over to differentially private ERM.

More recent works on private stochastic convex optimisation by Iyengar et al. [56] (2019) and Bassily et al. [6] (2019) do not require convergence to an exact minimum. Works like these show that it suffices to obtain an approximate minimum for the objective function which makes these forms of objective perturbation more feasible in a practical setting for stochastic convex optimisation. However, these papers still impose the condition of convexity on the loss function. Neel et al. (2019) [83] does away with the requirement for convexity of the loss function, instead merely requiring it to be bounded while working with a discrete domain. For a continuous domain, the authors merely in addition that the loss function be Lipschitz in its continuous parameter.

Gradient Perturbation

Another popular way of performing differentially private machine learning is via *gradient perturbation* which involves performing gradient descent with noise addition. The idea of performing noisy gradient descent was initially proposed by Williams and McSherry [119] in 2010. A simple version of gradient descent was proposed by Song, Chaudhuri, and Sarwate [105] in 2013, which involved performing stochastic gradient descent (SGD), w.r.t. a convex loss function and an L_2-regularised objective (ergo strongly convex loss functions), with the following SGD update iteration,

$$w_{t+1} = w_t - \eta_t(\lambda w_t + \nabla\ell(w_t, x_t, y_t) + Z_t) \text{ for } Z_t \sim \mathcal{D}$$

where $\mathcal{D}$ is a distribution with the probability density function $\rho(z) = e^{\left(\frac{\alpha}{2}\right)\|z\|}$, and Z_t is some random noise drawn from $\mathcal{D}$. This guarantees ε-DP given that the norm of the gradient of the loss function, $\nabla\ell(w_t, x_t, y_t) \leq 1, \forall\, w$, and $\forall\, (x_t, y_t)$.

Bassily et al. [7] (2014) provided improvements to gradient perturbation, with the requirement that the loss function is Lipschitz bounded and that the domain of optimisation is bounded. Their algorithm adds Gaussian noise to the computed gradient, and thus via advanced composition guarantees achieves better and tighter risk bounds (vis-à-vis previous works) that are fairly comparable to theoretical bounds. They also provided algorithms for tasks like stochastic gradient descent, exponential sampling based convex optimisation etc. Their popular algorithm for private SGD is given by Algorithm 2.[5]

Algorithm 2 Differentially private stochastic gradient descent

```
Require: Training data $\mathcal{D} = \{d_1, \ldots, d_n\}$, privacy parameters $(\varepsilon, \delta)$, $L$-Lipschitz loss function $\ell$,
    convex set $\mathcal{C}$, and the learning rate function $\eta : [n^2] \to \mathbb{R}$.
Arbitrarily choose any $w_1$ from $\mathcal{C}$.
for $t = 1$ to $n^2 - 1$ do
    Pick $d \sim_u \mathcal{D}$ with replacement.
    $w_{t+1} = \prod_C (w_t - \eta(t)\,[n\nabla\ell(w_t; d) + b_t])$ where $b_t \sim \mathcal{N}(0, \mathbb{I}_p\sigma^2)$.
end for
Output $w^{\text{priv}} = w_{n^2}$.
```

The expected excess risk for algorithm 2 is shown to be $\tilde{O}\left(\frac{\|\mathcal{C}\|_2 L\sqrt{p\log(1/\delta)}}{\varepsilon}\right)$, and in addition, it satisfies (ε, δ)-differential privacy.

Abadi et al. [1] (2016) provided a gradient perturbation algorithm named DP-SGD for deep learning purposes. This is discussed in some more detail in Sect. 7.4.2.

In 2016, Papernot et al. provided an algorithm for gradient perturbed SGD that clips the gradients so as to bound their norms to allow noise addition via sensitivity based mechanisms, even for non-Lipschitz loss functions with unbounded variants.

Improvements, Practical Issues and Mitigation

Differentially private gradient descent comes with a set of auxiliary challenges that have been the subject of study to make it more practically feasible.

In 2021, Tran et al. [110] showed that differentially private ERM, in particular gradient perturbation involving gradient clipping, can incur a higher level of unfairness towards certain vulnerable/minority groups than non-private ERM, and they provide a mitigating algorithm for differentially private ERM that corrects for better fairness and higher utility after noise addition.

Xie et al. [121] (2021) note that intuitively the value of the gradient, and hence that of the gradient norm is inversely proportional to the number of iterations completed leading to varying privacy leakage risk across iterations, and that most approaches to implementing DP-SGD involve splitting the privacy budget evenly across iterations. Also, as the training process approaches convergence, the values of the gradients, being small, must be reported more accurately. They propose an adaptive, noise-

[5] $\sim_u$ denotes choosing uniformly at random, and $\mathbb{I}_p$ is the p-dimensional identity matrix.

reducing algorithm for DP-SGD that involves adaptively allocating a share of the privacy budget to each iteration.

Chen et al. [18] (2020) examine DP-SGD from a geometric perspective and note that gradient clipping can lead to a substantial bias in the update direction in each step of training, and may even lead to the update leading away from the optimum in some cases. They provide theoretical and empirical analyses in this regard and present a correction method to reduce the aforementioned bias by adding noise to the gradient prior to clipping.

Liu and Talwar [71] showed in 2018 that repeated hyperparameter selection for running ML models multiple times while finetuning hyperparameters can increase privacy loss significantly as opposed to a single run of an ML algorithm. In addition, they proposed a mitigating strategy that involves searching for hyperparameters randomly along with a random stopping rule. Papernot and Steinke [89] built upon this work in 2021, analysing this problem via the lens of Rényi DP and demonstrated this additional privacy loss issue successfully for SVMs trained on certain data distributions. They improved upon the mitigating strategy, providing improvements to the stopping rule that further reduced privacy loss significantly. Chaudhuri et al. [17] (2013) pointed out that validating a model and training it with different training parameters leads to an increase in privacy loss (training a model on the same set of ε-DP perturbed training data k times yields $k\varepsilon$-DP), and thus propose an approach for carrying out this validation exercise without splitting the privacy budget or the training set across rounds of training. They define stability conditions for the validation score function over changes in the training and validation sets with some privacy parameters ε, δ, and use that to produce sufficient conditions for differentially private guarantees on the validation procedure.

In practice, performing per-example gradient clipping a naïve implementation of DP-SGD incurs takes much more time for private gradient descent over its non-private counterparts when using commonly available deep learning frameworks like PyTorch which only provide the aggregated gradient for each batch using auto-differentiation. Lee and Kifer (2020) [67] remedy this by introducing new methods for per-example gradient clipping that is compatible with auto-differentiation in these frameworks, and thus provide a much faster practical implementation of DP-SGD. They achieve this by extending a trick given by Goodfellow [51] for calculating per-example gradients using auto-differentiation to various neural networks: given the preactivation of a layer, $z = Wx + b$, the auto-differentiator is asked to calculate $\frac{\partial L}{\partial z}$ and computing the per-example gradient as $\frac{\partial L}{\partial W} = \frac{\partial L}{\partial z} \otimes x$ for the example x.[6] Lee and Kifer extend this to neural networks and use it to compute the per-example gradient norms with the auto-differentiator and subsequently the clipping weights $v_i = \min(1, \frac{C}{\nabla_\theta \ell(f_\theta(x_i), y_i)})$ for a machine learning model f_θ with parameters θ and a clipping bound C. The auto-differentiator is then asked to provide the gradient of the reweighted objective function $L = \sum_{x_i \in B} v_i \nabla_\theta \ell(f_\theta(x_i), y_i)$ and then Gaussian noise can be added to the resulting value to privatise it.

[6] $\otimes$ here is the vector outer product.

Wang et al. [115] (2021) demonstrated that differentially private machine learning algorithms like DP-SGD can affect the prediction accuracy of the resulting privately trained model and can be highly unstable as different runs may yield models with significantly different prediction accuracies. This is due to the loss function having irregularities and several local minima, and perturbing the gradients can lead the algorithm down a different path than in a non-private setting. They thus propose smoothing the loss function so it has one, flat loss surface, and thus the training will be more robust and tolerant to noise addition.

Tràmer and Boneh (2021) [109] study how, ceteris paribus, varying the batch size and the learning rate jointly affect the learning of models like private image classifiers, and they propose a linear scaling rule that states that upon scaling the batch size and the learning rate by the same constant yields models with the same performance. They also note that on moderate privacy budgets, simpler linear models trained on handcrafted features outperform end-to-end deep learning algorithms on several tasks even if the latter may have more trainable parameters. To outperform these linear models trained on handcrafted features, these private deep learning models require either much more private data, or access to features learned on public data from a similar domain.

Ligett et al. [69] noted that most approaches to making empirical risk minimisation differentially private focus on fixing the privacy parameters (viz. ε) first and then attempt to maximise the accuracy of the learning subject to that. They introduce a noise reduction framework for differentially private ERM that with respect to a specified accuracy constraint searches the space of privacy levels that empirically satisfies the accuracy constraint. This search however can be computationally expensive as it requires running the learning algorithm for various privacy levels and whether they satisfy the accuracy constraint empirically.

Applying techniques like DP-SGD on a large-scale, as in large neural networks, continues to be a practical challenge. Da et al. [124] (2021) introduce a method to make it more feasible by reducing memory costs by modifying how relevant weight and gradient vectors/matrices are represented and presenting a correspondingly modified gradient perturbation algorithm.

Private Aggregation of Teacher Ensembles (PATE)

Taking inspiration from the sample-and-aggregate framework introduced by Nissim et al. in [84] and the concept of distillation of models as a strategy to protect against adversarial input perturbation/poisoning attacks [87], Papernot et al. [86] (2016) introduced a technique for private knowledge transfer from private teacher models to a student model by aggregating votes from the teacher models and adding random noise to the aggregation process.

This was further improved on by Papernot et al. [88] in 2018.[7] PATE (described in Fig. 7.3 is implemented by choosing a number of k many disjoint teacher models and

[7] Not to be confused with the 2016 paper coauthored by Papernot introducing PATE.

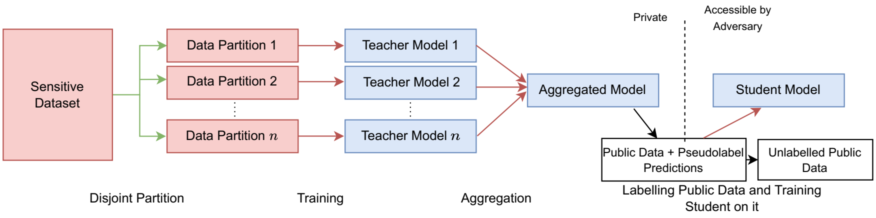


Fig. 7.3 Diagram illustrating the working of PATE [88]

partitioning a sensitive dataset into disjoint subsets, one each to train a private teacher model on, and then they are given public unlabelled data and they output (pseudo-)labels, and these are treated as *votes* that are aggregated into a vote histogram to which noise, viz. Laplace noise, is added and then the prediction with the highest number of votes in the noised histogram is released. This is used to train the student model. Ideally since these teacher models are disjoint, i.e. trained on disjoint subsets of the training data, and are assumed to have high accuracy, then a overwhelming majority of them shall vote for one (hopefully correct) label. In this case, the most voted for prediction shall have a large difference of votes from its nearest competitors and thus this can be released exactly, else an output with some randomisation is output by the ensemble. Then the student model is trained using these private labels and public data, using a fixed number of queries as the privacy cost increases with the number of queries, in a semi-supervised manner.

The case for PATE's privacy is further helped along by the fact that any given training point can belong to at most one teacher model's training dataset and given that the teacher models, even those without that point in their respective training datasets, almost always predict correctly, thus intuitively providing a measure of differential privacy. One noteworthy takeaway is that PATE is model-agnostic and can be implemented atop any chosen type of teacher models.

Applications in Deep Learning

Table 7.4 provides a brief summary of the works discussed in this section.

Foundations of Differentially Private Deep Learning

The study of applying differential privacy to deep learning models viz. neural networks is a natural extension of the work done in differentially private machine learning. This was initiated by Abadi, Chu et al. in 2016 [1], which adapted existing gradient perturbation techniques from works like [7] to create an algorithm called DP-SGD for differentially private training of neural networks. Abadi et al. propose

Table 7.4 Summary of some of the applications of differential privacy to various (deep) learning tasks

Category	Work	Remarks
Gradient perturbation in deep learning	[1]	Introduces DP-SGD
	[9]	Extends [1] with the use of Gaussian DP
DP for computer vision	[55]	Training CNNs with DP (DPAGD-CNN)
	[129]	Private-kNN
	[72]	Transfer learning for computer vision using sparse subnetworks and DP
	[49]	Pre-training computer vision models on public data with private finetuning (AdaMix)
DP for Graph Learning Tasks	[98]	Node-level privacy for GNNs using randomised response/LDP
	[85]	Protection of private GNNs by training public student models
	[97]	Attempt to convert LDP based node privacy approaches into a CDP setting
	[80]	Adapt DP-SGD to train GNNs
	[11]	Efficient correlation clustering on graphs using DP
DP for natural language processing	[39]	Language Modelling with DP using metric-based LDP
	[38]	Language Modelling with DP using metric-based LDP
	[74]	Private LSTM using private federated averaging
	[68]	Efficient finetuning of large transformer models using DP-SGD
	[65]	Privately finetuning an initial public language model for better accuracy

sampling a subset $\tilde{\mathcal{D}}$ of a fixed size L of a dataset $\mathcal{D}$ of size n uniformly at random and then for every $z \in \tilde{\mathcal{D}}$, the gradient of the loss function is calculated and clipped to have an ℓ_2 upper bounded by some parameter U, following which they are averaged and privatised by the addition of Gaussian noise calibrated to the sensitivity bound U, and this is performed iteratively until the termination of the training process. Note that due to clipping, the loss function is not restricted to being Lipschitz. This procedure endows (ε, δ)-differential privacy.

Inspired by the introduction of f-DP, particularly Gaussian DP, Bu et al. [9] (2020) extend the work done by Abadi et al. [1] by using Gaussian DP. The authors provide an improved analysis of differentially private deep learning, and noisy versions of stochastic gradient descent and Adam optimisation using Gaussian DP and its benefits over previously defined variants, such as improved handling of composition and subsampling, without requiring the development of sophisticated analysis tools (viz. the moments accountant) as in [1]. It is shown that similar privacy guarantees can be achieved by the use of f-DP/Gaussian DP as compared to using (ε, δ)-DP or the moments accountant, thus yielding models with higher utility.

Applications of Differentially Private Learning to Particular Tasks

For Computer Vision

In this subsection, we describe a few applications of differential privacy to deep learning tasks.

There has been some work on applying differential privacy to deep learning models for computer vision (viz. convolutional neural networks (CNNs)). Huang et al. [55] (2019) present an algorithm which they call DPAGD-CNN (Differentially Private Adaptive Gradient Descent for CNNs) which trains CNNs by varying the amount of privacy budget available for adding noise to the gradient and optimal step size adaptively and accounting for privacy using zCDP.

For tasks with limited data availability like those related to computer vision, getting ample labelled data is often expensive and splitting the datasets to train disjoint teacher models as in PATE will yield suboptimal accuracy. Zhu et al. [129] (2020) proposed a method called *Private-kNN* which avoids splitting the training private dataset, given a student model, a feature extractor, and public unlabelled data. It involves picking a random subset from the private dataset with Poisson sampling, and then running the kNN algorithm on it with the aid of the feature extractor, and this process is performed iteratively, with the feature extractor being updated by the student (deep) model with every iteration. Query responses are released given that they pass noisy screening by having a large degree of consensus in voting. Subsequently, the student model is trained using the released query responses as pseudo-labels in a self-supervised manner. The authors demonstrate, with Rényi DP privacy accounting and by the principle of privacy-amplification-via-subsampling, that their method provides significant improvements on existing methods' privacy bounds despite its iterative nature, rendering it a practical private deep-learning method for computer vision.

Luo et al. [72] (2021) note that the assumption on the availability of ample public data made by most DP transfer learning models can be unrealistic, especially for computer vision and visual recognition tasks, and that traditional models of performing computer vision tasks with differential privacy (viz. DP SGD) work only on simple datasets and shallow networks. They contend that in order to improve the privacy-utility tradeoff in this context, the number of training parameters must be minimised. To that end, they provide novel methods of performing transfer learning that produce an optimal, sparse subnetwork.

Golatkar et al. [49] (2022) note that pre-training models on public data might be beneficial for language models, but for computer vision tasks, they can lead to a heavy privacy-utility tradeoff. To this end, they introduce AdaMix which involves pre-training a model with few-shot or cross model zero-shot learning on public data prior to private finetuning of the model using noised, projected, private gradients (w.r.t. an adaptively changing clipping threshold that is large at first and reduces in size to ensure higher accuracy at first and better privacy towards the end of the training) using a private dataset, which vastly improves on the privacy-utility tradeoff of its baselines.

For Graph Neural Networks and Learning on Graphs

There has been a brief body of work regarding differential privacy for graph neural networks (GNNs), which are neural networks that work on graph based data that comprises of several nodes that contain some node data and are joined by edges according to how they are related in a given context. Several of those, prominently Sajadmanesh et al. [98] (2021), propose using locally differentially private techniques like randomised response to perturb (randomly chosen) bits of the node data at the node level prior to training to ensure privacy of the nodes' sensitive data. Others like Olatunji et al. (2021) [85] focus on protecting proprietary/sensitive GNN models by using the secret GNN model as a teacher to teach a public "student" model without revealing the private model's weights. Olatunji et al. achieve this with the use of central differentially private mechanisms along with privacy-amplification-via-subsampling by randomly selecting an induced subgraph of the teacher model's graph and using it to train the student model.

Sisong et al. [97] (2021) aim to convert the LDP based node privacy problem into a centrally differentially private one by using (seemingly) trusted secure hardware like Intel's SGX to do differentially private calculations prior to releasing node data to the analyst for training the GNN, thus leading to significant improvements in terms of utility and accuracy of training. However Intel's SGX has been shown to be vulnerable to various attacks [37] and has been deprecated, rendering this approach ineffective in a practical setting for the moment.

More recently, Mueller et al. [80] (2022) presented a method to perform graph level classification on multi-graph datasets by adapting the DP-SGD algorithm to use it to train graph neural networks.

Eliáš et al. [34] (2020) proposed a (ε, δ)-differentially private algorithm to produce, given a graph G (with potentially sensitive information) with n vertices and m edges, a synthetic graph G' that approximates all the cuts of G up to an additive error of $O\left(\sqrt{\frac{mn}{\varepsilon}}\log^2(\frac{n}{\delta})\right)$, and $o(m)$, $\forall\, m > n \log^C n$, providing good approximations for sparse graphs as well. Using ideas from this and differentially private noise addition, Bun, Eliáš, and Kulkarni [11] (2021) proposed efficient differentially private algorithms for performing privacy-preserving correlation clustering for weighted and unweighted graphs with subquadratic error.[8]

[8] A task in unsupervised machine learning that involves clustering a set of objects based on the given information about how similar/dissimilar an object is to another. Introduced by Bansal, Blum, and Chawla [5] in 2002.

For Natural Language Processing (NLP)

There has also been some work on differentially private NLP/language modelling. Using DP-SGD from [1] naïvely can lead to punishing privacy-accuracy tradeoffs for language modelling. Feyisetan et al. [39] (2020) and Fernandes et al. [38] (2019) instead employ differential privacy for language modelling, for text perturbation to ensure geo-indistinguishability in location data and for author obfuscation respectively, with respect to metric-based relaxations of local DP, and add noise to the vector embedding of a word.

McMahan et al. [74] (2018) provide a private LSTM model that utilises a noised version of the federated averaging algorithm [73][9] to give a trained model with strong privacy guarantees and a high accuracy relative to its non-private counterpart for a large enough dataset.

Kerrigan et al. [65] (2020) improve on the state-of-the-art differentially private language models in terms of the privacy-utility tradeoff by first training a public/non-private model on a large public dataset before private finetuning by using DP-SGD using private, out-of-distribution dataset.

Li et al. [68] (2021) describe methods to efficiently fine tune large transformer models with millions of parameters directly with ε-differential privacy, using DP-SGD, for $\varepsilon \in \{3, 8\}$. Prominently, they introduce a relatively computationally efficient and memory efficient technique known as *ghost-clipping* to improve upon the utility of DP-SGD, which when naïvely implemented incurs substantial memory overhead while clipping per-example gradients. Ghost clipping involves arriving at the per-example gradient norms without substantiating the per-example gradients themselves, and achieves a significantly lower memory complexity as a result. This is inspired by the technique to efficiently calculate gradient norms introduced by Goodfellow in 2015 [51].

In Federated Learning

When it comes to privacy-preserving machine learning, federated learning is often discussed along with DPML. Consider a scenario with multiple users that hold personal data that is required for training/updating a model by a central authority/server, but the users may not want to part with their personal, and potentially sensitive, data in a raw form. Apart from privacy concerns, sending huge volumes of one's personal data, viz. photographs or voice recordings, will be expensive in terms of communication costs. This is a privacy preserving variant of an older technique known as *distributed (machine) learning* which involves outsourcing certain training tasks for a model to multiple nodes which are all under the control of the same central authority.

Introduced by McMahan et al. [73] in 2016, federated learning, or FL for short, provides a solution for this by having the central server provide the devices with

[9] Federated averaging shall be discussed in the next subsection.

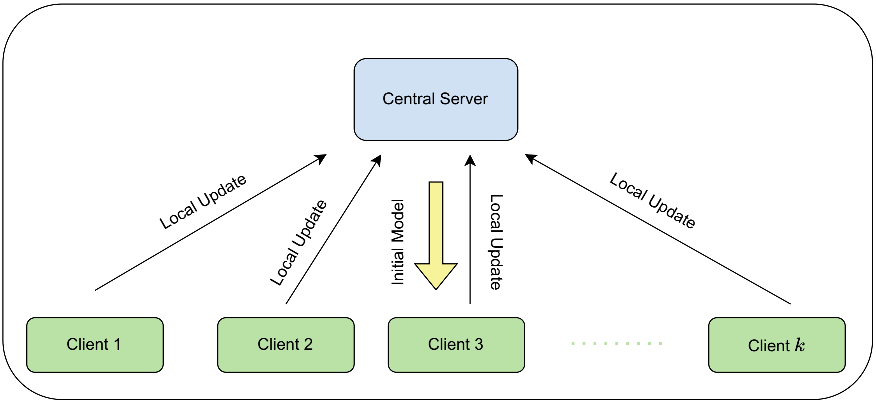


Fig. 7.4 An overview of the federated learning process. The central server provides the clients with a global, initial model. The clients train the model on their local data and send the result of their local training (local updates) back to the server. The server then aggregates the local updates and updates the global model

an initial model and have them train said model on their personal data, and send the resulting weight updates or gradients to the central server to be aggregated as a weighted arithmetic mean, via an algorithm known as `FederatedAveraging`. This process has been shown to yield pretty accurate models as a result. Figure 7.4 provides an overview of the process of federated learning.

Federated learning is a powerful privacy-preserving and communication cost-cutting technique, but it has a variety of facets that have been worked on since its conception; we shall restrict our focus to privacy-related concerns and application of DP to FL in particular. Attacks on the models resulting from FL, including membership inference attacks as discussed earlier, cannot be ruled out, and one significant line of work that seeks to improve on the privacy of FL is by applying differential privacy to it.

Works like those by Wei et al. (2020) [118] use a commonly used method of clipping the weight updates with respect to a specified clipping bound C and adding differentially private random noise to them (viz. by applying the Gaussian mechanism) to achieve DP guarantees. Paul et al. [92] introduced FLaPS, a paradigm to conduct federated learning with scalability and enhanced privacy guarantees with differential privacy guarantees being endowed by BUDS [101] and ARA [91]; FLaPS involves taking devices participating in the training process and clustering them into silos and assigning a cluster centre among the silo members, and aggregating their privatised data securely using ARA and training the initial model provided by the central server using the aggregated data and privatising the model's weights with BUDS. Following this, the cluster centres send these reports to the central server for aggregation using ARA and `FederatedAveraging`, in that order. These methods are shown to provide DP guarantees, and the latter additionally provides scalability and communication efficiency by reducing the number of links between individual devices and the central server.

Truex et al. [112] (2020) present LDP-Fed which performs federated learning with local differentially privatised client updates that are accepted or rejected uniformly at random (thus achieving privacy loss reduction by privacy-amplification-via-subsampling) before the central server aggregates the accepted updates.

Hu et al. [54] (2020) have the members of a set of participating clients Ω^t train their local models with gradient perturbation with Gaussian noise, and then perturb their respective local weight updates p_i^t by adding some random value r_i^t to it such that $\sum_{i \in \Omega^t} r_i^t = 0$, generated using a protocol involving a certain pseudorandom function and a seed agreed upon by participating and mutually communicating clients during each round. This allows the central server, which may be honest-but-curious, to aggregate the local weight updates securely without having an idea as to what each of these local weight updates actually look like.

Girgis et al. [47] (2021) presented a method to learn a model with communication constraints and provide privacy guarantees with reasonable utility of the model. The server chooses a random subset of clients at each round, each of which use a random subset of their personal training data, and privatise their responses by clipping their gradients and using an LDP mechanism to privatise their gradients. Following this, the aggregating central server receives a random permutation of these updates after shuffling via a secure shuffler. This essentially is a subsampled shuffle model. The same authors [48] (2021) extend their work on CLDP-SGD to present an analogous differentially private approach to distributed learning, and provide a stronger privacy analysis of CLDP-SGD using RDP.

Andrew et al. [3] (2021) note that there is no a priori optimal value for the update clipping bound for noise addition across various learning tasks, and the update norm distribution is dependent on the model, client learning rate, amount of data possessed by each client and other such parameter. The authors then propose adaptively choosing a clipping bound at a particular quantile of the update norm distribution at any point in time, instead of adhering to a fixed clipping bound specified beforehand, thus producing a method that is shown to outperform any prior choice of a fixed bound.

Truex et al. [111] propose a hybrid approach to federated learning which combines differential privacy with secure multiparty computation (SMC) with a tunable trust parameter to provide better model accuracy along with provable privacy guarantees and protecting against extraction attacks and collusion threats. The clients are queried by the server/aggregated and respond with adding differentially private noise to their query responses and encrypting them homomorphically, which the aggregator can then aggregate. The aggregator then asks a sufficiently large subset of clients (determined by the trust parameter) to help decrypt the aggregate value. The combined usage of differential privacy and SMC makes sure that the model output is differentially private and that the exchange of non-private messages remains protected without information leaks. The authors also provide algorithms to implement this approach on various ML and DL models viz. CNNs, SVMs, decision trees, etc.

Papernot et al. [90] (2021) note that the implementation of differentially private techniques for learning are often used on models that are shown to be successful in a non-private setting, leading to sub-optimal private training (sub-optimal utility for a given privacy level) in some cases. They therefore suggest selecting model architectures explicitly for private training in the first place. They also note that the choice of activation function (and the bounds on them or lack thereof) plays a major role in determining the sensitivity of private deep learning, and that bounded activation functions like the tempered sigmoid function consistently outperform unbounded activation functions like ReLU.

Xu et al. [123] (2021) discuss the application of differential privacy to asynchronous decentralised distributed learning and introduce A(DP)^{2}SGD, which is a differentially private version of asynchronous decentralized parallel stochastic gradient descent (ADPSGD), which helps protect from information leaks during communication between nodes. This essentially entails the addition of Gaussian noise to the local gradient by each client while carrying out the ADPSGD process.

Industrial Deployments

The utility of differential privacy has been widely recognised by industry and data handling organisations, but practically implementing it in a manner that is easy to use, even by users without an in-depth understanding of differential privacy has proven to be a challenge and an important task for its widescale adoption. We shall very briefly discuss some practical implementations (mentioned in Table 7.5) of DP below.

Table 7.5 Table summarising the different practical deployments and industrial implementations of DP discussed in this section

Organisation	Work/industrial implementation	Year
Google	RAPPOR	2014
	Privacy on Beam	2021
	DP SQL	2020
	Plume	2022
Apple	Sequence Fragment Puzzle	2017
Microsoft	PINQ	2010
	One-Bit Estimation	2017
	LinkedIn Audience Engagement API	2020
Uber	FLEX	2018

By Google

RAPPOR

RAPPOR [36] (2014), as discussed, is the first well-known industrial implementation of differential privacy, which was deployed by Google for the Chromium browser. It consists of a few layers: a bloom filter and then two rounds of randomised response (for logitudinal privacy and then to deidentify the user from the bloom filter output with one application of randomised response). A string from a known universe of strings is passed through these layers by a user, following which the resulting reports are sent from each user to an aggregator who infers useful information from the resulting aggregation.

Privacy on Beam and Differentially Private SQL

Google introduced an end-to-end differential privacy solution for Apache Beam called Privacy on Beam [52] that can be used without any particular expertise with differential privacy. Wilson et al. [120] [52] introduced a system to answer various SQL queries with user-level differential privacy, and empirically demonstrate the utility, robustness, and scalabity of this system.

Plume (2022)

Amin et al. [2] adapt and modify the MapReduce model [21] of distributed computation to introduce Plume for Google, which provides scalable differential privacy for large databases. The privacy budget is controlled by limiting how many keys any user can contribute records to, then out of these keys a safe key set S is produced, and instead of using a non-private aggregation algorithm as in MapReduce, a differentially private mechanism like the Laplace mechanism is used to add noise to the values corresponding to the keys in S.

By Apple

Apple [108] took inspiration from the Count Sketch algorithm which was developed by Charikar et al. [14] to efficiently estimate the most frequent items in a data stream using limited storage space.

Apple's privacy system in the paper utilised an LDP randomisation technique known as Sequence Fragment Puzzle for privatisation at the user-level. Each word is broken up into fragments and the frequency of each word is calculated. The user then concatenates a random substring of the string (word) with the hash of the entire string and privatises it, and transmits it with the index at which the substring starts in the string. The transmission of these messages is further endowed with privacy and security guarantees by delaying the transmission of these messages, then randomly subsampling the messages that are received and removing identifying details like the user's IP address from the messages and using TLS encryption to send it to the server.

This work notably improves on RAPPOR in that while RAPPOR only supports the privatisation of the members of a fixed universe of strings, Sequence Fragment Puzzle allows for the discovery of new strings. However, it has faced criticism about some of its facets from works like [107].

By Microsoft

PINQ

McSherry [75] (2010) introduced Privacy Integrated Queries (PINQ), an API resembling and extending Language Integrated Queries[10] (LINQ), which can be used to perform privacy-preserving data analysis on sensitive datasets. Proserpio et al. [93] designed an extension to PINQ known as Weighted PINQ or wPINQ that assigns weights to every row in the database and then scales the weights of a row in a join to ensure that the overall sensitivity is 1. It supports general equijoins.

One-Bit Estimation

Ding et al. [22] (2017) utilise randomised response to generate local reports by users starting from a raw local value $X_i \in [0, m]$ as follows,

$$Y_i = \begin{cases} 1 \text{ with probability } = \frac{1}{e^\varepsilon+1} + \frac{X_i}{m} \cdot \frac{e^\varepsilon-1}{e^\varepsilon+1} \\ 0 \text{ otherwise} \end{cases} .$$

These are then aggregated to gain an unbiased average report from the local reports as follows,

$$\hat{\mu} = \frac{m}{n} \sum_{i=1}^{n} \frac{Y_i \cdot (e^\varepsilon + 1) - 1}{e^\varepsilon - 1}.$$

Owing to the use of randomised response, this value in $[0, m]$ is converted into a single bit long report. The authors also provide a method to perform memoisation using one-bit estimation to protect rapidly updated data from longitudinal attacks.

[10] Which is an SQL-like declarative query language extension for .NET languages.

LinkedIn Audience Engagement API

Rogers et al. [95] (2020) introduced a system to provide user-level privacy guarantees via differential privacy while being able to provide audience engagement insights to enable marketing analytics and related applications. In particular, the authors describe a number of DP algorithms (for cases where the data domain is reasonably sized and known, and where the data domain is unknown or very large in size) that help the LinkedIn Audience Engagement API to carry out privacy-preserving data analysis. They also introduce a privacy budget management system that tracks an analyst's privacy budget even over multiple data centres.

By Uber

Johnson et al. [59] (2018), introduced key innovations to enable the practical use of differential privacy. Some of their most prominent contributions include the introduction of *elastic sensitivity* which is a novel and convenient method to approximate and upper bound the local sensitivity, and can be used to obtain parameters to employ any local sensitivity based DP mechanism. Building on top of that, they propose FLEX, an end-to-end differential privacy solution for real-world SQL queries that uses elastic sensitivity.

Bibliometric Analysis

With a profusion of research in differential privacy and its various application being published in recent years, we shall provide some very brief bibliometric insights into the same to inform about trends in research and future directions.

For this, we shall be using the arXiv dataset, given that most significant works on differential privacy are available on arXiv. Starting from 2006, a total of 1653 papers have been published on arXiv with the term 'differential privacy' and case variations thereof in their abstracts, with a total of 1020 authors having published at least 2 papers on DP, and a total of 479 authors having published at least 3 papers.

The left subtable of Table 7.6 provides some simple statistics on authorship of these papers.

Now focusing on the literature published since 2012, it is observed that a total of 1609 papers were published with differential privacy being mentioned in their abstracts on arXiv, with a total of 998 authors having contributed to at least 2 papers, and with a total of 469 authors having contributed to at least 3 papers since 2012. This indicates that the bulk of research on differential privacy has occurred in the last decade.

Table 7.6 Authorship statistics by number of papers published on Differential Privacy (on the left) and that on Differential Privacy *and* Machine Learning (on the right) (since 2005)

Statistic/percentile	Value	Statistic/percentile	Value
Mean	1.834890	Mean	1.316706
Standard deviation	2.358568	Standard deviation	0.788847
Minimum	1	Minimum	1
25%	1	25%	1
50%	1	50%	1
75%	2	75%	1
Maximum	42	Maximum	9

Out of those 1609 works published since 2012, 418 papers feature the terms "machine learning", "gradient descent", "empirical risk", and "deep learning" (and case variations thereof) in their abstracts.

The right subtable of Table 7.6 provides statistics on authorship of papers per author on differential privacy mentioning topics related to machine learning in their abstracts. Note that while the papers in this subset of the data mention machine learning, deep learning, and/or ERM, they might not deal with DPML, so the number of papers that actually deal with the applications of differential privacy to machine learning might be lesser than the number reported.

Figure 7.5 depicts the number of papers mentioning DPML and those mentioning differential privacy as a whole since the years since 2011. It can be seen that the number of publications on differential privacy and machine learning with differential privacy have seen a consistent and significant increase in recent years, and that DPML has grown to account for a significant proportion of DP publications in the last 3–4 years.

In addition, works on differentially private machine learning have become a staple of top AI conferences in recent years. Some data on this is available on https://differentialprivacy.org and the respective conference websites. For instance, NeurIPS 2020 featured 31 works on differential privacy, and NeurIPS 2021 featured 48 works on the same. ICML 2021 and ICML 2020 featured 21 and 22 works on differential privacy respectively. COLT 2020 featured 9 papers dealing with differential privacy.

Remarks and Conclusion

This survey seeks to be a reflection of the massive strides made in recent years in the field of differential privacy, and the various applications of the same. It brings the focus back to differential privacy and technicalities of the same; in particular, some prominent variants of differential privacy and differentially privacy techniques,

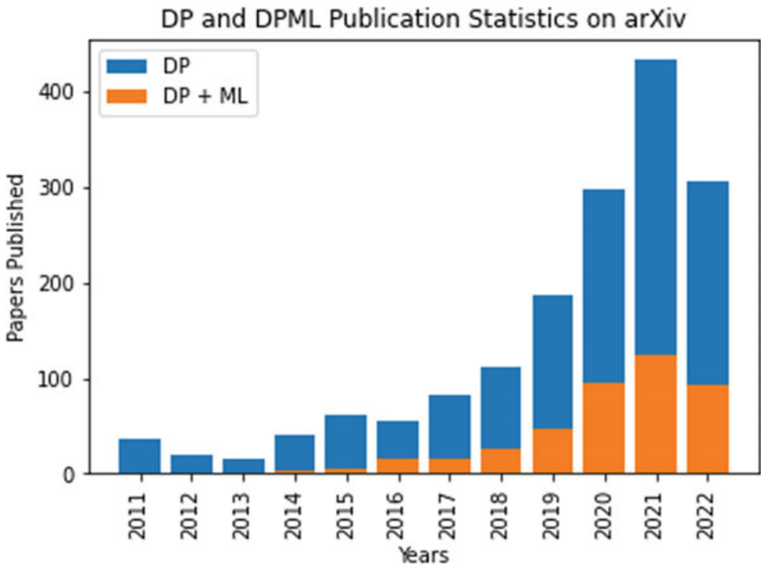


Fig. 7.5 Bar Graph showing the number of DP and DPML publications in each year; the 2022 statistics are as of June, 2022

accounting techniques and algorithms, and novel developments in terms of these were discussed. In addition, its real world applications from a differential privacy-first lens to fields like machine learning, deep learning, federated/distributed learning and DP-ERM were explored. We also discuss a few implementations of differential privacy in industry and for important tasks like census data privatisation.

This merely discusses a prominent subset of the profusion of research that has been done in differential privacy and its applications, and many of the techniques mentioned here have their own challenges in terms of feasibility of practical implementation, the privacy-utility tradeoff, the amount of data required to get high utility with high privacy, and improving on these remains the subject of much study. The goal of this survey is to augment existing survey literature on different facets/applications of differential privacy, and to show how differential privacy has become, and rightly so, the de-facto standard of privacy with wide ranging applications and implications.

References

1. Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L (2016) Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. https://doi.org/10.1145/2976749.2978318
2. Amin K, Gillenwater J, Joseph M, Kulesza A, Vassilvitskii S (2022) Plume: differential privacy at scale

3. Andrew G, Thakkar O, McMahan HB, Ramaswamy S (2021) Differentially private learning with adaptive clipping
4. Balle B, Barthe G, Gaboardi M (2018) Privacy amplification by subsampling: tight analyses via couplings and divergences
5. Bansal N, Blum A, Chawla S (2002) Correlation clustering. In: The 43rd annual IEEE symposium on foundations of computer science. Proceedings, pp 238–247. https://doi.org/10.1109/SFCS.2002.1181947
6. Bassily R, Feldman V, Talwar K, Thakurta A (2019) Private stochastic convex optimization with optimal rates. CoRR http://arxiv.org/abs/1908.09970
7. Bassily R, Smith A, Thakurta A (2014) Differentially private empirical risk minimization: efficient algorithms and tight error bounds
8. Boulemtafes A, Derhab A, Challal Y (2020) A review of privacy-preserving techniques for deep learning. Neurocomputing 384:21–45
9. Bu Z, Dong J, Long Q, Su WJ (2020) Deep learning with gaussian differential privacy. Harvard Data Sci Rev 2020:23
10. Bun M, Dwork C, Rothblum GN, Steinke T (2018) Composable and versatile privacy via truncated cdp. In: Proceedings of the 50th annual ACM SIGACT symposium on theory of computing. STOC 2018, association for computing machinery, New York, NY, USA, pp 74–86. https://doi.org/10.1145/3188745.3188946
11. Bun M, Eliáš M, Kulkarni J (2021) Differentially private correlation clustering
12. Bun M, Steinke T (2016) Concentrated differential privacy: simplifications, extensions, and lower bounds
13. Carlini N, Tramer F, Wallace E, Jagielski M, Herbert-Voss A, Lee K, Roberts A, Brown T, Song D, Erlingsson U, Oprea A, Raffel C (2021) Extracting training data from large language models
14. Charikar M, Chen K, Farach-Colton M (2002) Finding frequent items in data streams. In: Proceedings of the 29th international colloquium on automata, languages and programming. ICALP '02, Springer-Verlag, Berlin, Heidelberg, pp 693–703
15. Chaudhuri K, Monteleoni C (2008) Privacy-preserving logistic regression. In: Koller D, Schuurmans D, Bengio Y, Bottou L (eds) Advances in neural information processing systems, vol 21. Curran Associates, Inc. (2008). https://proceedings.neurips.cc/paper/2008/file/8065d07da4a77621450aa84fee5656d9-Paper.pdf
16. Chaudhuri K, Monteleoni C, Sarwate AD (2011) Differentially private empirical risk minimization. J Mach Learn Res 12(29):1069–1109. http://jmlr.org/papers/v12/chaudhuri11a.html
17. Chaudhuri K, Vinterbo SA (2013) A stability-based validation procedure for differentially private machine learning. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ (eds) Advances in neural information processing systems, vol 26. Curran Associates, Inc. (2013). https://proceedings.neurips.cc/paper/2013/file/e6d8545daa42d5ced125a4bf747b3688-Paper.pdf
18. Chen X, Wu ZS, Hong M (2020) Understanding gradient clipping in private sgd: a geometric perspective. ArXiv abs/2006.15429
19. Culnane C, Rubinstein BIP, Teague V (2019) Two data points enough to spot you in open transport records. https://pursuit.unimelb.edu.au/articles/two-data-points-enough-to-spot-you-in-open-transport-records
20. Cunha M, Mendes R, Vilela JP (2021) A survey of privacy-preserving mechanisms for heterogeneous data types. Comput Sci Rev 41:100403. https://doi.org/10.1016/j.cosrev.2021.100403
21. Dean J, Ghemawat S (2004) Mapreduce: simplified data processing on large clusters. In: OSDI'04: sixth symposium on operating system design and implementation, pp 137–150. San Francisco, CA
22. Ding B, Kulkarni J, Yekhanin S (2017) Collecting telemetry data privately. CoRR http://arxiv.org/abs/1712.01524

23. Dinur I, Nissim K (2003) Revealing information while preserving privacy. In: Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems. PODS ’03, Association for Computing Machinery, New York, NY, USA, pp 202–210. https://doi.org/10.1145/773153.773173
24. Dong J, Roth A, Su WJ (2020) Gaussian differential privacy. J Roy Stat Soc Ser B (Stat Methodol) 84
25. Duchi JC, Jordan MI, Wainwright MJ (2013) Local privacy and statistical minimax rates. In: 2013 IEEE 54th annual symposium on foundations of computer science, pp 429–438. https://doi.org/10.1109/FOCS.2013.53
26. Dwork C (2008) Differential privacy: a survey of results. In: Agrawal M, Du D, Duan Z, Li A (eds) Theory and applications of models of computation. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 1–19
27. Dwork C, Kenthapadi K, McSherry F, Mironov I, Naor M (2006) Our data, ourselves: privacy via distributed noise generation. In: Vaudenay S (ed) Advances in cryptology—EUROCRYPT 2006, 25th annual international conference on the theory and applications of cryptographic techniques, St. Petersburg, Russia, May 28–June 1, 2006, Proceedings. Lecture Notes in Computer Science, vol 4004. Springer, pp 486–503. https://doi.org/10.1007/11761679_29
28. Dwork C, McSherry F, Nissim K, Smith A (2017) Calibrating noise to sensitivity in private data analysis. J Privacy and Confidentiality 7(3):17–51. https://doi.org/10.29012/jpc.v7i3.405
29. Dwork C, Naor M, Reingold O, Rothblum G, Vadhan S (2009) On the complexity of differentially private data release: Efficient algorithms and hardness results. In: Proceedings of the 41st annual ACM symposium on theory of computing (STOC ‘09). Bethesda, MD (31 May–2 June 2009), pp 381–390. http://dl.acm.org/citation.cfm?id=1536467
30. Dwork C, Roth A (2014) The algorithmic foundations of differential privacy. Found Trends Theor Comput Sci 9(3-4):211–407. https://doi.org/10.1561/0400000042
31. Dwork C, Rothblum GN (2016) Concentrated differential privacy
32. Dwork C, Rothblum GN, Vadhan S (2010) Boosting and differential privacy. In: 2010 IEEE 51st annual symposium on foundations of computer science, pp 51–60. https://doi.org/10.1109/FOCS.2010.12
33. Dwork C, Smith A, Steinke T, Ullman J (2017) Exposed! a survey of attacks on private data. Ann Rev Stat Its Appl
34. Eliáš M, Kapralov M, Kulkarni J, Lee YT (2020) Differentially private release of synthetic graphs, pp 560–578. https://doi.org/10.1137/1.9781611975994.34
35. Erlingsson U, Feldman V, Mironov I, Raghunathan A, Talwar K, Thakurta A (2020) Amplification by shuffling: from local to central differential privacy via anonymity
36. Erlingsson Ú, Korolova A, Pihur V (2014) RAPPOR: randomized aggregatable privacy-preserving ordinal response. CoRR http://arxiv.org/abs/1407.6981
37. Fei S, Yan Z, Ding W, Xie H (2021) Security vulnerabilities of sgx and countermeasures: a survey. ACM Comput Surv 54(6). https://doi.org/10.1145/3456631
38. Fernandes N, Dras M, McIver A (2019) Generalised differential privacy for text document processing. ArXiv abs/1811.10256
39. Feyisetan O, Balle B, Drake T, Diethe T (2020) Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations. In: Proceedings of the 13th international conference on web search and data mining
40. Fioretto F, Tran C, Van Hentenryck P, Zhu K (2022) Differential privacy and fairness in decisions and learning tasks: a survey. https://doi.org/10.48550/ARXIV.2202.08187
41. Fredrikson M, Jha S, Ristenpart T (2015) Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. CCS ’15, association for computing machinery, New York, NY, USA, pp 1322–1333. https://doi.org/10.1145/2810103.2813677
42. Fredrikson M, Lantz E, Jha S, Lin S, Page D, Ristenpart T (2014) Privacy in pharmacogenetics: an end-to-end case study of personalized warfarin dosing. In: Proceedings of the 23rd USENIX conference on security symposium. SEC’14, USENIX Association, USA, pp 17–32

43. Fung BCM, Wang K, Chen R, Yu PS (2010) Privacy-preserving data publishing: a survey of recent developments. ACM Comput Surv 42(4). https://doi.org/10.1145/1749603.1749605
44. Ganta SR, Kasiviswanathan SP, Smith A (2008) Composition attacks and auxiliary information in data privacy
45. Garfinkel S, Abowd JM, Martindale C (Oct2018) Understanding database reconstruction attacks on public data: these attacks on statistical databases are no longer a theoretical danger. Queue 16(5):28–53. https://doi.org/10.1145/3291276.3295691
46. Geumlek J, Song S, Chaudhuri K (2017) Rényi differential privacy mechanisms for posterior sampling. CoRR http://arxiv.org/abs/1710.00892
47. Girgis A, Data D, Diggavi S, Kairouz P, Theertha Suresh A (2021) Shuffled model of differential privacy in federated learning. In: Banerjee A, Fukumizu K (eds) Proceedings of the 24th international conference on artificial intelligence and statistics. Proceedings of machine learning research, vol 130. PMLR (13–15 Apr 2021), pp 2521–2529. https://proceedings.mlr.press/v130/girgis21a.html
48. Girgis AM, Data D, Diggavi S (2021) Renyi differential privacy of the subsampled shuffle model in distributed learning
49. Golatkar A, Achille A, Wang YX, Roth A, Kearns M, Soatto S (2022) Mixed differential privacy in computer vision. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 8376–8386
50. Gong M, Xie Y, Pan K, Feng K, Qin A (2020) A survey on differentially private machine learning [review article]. IEEE Comput Intell Mag 15(2):49–64. https://doi.org/10.1109/MCI.2020.2976185
51. Goodfellow IJ (2015) Efficient per-example gradient computations. ArXiv abs/1510.01799
52. Google: Google’s privacy on beam library. https://github.com/google/differential-privacy/tree/main/privacy-on-beam
53. Hardt M, Rothblum GN (2010) A multiplicative weights mechanism for privacy-preserving data analysis. In: 2010 IEEE 51st annual symposium on foundations of computer science, pp 61–70. https://doi.org/10.1109/FOCS.2010.85
54. Hu R, Guo Y, Gong Y (2020) Concentrated differentially private and utility preserving federated learning
55. Huang X, Guan J, Zhang B, Qi S, Wang X, Liao Q (2019) Differentially private convolutional neural networks with adaptive gradient descent. In: 2019 IEEE fourth international conference on data science in cyberspace (DSC), pp 642–648. https://doi.org/10.1109/DSC.2019.00105
56. Iyengar R, Near JP, Song D, Thakkar O, Thakurta A, Wang L (2019) Towards practical differentially private convex optimization. In: 2019 IEEE symposium on security and privacy (SP), pp. 299–316. https://doi.org/10.1109/SP.2019.00001
57. Jain P, Thakurta AG (2014) dimension independent risk bounds for differentially private learning. In: Xing EP, Jebara T (eds) Proceedings of the 31st international conference on machine learning. Proceedings of machine learning research, vol 32. PMLR, Beijing, China, pp 476–484. https://proceedings.mlr.press/v32/jain14.html
58. Jiang H, Li J, Zhao P, Zeng F, Xiao Z, Iyengar A (2021) Location privacy-preserving mechanisms in location-based services: A comprehensive survey. ACM Comput Surv 54(1). https://doi.org/10.1145/3423165
59. Johnson N, Near JP, Song D (2018) Towards practical differential privacy for sql queries. Proc VLDB Endow 11(5):526–539. https://doi.org/10.1145/3177732.3177733
60. Kairouz P, Oh S, Viswanath P (2015) The composition theorem for differential privacy. In: Bach F, Blei D (eds) Proceedings of the 32nd international conference on machine learning. Proceedings of machine learning research, vol 37. PMLR, Lille, France, pp 1376–1385. https://proceedings.mlr.press/v37/kairouz15.html
61. Kamath G, Ullman J (2020) A primer on private statistics
62. Kaplan H, Mansour Y, Stemmer U (2020) The sparse vector technique. https://doi.org/10.48550/ARXIV.2010.00917
63. Kasiviswanathan SP, Lee HK, Nissim K, Raskhodnikova S, Smith AD (2008) What can we learn privately? CoRR http://arxiv.org/abs/0803.0924

64. Kasiviswanathan SP, Rudelson M, Smith A (2012) The power of linear reconstruction attacks
65. Kerrigan G, Slack D, Tuyls J (2020) Differentially private language models benefit from public pre-training. ArXiv abs/2009.05886
66. Kifer D, Smith A, Thakurta A (2012) Private convex empirical risk minimization and high-dimensional regression. In: Mannor S, Srebro N, Williamson RC (eds) Proceedings of the 25th annual conference on learning theory. Proceedings of machine learning research, vol 23. PMLR, Edinburgh, Scotland, pp 25.1–25.40. https://proceedings.mlr.press/v23/kifer12.html
67. Lee J, Kifer D (2021) Scaling up differentially private deep learning with fast per-example gradient clipping. Proc Privacy Enhancing Technol (1). https://doi.org/10.2478/popets-2021-0008
68. Li X, Tramèr F, Liang P, Hashimoto TB (2021) Large language models can be strong differentially private learners. ArXiv https://arxiv.org/abs/2110.05679
69. Ligett K, Neel S, Roth A, Waggoner B, Wu Z (2017) Accuracy first: selecting a differential privacy level for accuracy-constrained ERM. Advances in Neural Information Processing Systems 2017-December, pp 2567–2577. Publisher Copyright: 2017 Neural information processing systems foundation. All rights reserved.; 31st Annual Conference on Neural Information Processing Systems, NIPS 2017; Conference date: 04-12-2017 Through 09-12-2017
70. Liu B, Ding M, Shaham S, Rahayu W, Farokhi F, Lin Z (2021) When machine learning meets privacy: a survey and outlook. ACM Comput Surv 54(2). https://doi.org/10.1145/3436755
71. Liu J, Talwar K (2018) Private selection from private candidates
72. Luo Z, Wu DJ, Adeli E, Fei-Fei L (2021) Scalable differential privacy with sparse network finetuning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 5059–5068
73. McMahan HB, Moore E, Ramage D, Hampson S, Arcas BA (2016) Communication-efficient learning of deep networks from decentralized data. https://doi.org/10.48550/ARXIV.1602.05629
74. McMahan HB, Ramage D, Talwar K, Zhang L (2018) Learning differentially private recurrent language models
75. McSherry F (2010) Privacy integrated queries: an extensible platform for privacy-preserving data analysis. Commun ACM 53(9):89–97. https://doi.org/10.1145/1810891.1810916
76. McSherry F, Talwar K (2007) Mechanism design via differential privacy. In: 48th annual IEEE symposium on foundations of computer science (FOCS’07), pp 94–103. https://doi.org/10.1109/FOCS.2007.66
77. Mironov I (2012) On significance of the least significant bits for differential privacy. In: Proceedings of the 2012 ACM conference on computer and communications security. CCS ’12, Association for Computing Machinery, New York, NY, USA, pp 650–661. https://doi.org/10.1145/2382196.2382264
78. Mironov I (2017) Rényi differential privacy. In: 2017 IEEE 30th computer security foundations symposium (CSF), pp 263–275. https://doi.org/10.1109/CSF.2017.11
79. Mironov I, Talwar K, Zhang L (2019) Rényi differential privacy of the sampled gaussian mechanism
80. Mueller TT, Paetzold JC, Prabhakar C, Usynin D, Rueckert D, Kaissis G (2022) Differentially private graph classification with GNNS. ArXiv https://arxiv.org/abs/2202.02575
81. Murtagh J, Vadhan S (2016) The complexity of computing the optimal composition of differential privacy
82. Narayanan A, Shmatikov V (2006) How to break anonymity of the Netflix prize dataset. CoRR http://arxiv.org/abs/cs/0610105
83. Neel S, Roth A, Vietri G, Wu ZS (2019) Differentially private objective perturbation: beyond smoothness and convexity. ArXiv https://arxiv.org/abs/1909.01783
84. Nissim K, Raskhodnikova S, Smith A (2007) Smooth sensitivity and sampling in private data analysis. In: Proceedings of the thirty-ninth annual ACM symposium on theory of computing, pp 75–84. STOC ’07, Association for computing machinery, New York, NY, USA. https://doi.org/10.1145/1250790.1250803

85. Olatunji IE, Funke T, Khosla M (2021) Releasing graph neural networks with differential privacy guarantees. CoRR https://arxiv.org/abs/2109.08907
86. Papernot N, Abadi M, Erlingsson U, Goodfellow I, Talwar K (2016) Semi-supervised knowledge transfer for deep learning from private training data. https://doi.org/10.48550/ARXIV.1610.05755
87. Papernot N, McDaniel PD, Wu X, Jha S, Swami A (2015) Distillation as a defense to adversarial perturbations against deep neural networks. CoRR http://arxiv.org/abs/1511.04508
88. Papernot N, Song S, Mironov I, Raghunathan A, Talwar K, Erlingsson U (2018) Scalable private learning with pate. https://doi.org/10.48550/ARXIV.1802.08908
89. Papernot N, Steinke T (2021) Hyperparameter tuning with renyi differential privacy
90. Papernot N, Thakurta A, Song S, Chien S, Erlingsson Ú (2021) Tempered sigmoid activations for deep learning with differential privacy. In: AAAI
91. Paul S, Mishra S (2020) ARA: aggregated RAPPOR and analysis for centralized differential privacy. CoRR http://arxiv.org/abs/2001.01618
92. Paul S, Sengupta P, Mishra S (2020) Flaps: federated learning and privately scaling
93. Proserpio D, Goldberg S, McSherry F (2014) Calibrating data to sensitivity in private data analysis: a platform for differentially-private analysis of weighted datasets. Proc VLDB Endow 7(8):637–648. https://doi.org/10.14778/2732296.2732300
94. Rényi A (1961) On measures of entropy and information. In: Proceedings of the 4th Berkeley symposium on Mathematical Statistics and Probability, vol 1, pp 547–561
95. Rogers R, Subramaniam S, Peng S, Durfee D, Lee S, Kancha SK, Sahay S, Ahammad P (2020) Linkedin's audience engagements API: a privacy preserving data analytics system at scale
96. Roth A, Roughgarden T (2011) Interactive privacy via the median mechanism
97. Ru S, Zhang B, Jie Y, Zhang C, Wei L, Gu C (2021) Graph neural networks for privacy-preserving recommendation with secure hardware. In: 2021 international conference on networking and network applications (NaNA), pp 395–400. https://doi.org/10.1109/NaNA53684.2021.00075
98. Sajadmanesh S, Gatica-Perez D (2020) Locally private graph neural networks. https://doi.org/10.48550/ARXIV.2006.05535
99. Samarati P, Sweeney L (1998) Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Tech. rep. (1998)
100. Sarwate A, Chaudhuri K (2013) Signal processing and machine learning with differential privacy: algorithms and challenges for continuous data. Signal Process Mag IEEE 30:86–94. https://doi.org/10.1109/MSP.2013.2259911
101. Sengupta P, Paul S, Mishra S (2020) Buds: balancing utility and differential privacy by shuffling. https://doi.org/10.48550/ARXIV.2006.04125
102. Sengupta P, Paul S, Mishra S (2020) Learning with differential privacy. CoRR https://arxiv.org/abs/2006.05609
103. Sengupta P, Paul S, Mishra S (2021) Buds+: better privacy with converger and noisy shuffling. Digital Threats. https://doi.org/10.1145/3491259
104. Shokri R, Stronati M, Song C, Shmatikov V (2017) Membership inference attacks against machine learning models
105. Song S, Chaudhuri K, Sarwate AD (2013) Stochastic gradient descent with differentially private updates. In: 2013 IEEE global conference on signal and information processing, pp 245–248. https://doi.org/10.1109/GlobalSIP.2013.6736861
106. Sweeney L (2002) k-anonymity: a model for protecting privacy. Int J Uncertainty, Fuzziness Knowl-Based Syst 10(05):557–570. https://doi.org/10.1142/S0218488502001648
107. Tang J, Korolova A, Bai X, Wang X, Wang X (2017) Privacy loss in apple's implementation of differential privacy on macos 10.12. ArXiv https://arxiv.org/abs/1709.02753
108. Team ADP (2017) Learning with privacy at scale. https://arxiv.org/pdf/2109.08604.pdf
109. Tramèr F, Boneh D (2021) Differentially private learning needs better features (or much more data). In: International conference on learning representations (ICLR). https://arxiv.org/abs/2011.11660

110. Tran C, Dinh MH, Fioretto F (2021) Differentially private deep learning under the fairness lens. CoRR https://arxiv.org/abs/2106.02674
111. Truex S, Baracaldo N, Anwar A, Steinke T, Ludwig H, Zhang R (2019) A hybrid approach to privacy-preserving federated learning. Informatik Spektrum 1–2
112. Truex S, Liu L, Chow KH, Gursoy ME, Wei W (2020) Ldp-fed: Federated learning with local differential privacy
113. Vadhan S (2017) The complexity of differential privacy. Springer, Yehuda Lindell, pp 347–450. https://doi.org/10.1007/978-3-319-57048-8_7
114. Wang T, Zhang X, Feng J, Yang X (2020) A comprehensive survey on local differential privacy toward data statistics and analysis. Sensors 20(24):7030. https://doi.org/10.3390/s20247030
115. Wang W, Wang T, Wang L, Luo N, Zhou P, Song D, Jia R (2021) Dplis: boosting utility of differentially private deep learning via randomized smoothing. https://doi.org/10.48550/ARXIV.2103.01496
116. Warner SL (1965) Randomized response: a survey technique for eliminating evasive answer bias. J Am Stat Assoc 60(309):63–69. http://www.jstor.org/stable/2283137
117. Wasserman LA, Zhou S (2008) A statistical framework for differential privacy. J Am Stat Assoc 105:375–389
118. Wei K, Li J, Ding M, Ma C, Yang HH, Farokhi F, Jin S, Quek TQS, Poor HV (2020) Federated learning with differential privacy: algorithms and performance analysis. IEEE Trans Inform Forensic Secur 15:3454–3469. https://doi.org/10.1109/TIFS.2020.2988575
119. Williams O, Mcsherry F (2010) Probabilistic inference and differential privacy. In: Lafferty J, Williams C, Shawe-Taylor J, Zemel R, Culotta A (eds) Advances in neural information processing systems, vol 23. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2010/file/fb60d411a5c5b72b2e7d3527cfc84fd0-Paper.pdf
120. Wilson RJ, Zhang CY, Lam WKC, Desfontaines D, Simmons-Marengo D, Gipson B (2020) Differentially private sql with bounded user contribution. Proc Privacy Enhancing Technol 2020:230–250
121. Xie Y, Li P, Wu C, Wu Q (2021) Differential privacy stochastic gradient descent with adaptive privacy budget allocation. In: 2021 IEEE international conference on consumer electronics and computer engineering (ICCECE), pp 227–231
122. Xiong X, Liu S, Li D, Cai Z, Niu X (Oct2020) A comprehensive survey on local differential privacy. Secur Commun Netw 2020:8829523. https://doi.org/10.1155/2020/8829523
123. Xu J, Zhang W, Wang F (2021) A(dp)2sgd: asynchronous decentralized parallel stochastic gradient descent with differential privacy. IEEE Trans Pattern Anal Mach Intell
124. Yu D, Zhang H, Chen W, Yin J, Liu TY (2021) Large scale private learning via low-rank reparametrization. https://doi.org/10.48550/ARXIV.2106.09352
125. Zhang C, Bengio S, Hardt M, Recht B, Vinyals O (2016) Understanding deep learning requires rethinking generalization. CoRR http://arxiv.org/abs/1611.03530
126. Zhang L, Zhu T, Xiong P, Zhou W, Yu PS (20210 More than privacy: adopting differential privacy in game-theoretic mechanism design. ACM Comput Surv 54(7). https://doi.org/10.1145/3460771
127. Zhao Y, Chen J (2021) A survey on differential privacy for unstructured data content. ACM Comput Surv. https://doi.org/10.1145/3490237
128. Zhu Y, Wang YX (2020) Improving sparse vector technique with renyi differential privacy. In: Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H (eds) Advances in neural information processing systems, vol 33. Curran Associates, Inc., pp 20249–20258. https://proceedings.neurips.cc/paper/2020/file/e9bf14a419d77534105016f5ec122d62-Paper.pdf
129. Zhu Y, Yu X, Chandraker M, Wang YX (2020) Private-KNN: practical differential privacy for computer vision. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)

Chapter 8
Secure Certificateless Maximum Achievable Throughput in Successive IoT Relay Networks

Akash Kumar, Sujit Sangram Sahoo, Sumit Kumar, Pravas Ranjan Bal, Jitendra Kumar Rout, and Vijay Kumar Chaurasiya

Abstract We assess the benefits of using simple and successive relaying in energy harvesting networks and compare the simple and successive relay networks in terms of maximum achievable throughput. Both the relay and source have the potential to harvest energy and do not generate any traffic on their own. This study investigates the interplay between energy harvesting, relaying, and stability. We assume that the source transmits to the destination using relay-based network cooperation. The relay regulates the source arrival data packets by accepting the part of the successfully received data packets received in the relay. However, our work is based on two relay networks. In the case of a simple relaying network, we exploit the partial relay cooperation in which flow can be controlled by partial relaying. The system should be stable when the relay and source are stable, and we applied the stability condition individually for both the relay and source. Then the stability for both the relay and the source are combined to get a stable system. The relay network secure communication depends on certificateless cryptography and the communication cost is very less. Our

A. Kumar
Department of CSE, Siksha 'O' Anusandhan, Bhubaneswar 751030, India
e-mail: akashkumarcse@soa.ac.in

S. S. Sahoo (✉)
Department of Information Technology, Indian Institute of Information Technology, Allahabad, Prayagraj, India
e-mail: sssahoo@ieee.org

S. Kumar
Department of CSE, Thapar Institute of Engineering and Technology, Patiala, India
e-mail: sumit.093040@gmail.com

P. R. Bal
Birla Institute of Technology Mesra, Ranchi, Jharkkand, India

J. K. Rout
Department of CSE, National Institute of Technology Raipur, Raipur, India
e-mail: jkrout.cs@nitrr.ac.in

V. K. Chaurasiya
Department of Information Technology, Indian Institute of Information Technology-Allahabad, Prayagraj, India
e-mail: vijayk@iiita.ac.in

D. Gountia et al. (eds.), *Information Technology Security*, Springer Tracts in Electrical and Electronics Engineering, https://doi.org/10.1007/978-981-97-0407-1_8

model work for small devices and provides the security authenticity with a trusted third party. We compare the successive partial relay network's approach performance with the full relay, no-relay, and simple relay performance.

Keywords Certificateless cryptography · Replay attack · Elliptic curve · Digital signature · Energy harvesting · Relay networks · Throughput · Cooperative communication

Introduction

Security is the main concern of the communication network. It is very much essential in the IoT node communication. The nodes are very less capacity in terms of the memory and computation. So, they require the lightweight cryptography for digital data transfer. the lightweight cryptography requires less communication with the certificate authority or no certificate authority. Instead of certificate-based cryptography the certificateless cryptography is more crucial in terms of cost-effective for less power devices. To make the system more lightweight with less bit of data the elliptic curve cryptography is required. Further, the security achievements need the digital signature and cryptographic hash function to make the system authentic. Moreover, they require the public and private keys for the data communication. All the above considers the security but the communication must be updated with considering the energy requirements. Again, to make the efficient network the energy flow with very less capacity makes the network smart and it considers the wireless network.

Now things have become smart to connect with the internet. Wireless networks work as a bridge between things and the internet. Sensor nodes perform different tasks based on the application requirement and send them to the cloud device for further computing operation with the help of the wireless relay nodes. The demand for IoT-based wireless networks is increasing, which requires both secure communication and a longer lifetime of the networks. It needs to add some security schemes for wireless networks to make reliable communication. To increase the lifetime of networks, it adopts energy harvesting techniques [1–3]. Energy harvesting techniques involve the process of harvesting and utilizing the energy available from renewable sources in the environment. The goal of energy harvesting is to convert these renewable energy forms into usable electrical power for various applications. The renewable energy source makes them the infrastructural less wireless network [4] into operational mode without any need of battery replacement [5]. Different types of renewable environment energy sources are used to convert into electrical energy to charge the wireless network energy storage devices such as batteries, super-capacitor/capacitor [1–3, 6–9]. Various networks have been used to recharge sensor nodes, including rechargeable sensor networks [10] and energy harvesting active networked tags (EnHANTs) [11, 12]. Energy harvesting networks have gained significant attention due to their applications in various fields. Exploring different aspects of these networks is essential for understanding their potential and optimizing their performance. One crucial aspect to consider is the modeling of energy as an exogenous stochastic process in systems where nodes extract energy from the environment. An exogenous rechargeable processes use energy from renewable resources including thermal, wind, solar, and vibration energy. Nodes within the network harness this energy to power their operations. The energy harvested from renewable sources can be buffered in batteries, enabling the nodes to store excess energy for later use. This stored energy is then utilized to support data transmission and other activities within the network. Unlike, battery-powered systems, energy is unpredictability or non-deterministic in nature. It follows a random process with stochastic fluctuations over time. Without considering the energy harvesting mechanism, we treat harvested energy as a stochastic process. When dealing with non-rechargeable battery-powered nodes, our common goal is to maximize network battery lifetime or reduce battery use [13, 14]. So, we use the half-duplex relaying scheme for both the case of simple relaying network and successive relaying network. In the half-duplex technique, we can not transmit or receive the data simultaneously, but maximize the lifetime of the network. However, the ability to harvest enables us to take into account the steady-state performance measures of a network, including its throughput, fairness, and stability. Cooperative diversity allows individual users with single antennas to leverage spatial diversity by utilizing relay nodes to transmit data. Employing multiple relays, as opposed to a single relay, extends coverage and reduces the need for high transmit power [15–17]. Relay selection schemes are categorized into two parts: single and multiple relay selection schemes. The complexity of the relay system exponentially increases as the number of nodes increases in the relay scheme [18, 19]. Using the relay node, we select two transmission schemes for sending information from source to destination. The first transmission scheme is a simple partial relaying scheme, and the second is a successive one. This paper considers a simple relay system consisting of a source, relays, and destination. Both relays and sources in this system are equipped with energy harvesting capability, and all nodes operate within the same frequency band. Packet arrival at the source and energy arrival at both the relay and source are modeled as discrete-time stochastic processes. We focus on a two-hop network with a direct line of sight between the source and destination, allowing each packet to reach the destination by passing through no more than one relay. Investigating a two-hop network is essential and provides significant insights into understanding and addressing network challenges. Its simple model deliberately provides insights into the interactions between relaying, energy harvesting, and stability in the cases of a simple relay network and successive relay network.

This paper determines the maximum achievable throughput of the simple and successive relaying networks, which contain a source, two relay nodes, and a destination. The relay regulates the relaying process by accepting a proportion of successfully received packets from the source and only transmits packets to the common media when the source is idle and starts evaluating and combining the stability condition for the relay and source data queues for both simple and successive relay networks. In the simple relay network, we also characterized the maximum achievable throughput for two special cases, no relaying, and full relaying. We characterize the maximum achievable throughput exact nature in both cases of the system. Moreover, previous work only approximates the characteristics that describe it. Partial cooperation was used in [20, 21] for non-energy harvesting relays. We can summarize the work as follows:

1. Our research focuses on examining the interaction among cooperative relaying, energy harvesting, and stability within wireless networks.
2. For simple relaying network, determine the maximum achievable throughput of the partial relaying strategy for the relay and source without generating its traffic.
3. In our study, we determine optimal relaying parameter value that maximizes the stable source throughput, considering a fixed relay throughput. This optimal value is derived in closed form, representing a mathematical expression that depends on the system parameters.
4. Calculate the maximum achievable throughput for successive relaying strategies and compare them with successive relay networks' no-relaying and full-relaying strategies.
5. The proposed model provides secure communication with very less communication cost and the signatures are lightweight.

The paper is structured as follows: a comprehensive literature survey discussed in Sect. 8.2, presenting an in-depth discussion of existing research in the field. Section 8.3 introduces the system model, providing a detailed explanation of the proposed approach. Section 8.4 showcases the results, while Sect. 8.5 presents the conclusion and the implications of our findings.

Related Work

In recent years, significant research has been conducted to investigate the transmission process in networks powered by energy harvesting [22–34]. Lie et al. [22], proposed a generic mathematical framework for optimizing single-hop transmission strategies in replenishable sensor networks while accounting for stochastic energy replenishment uncertainty. Based on Markov chain models and threshold values, the suggested optimum transmission policy improves the average reward rate significantly over the unconditional transmit-all strategy. In [23], described energy management strategies for energy harvesting sensor nodes to attain energy-neutral operation, optimal throughput, and mean latency. Extensions to fading channels and energy consumption in sensing are explored, emphasizing the benefits of energy storage and real-life data validation. Ho et al. [24] investigated energy allocation for wireless communications using energy harvesters, considering time-selective fading and unreliable energy sources. The aim is to maximize the throughput considering that energy allocation over the finite horizon and consider the available transmission energy and communication varies for every data transmission slot. It considers the optimal energy allocation to obtain the structural results, achieved with the help of convex optimization techniques and dynamic programming. Yang et al. [26] consider the adaptive change available energy of the nodes and data transmission rate dependent on the traffic load to achieve the minimum transmission completion time for single-user energy harvesting wireless communication system with the assumption that nodes are aware of the time required to harvest energy and the amount of energy harvested before transmission. Tutuncuoglu et al. [27] focused on transmission policies for energy harvesting wireless networks under the constraints on the battery storage or capacity and energy replenishment process. Optimal solutions are identified for minimizing transmission completion time and maximizing short-term throughput, taking into account battery capacity constraints and energy causality. The derived necessary conditions and algorithm validate the analytical findings through numerical results. Ozel et al. [28] addressed the optimization of the point-to-point data transmission in the wireless systems with rechargeable nodes and limited battery capacity. This problem achieves two objectives: maximizing throughput by the deadline and minimizing transmission completion time by the given amount of data transmission completed. Optimal offline policies are studied using a directional water-filling algorithm, while online policies are developed using stochastic dynamic programming. There are other near-optimal strategies with reduced complexity also presented. Numerical assessments are performed to evaluate the performance of various policies in various configurations. Kashef et al. [29] investigated the challenge of maximizing the average number of successfully transmitted packets per time slot at a source node equipped with energy harvesting capability over a time-varying channel. The optimal policy is demonstrated to be a threshold-type policy based on energy queue and channel state, and an upper bound on the optimal policy's performance is obtained. The effect of the channel's time-varying nature and the availability of channel state information (CSI) is also investigated, demonstrating that a greedy approach is the best option in some instances. Yang et al. [30] investigated the transmission completion time minimization problem in an AWGN broadcast channel with energy harvesting capabilities. The optimal transmission policy is derived, demonstrating structural properties and power allocation based on cut-off power levels. The analysis is extended to an M-user broadcast channel, and to find the globally optimal policy algorithm is proposed. Under various settings, the performance of the optimal policy is compared with sub-optimal policies. The primary goal of Antepli et al. [31] is established the structural properties of the optimum schedule and optimal policy uniqueness under specific assumptions that all weaker user bits are available at the beginning. Ozel et al. [33] addressed the optimal offline transmission policy for minimization of the transmission completion time by successfully transmitting the data packets to their destination in the two-user AWGN broadcast channel. The optimal offline transmission policy is derived using a directional water-filling algorithm, considering the battery's finite storage capacity. By performing directed water-filling repeatedly, the suggested method obtains the globally optimum policy.

Existing literature has introduced various relay selection schemes, highlighting the performance improvements achieved by selecting a single relay among multiple options. However, our work distinguishes itself by focusing on energy harvesting nodes, where relays have random energy availability. This paper examines the advantages of utilizing simple and successive relaying in energy harvesting networks. The study investigates the interplay between energy harvesting, relaying, and system stability.

System Model

Simple Relay Network

Figure 8.1 represents the simple relay network, which consists of a source (S), a relay (R_1, R_2), and a destination (D) nodes.

We assign an index to each node, with relay nodes having an index of two. The source and relay nodes have infinite data queues, denoted as Q1 and Q2, respectively. Fixed-length packets are stored in these queues. It is assumed that the source node (S) creates its own traffic, whereas the relay node (R_1, R_2) does not and is solely used to cooperate with the source. The data arriving at the source queue follows the Bernoulli process. Each relay and source node has infinite energy queue and operates in only half-duplex mode. Our analysis utilizes a slotted time model where

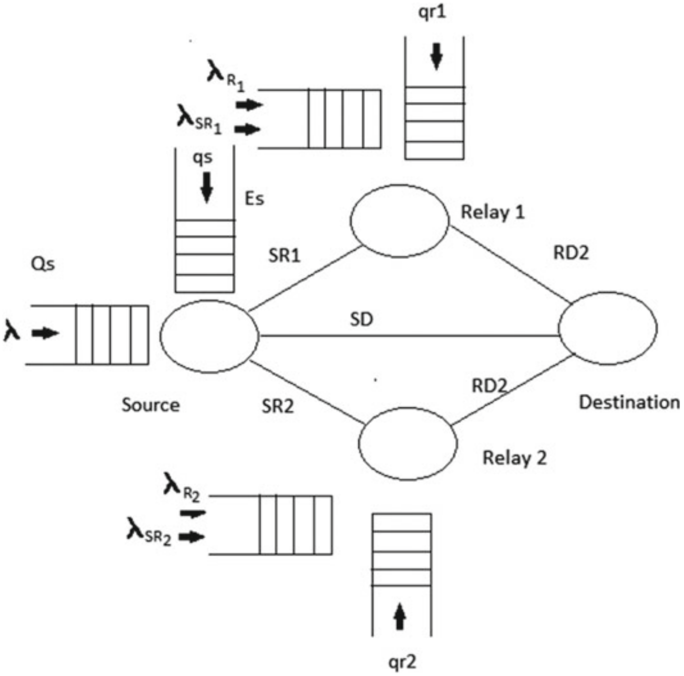


Fig. 8.1 Simple two relay network

every time slot represents the time required to transmit a single data packet. In this model, we consider that one unit of energy depletes from the energy queue during packet transmission. To decrease the complexity of the model, we assume energy is consumed only during data transmission, not in data processing and reception. When the destination cannot take the data directly through the source, data reach the destination node by accepting the packets in relay nodes, which store the data in the queue with the lowest index value. Therefore, the index is assigned to a relay node; a lower value signifies a higher priority in storing received packets. Nodes with a higher rate of energy harvesting are capable of making more transmission attempts compared to nodes with a lower energy harvesting rate. Similarly, nodes with a higher average success probability for their channel to the destination necessitate fewer retransmissions in comparison to nodes with a lower average success probability. We propose the success probability and product of the energy harvesting rate based on order criterion. The node's index value is arranged so that lower index values represent the higher production value to analyze the effect of the average channel success probability and energy arrival rate. In partial relay cooperation, every relay node accepts a specific proportion of successfully received packets. The proportion of received packets should align with the relay node's capacity to forward them. According to Loynes' theorem, if a queue's service and arrival processes are jointly stationary, the queue remains stable if the average service rate is higher than the average arrival rate. The fraction of packets received by relay i is determined by the flow control parameter r_i.

Successive Relay Network

Figure 8.2 represents the Successive relay network, which consists of a source (S), a relay (R_1, R_2), and a destination (D) nodes.

Successive relaying, also known as multiple-hop within spatial reuse, in successive relaying uses of phase 1 and phase 2. In the successive relaying phase 1, the source transmits the data R_1, R_2 transmits the data destination (D), and the successive relaying phase 2 source transmits the data R_2 and R_1 transmit the data destination (D). To maintain generality, we assume that in phase I of the network operation, relay node R_2 initiates transmission by delivering a non-zero quantity, denoted as $\xi > 0$, of dummy information, as it does not have any actual data to transmit at that time. To prevent any information loss in throughput, we can schedule phases 1 and 2 in succession while keeping the value of ξ small and neglecting ξ for convenience. The channels in the network, namely SD (source to destination), SR_1 (source to relay 1), SR_2 (source to relay 2), RD_1 (relay 1 to destination), and RD_2 (relay 2 to destination), are assumed to be independent erasure channels. The channels are independent of the energy harvesting processes and packet arrival at the relay and source. The packet's success probability represents the channel's quality-packet success probability denoted by the f_{SD} and f_{SR_i}, and f_{RD_i}. We employ partial relaying cooperation, where the relay node only accepts a certain proportion of the pack-

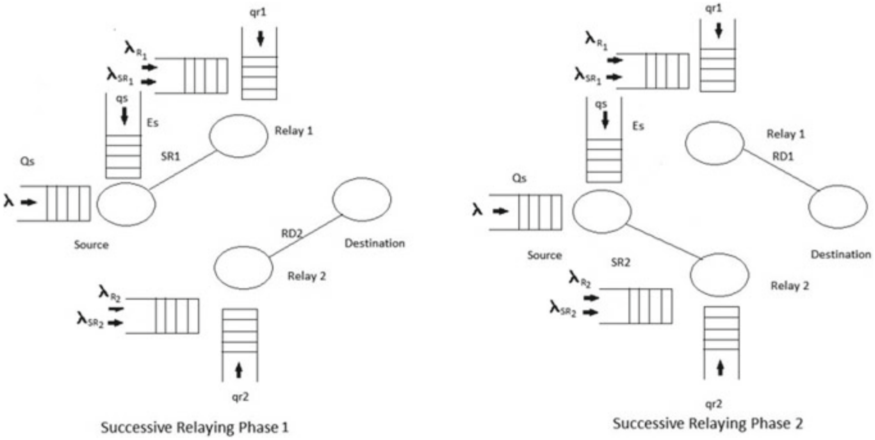


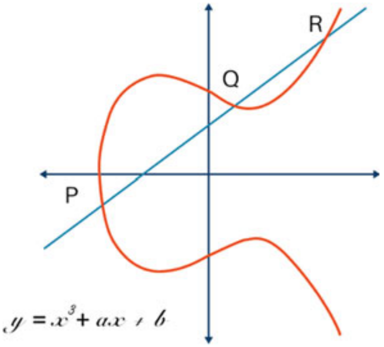
Fig. 8.2 Successive relay network for two relay nodes

ets that have been successfully received. It is important to distinguish between the acceptance of packets and their successful reception at the relay. Packet acceptance refers to keeping a packet in the relay's data queue and removing it from the source's data queue. On the other hand, successful reception means that the packet has been transmitted through the channel without errors. Not all successfully received packets are necessarily accepted at the relay due to limitations imposed by the available harvested energy. The relay's ability to forward packets determines the proportion of packets that can be accepted. The relaying parameter determines this packet proportion, which represents the probability of accepting a packet at the relay's data queue, given that the packet has been successfully received. In the paper referenced as [34], according to Loynes' theorem if a queue's arrival and service processes are jointly stationary, the queue remains stable if the average service rate is higher than the average arrival rate. The notations λ, λ_{SR_i}, μ_S, and μ_{R_i} help quantify the arrival and service rates at both the relay and source data queue, which are essential parameters for assessing the stability of the queues based on Loynes' theorem.

Secure Relay Networks

IoT uses open communication process and the network is easily accessible to participating parties. Further, the attacker accesses the data in an authorized way and it may modify the important information. So, relay networks require secure communication to transfer data from source to destination. They achieve all the communications through secure cryptographic signatures. But, the signatures should be cost-effective and lightweight. To achieve the secure communication, the most efficient cryptogra-

Fig. 8.3 Elliptic curve for signature generation



phy is elliptic curve cryptography($\mathcal{ECC}$). Moreover, it is very much important that to achieve less communication cost by using the certificateless cryptography [35].

Elliptic Curve Cryptography

$\mathcal{ECC}$ is a non-singular curve based on the finite field F_p and it is denoted by $E(F_p)$. The non-singular curve has many points which satisfy the $\mathcal{ECC}$ equation $y^2 = x^3 + ax + b$ mod p and the point of infinity is denoted by O. Both a and b ∈ to F_p and and satisfy the equation $4a^3 + 27b^2 \neq 0$.

We assume elliptic curve discrete logarithmic problem is hard where $D = k.g$, g is the generator point of group G and $k \in Z_p{}^*$.

Certificateless Signature in Relay Network Communication

The certificateless signature scheme contains key generation center ($\mathcal{KGC}$) for providing the partial private key. Basically, in certificateless scheme, the partial private and user secret creates the full private key[36].

1. Setup: It requires the large prime p and the finite field F_p. The one-way hash function H: {0, 1}* $\rightarrow Z_p$. The public parameters are $g \in Z_p{}^*$, H and n.
2. Partial key: The source and destination node of relay networks provide the own identity and the $\mathcal{KGC}$ generates a random secret $j \in Z_p{}^*$ and calculates the partial key L for both source and destination node differently. Again $\mathcal{KGC}$ calculates the partial private key, i.e., M and that uses the users($\mathcal{U}$) identity, hash function, and the random secret j. $\{sourcenode, destinationnode, Relay1, relay2\} \in \mathcal{U}$. We consider the communication process for both source and destination nodes in the relay networks and the actual process takes all the $\mathcal{U}$ and creates the parameters as well.
3. Private key: The private key generation requires the partial private key, the identity of source and destination node, and $\mathcal{U}$ secrets.

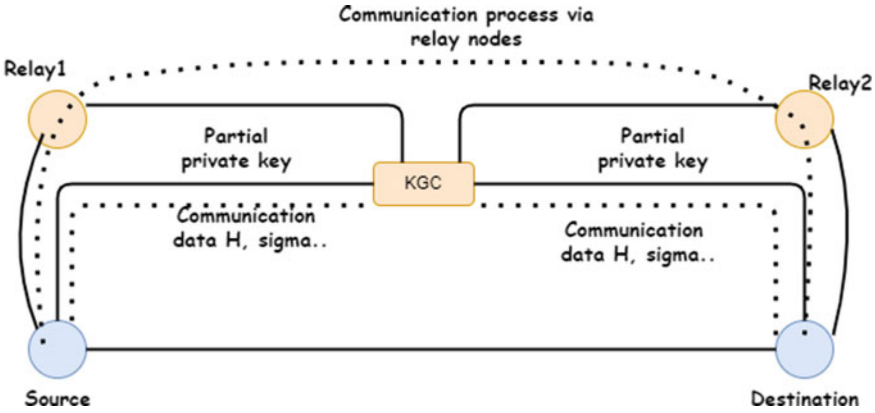

Communication process via relay nodes
Relay1
Relay2
Partial private key
Partial private key
KGC
Communication data H, sigma..
Communication data H, sigma..
Source
Destination

Fig. 8.4 Secure communication process

4. Public Key: Both $\mathcal{U}$ source and destination node get the public key from their respective private key. The process also requires the identity of $\mathcal{U}$.
5. Signature Generation: Both $\mathcal{U}$ generate the signature using their private key. When data transfer requires from source to destination, the source node sends the data with his signature to the destination. It requires the message and the H of the data. In every process of data transfer, the partial key can be provided by $\mathcal{KGC}$. If the data transfer will not happen directly then the participation of relay node will be considered. But, the relay node can do the signature process and communication as other base nodes as required.
6. Signature Verification: The signature verification will be done by destination. But, the communication happens when via the relay nodes then the signature verification process will happen in every receiver in a single path. It requires the public key of the source node to verify the signature.

All the processes consider the participating entities with their requirements. It is very much important that the communication process must be secure and authentic. The $\mathcal{KGC}$ is a trusted authority which has no knowledge regarding the communication process and it only provides the partial key. Again, it will not create any major problem to the the whole process without the private key and the private key requires user secret which is not easy without a particular user compromise. It is obvious that the authentication with the hash function H. The whole network ensures the security and authenticity with the help of the above process (see Fig. 8.4). The security of the model ensures confidentiality, integrity, replay attack, anonymity, non-repudiation, and authentication by the proposed signature scheme and is available in Fig. 8.5 (Table 8.1).

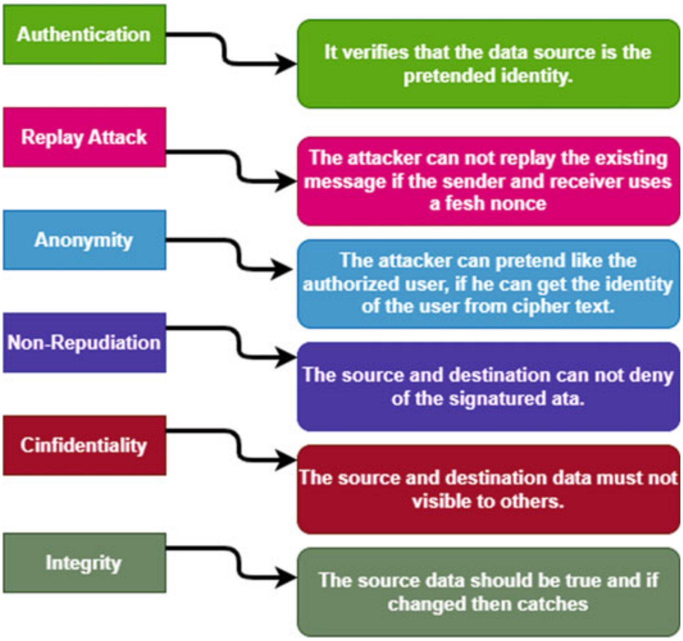

Authentication
It verifies that the data source is the pretended identity.
Replay Attack
The attacker can not replay the existing message if the sender and receiver uses a fesh nonce
Anonymity
The attacker can pretend like the authorized user, if he can get the identity of the user from cipher text.
Non-Repudiation
The source and destination can not deny of the signatured ata.
Cinfidentiality
The source and destination data must not visible to others.
Integrity
The source data should be true and if changed then catches

Fig. 8.5 Proposed model security ensureness

Stability Analysis

Simple Relay Network

The system remains stable as long as both the relay (R_i, where $i = 1, 2$) and source data queues are stable. The stability requirements for each queue are determined independently. For analysis of the stability describe with the help of the two probabilities. The probability of the packet received by the relay ($R_i(i = 1, 2)$), which is transmitted by the source, is calculated as the following equation:
For relay R_1,

$$P_{r_1} = r_1 f_{SR_1}(1 - f_{SD})(1 - r_2 f_{SR_1}) \tag{1}$$

For relay R_2,

$$P_{r_2} = r_2 f_{SR_2}(1 - f_{SD})(1 - r_1 f_{SR_2}) \tag{2}$$

Probability of the source node packets acceptance either by the relay R_1 and R_2 or by the destination denoted by the P_E. The following formula is used to get its value:

$$P_E = 1 - (1 - r_1 f_{SR_1})(1 - f_{SD})(1 - r_2 f_{SR_1}) \tag{3}$$

Table 8.1 Notation

Variable	Description
f_{SD}	Packet success probabilities from source to destination channel.
f_{SR_i}	Packet success probabilities from source to Relay R_i(i=1,2) channel
f_{RD_i}	Packet success probabilities Relay R_i(i=1,2) to destination channel
P_E	In the simple relay network, the probability of the source node packets acceptance either by the relay R_1 and R_2 or by the destination. destination. However, in the case of successive relaying, the packet is accepted only by the relay
P_{R_i}	Relay node packet acceptance probability, which is transmitted by the source
$P_r[E_{r_i} \neq 0]$	The probability of the relay R_i $(i = 1, 2)$ energy queue is not empty
q_S	Rate of energy arrival at the source node
μ_{R_i}	The data queue average service rate
q_{R_i}	Rate of energy arrival at the relay R_i(i=1,2) nodes
μ_S	The source data queue average service rate
λ	The source data queue average arrival rate
λ_{SR_i}	Average arrival rate of the R_i (i=1,2) data queue due to source transmission
ρ_S	Probability that channel is occupied by the source transmission
r_i	Optimal relaying parameter
Q_S	Source data queue
Q_{R_i}	(i=1,2) data queue
E_S	Source energy queue
E_{R_i}	Relay energy queue
$P_r[E_S \neq 0]$	The probability of the source energy queue is not empty
ρ_{R_i}	Probability that R_i transmission occupied the channel

Probability of the source energy queue is not empty,

$$P[E_s \neq 0] = q_S \tag{4}$$

Probability that source transmission occupied the channel is

$$\rho_s = \frac{\lambda}{P_E} \tag{5}$$

For relay R_1, SR_1 value computed as

$$\lambda_{SR_1} = \frac{\lambda P_{R_1}}{P_E} \tag{6}$$

For relay R_2

$$\lambda_{SR_2} = \frac{\lambda P_{R_2}}{P_E} \tag{7}$$

The probability that R_i $(i = 1, 2)$ transmission occupied the channel,
For relay R_1,

$$\rho_{R_1} = \frac{\lambda_S R_1}{f_{RD_1}} \tag{8}$$

For relay R_2,

$$\rho_{R_2} = \frac{\lambda_S R_2}{f_{RD_2}} \tag{9}$$

Probability of relay R_i $(i = 1, 2)$ energy queue is not empty, For relay R_1,

$$P_r[E_{R_1} \neq 0] = \frac{\min(q_{R_1}, 1 - \rho_S - \rho_{R_2})}{1 - \rho_S - \rho_{R_2}} \tag{10}$$

For relay R_2,

$$P_r[E_{R_1} \neq 0] = \frac{\min(q_{R_2}, 1 - \rho_S - \rho_{R_1})}{1 - \rho_S - \rho_{R_1}} \tag{11}$$

The product of the $P_r[E_S \neq 0]$ and P_E represent the service rate (μ_S) of the source data queue, whereas $P_r[E_S \neq 0]$ represents the probability that the source energy queue is not empty.

$$\mu_S = P_r[E_S \neq 0]P_E \tag{12}$$

The relay R_i (i=1,2) data queue service rate is calculated as follows:
For relay R_1,

$$\mu_{R_1} = f_{RD_1} \min(q_{R_1}, 1 - \rho_S - \rho_{R_2}) \tag{13}$$

For relay R_2,

$$\mu_{R_2} = f_{RD_2} \min(q_{R_2}, 1 - \rho_S - \rho_{R_1}) \tag{14}$$

The system is stable if and only if both the relay $R_i (i = 1, 2)$ and source data queue have to be stable. If $\lambda \leq \mu_S$ and $\lambda_{SR_i} \leq \mu_{R_i}$, here $(i = 1, 2)$ both condition should be satisfied. So the stability condition for R_i (i=1,2) is

$$\begin{aligned}\lambda < \min\Bigg(&\mu_S, q_{R_1} f_{RD_1} \frac{P_E}{P_{R_1}}, q_{R_2} f_{RD_2} \frac{P_E}{P_{R_2}}, f_{RD_1} f_{RD_2} \frac{P_E}{(f_{RD_2} + 1)P_{R_1} + f_{RD_1} f_{RD_2}}, \\ &f_{RD_1} f_{RD_2} \frac{P_E}{(f_{RD_1} + 1)P_{R_2} + f_{RD_1} f_{RD_2}}\Bigg)\end{aligned} \tag{15}$$

For partial relaying optimization, r_i optimal value is the solution of the problem, we applied the relationship between relaying parameter 1 and 2 that $r_2 = \eta r_1$, where $\eta =$ any real value.

For relay R_1,

$$r_1 = \min\Big(1, \max\Big(\Big(\Big(\frac{1-q_S}{q_S}\Big)(f_{RD_1} - f_{RD_2})\Big(1 - \frac{1}{(1+f_{RD_1})(1+f_{RD_2})}\Big) \\ \frac{1}{(f_{SR_1} - f_{SR_2}\mu_S)}, \frac{q_{R_1}f_{RD_1} - q_{R_2}f_{RD_2}}{(q_S(1-f_{SD})(f_{SR_1} - \mu_S f_{SR_2}))}\Big)\Big) \tag{16}$$

For relay R_2,

$$r_2 = \min\Big(1, \max\Big(\eta\Big(\frac{1-q_S}{q_S}\Big)(f_{RD_1} - f_{RD_2})\Big(1 - \frac{1}{(1+f_{RD_1})(1+f_{RD_2})}\Big) \\ \frac{1}{(f_{SR_1} - f_{SR_2}\mu_S)}, \eta\frac{q_{R_1}f_{RD_1} - q_{R_2}f_{RD_2}}{(q_S(1-f_{SD})(f_{SR_1} - \mu_S f_{SR_2}))}\Big)\Big) \tag{17}$$

In a case of no relaying parameter value is zero, the source will directly transfer the data to the destination. In a case of full relaying relaying parameter value is one, the source will send the through relay only.

Successive Relay Network

Phase 1

Source transmitted to data relay R_1, and relay R_2 transmits to the data destination. P_{R_1} represents relay node R_1 packet acceptance probability, which is transmitted by the source,

$$P_{R_1} = r_1 f_{SR_1} \tag{18}$$

The value of P_{E_1} is

$$P_{E_1} = r_1 f_{SR_1} \tag{19}$$

Probability that the source energy queue is not empty,

$$P_r[E_{S_2}] = q_S \tag{20}$$

The probability that the source transmission occupies channel 1,

$$\rho_{S_1} = \frac{1}{2} \tag{21}$$

SR_1 value is calculated as follows:

$$\lambda_{SR_1} = \lambda\frac{P_{R_1}}{P_{E_1}} \tag{22}$$

The probability that R_1 transmission occupied the channel,

$$\rho_{R_1} = \frac{\lambda_{SR_1}}{f_{RD_1}} \tag{23}$$

The service rate of the S data queue,

$$\mu_{S_1} = P_r[E_{S_1} \neq 0]P_{E_1} \tag{24}$$

Probability that energy queue of the relay node R_1 is not empty,

$$P_r[E_{R_1} \neq 0] = \frac{\min(q_{R_2}, \frac{1}{2} - \rho_{R_2})}{\frac{1}{2} - \rho_{R_2}} \tag{25}$$

The data queue relay R_1 service rate is calculated as follows:

$$\mu_{R_1} = f_{RD_1} \min(q_{R_1}, \frac{1}{2} - \rho_{R_2}) \tag{26}$$

If both the relay R_1 and source data queues are stable, the system is stable. If $\lambda_1 < \mu_{S_2}$ and $\lambda_{SR_1} < \mu_{R_1}$ both condition should be satisfied. So stability condition for relay R_1 is

$$\lambda_1 < \min\left(q_S P_{E_1}, q_{R_1} f_{RD_1}, f_{RD_1} \frac{f_{RD_2}}{2(f_{RD_1} + f_{RD_2})}\right) \tag{27}$$

For partial relaying optimization, r_1 optimal value is the solution of the problem,

$$r_1 = \min\left(1, \max\left(\frac{q_{R_1} f_{RD_1}}{q_S f_{SR_1}}, \frac{f_{RD_1} f_{RD_2}}{2q_S f_{SR_1}(f_{RD_1} + f_{RD_2})}\right)\right) \tag{28}$$

Phase 2

Source transmitted to data relay R_2 and transmitted to relay R_1 transmits to the data destination. P_{R_2} represents relay node R_2 packet acceptance probability, which is transmitted by the source,

$$P_{R_2} = r_2 f_{SR_2} \tag{29}$$

P_{E_2} also we calculate as,

$$P_{E_2} = r_2 f_{SR_2} \tag{30}$$

Probability of source energy queue is not empty is

$$P_r[E_{S_2} \neq 0] = q_S \tag{31}$$

The probability that the source transmission occupies channel 2,

$$\rho_{S_2} = \frac{1}{2} \tag{32}$$

SR_2 value is calculated as follows:

$$\lambda_{SR_2} = \lambda \frac{P_{R_2}}{P_{E_2}} \tag{33}$$

The probability that R_2 transmission occupied the channel,

$$\lambda_{R_2} = \frac{\lambda_{SR}}{f_{RD_2}} \tag{34}$$

S data queue service rate is

$$\mu_{S_2} = P_r[E_{S_2} \neq 0]P_{E_2} \tag{35}$$

Probability that energy queue of the relay node R_2 is not empty,

$$P_r[E_{R_2} \neq 0] = \frac{\min(q_{R_2}, \frac{1}{2} - \rho_{R_1})}{\frac{1}{2} - \rho_{R_1}} \tag{36}$$

If both the relay R_2 and source data queues are stable, the system is stable. If $\lambda_2 < \mu_{S_2}$ and $\lambda_{SR_2} < \mu_{R_2}$ both condition should be satisfied. So stability condition for is relay R_2,

$$\lambda_2 < \min\Big(\mu_S P_{E_2}, q_{R_2} f_{RD_2}, f_{RD_1} \frac{f_{RD_2}}{2(f_{RD_1} + f_{RD_2})}\Big) \tag{37}$$

For partial relaying optimization, r_2 optimal value is the solution of the problem,

$$r_2 < \min\Big(1, \max\big(\frac{q_{R_2} f_{RD_2}}{q_S f_{SR_2}}, \frac{f_{RD_1} f_{RD_2}}{2q_S f_{SR_2}(f_{RD_1} + f_{RD_2})}\big)\Big) \tag{38}$$

So, maximum achievable throughput of successive relaying system is

$$\begin{aligned} \lambda < \min\Big(\mu_S P_{E_1}, q_{R_1} f_{RD_1}, f_{RD_1} \frac{f_{RD_2}}{2(f_{RD_1} + f_{RD_2})}\Big) \\ + \min\Big(\mu_S P_{E_2}, q_{R_2} f_{RD_2}, f_{RD_1} \frac{f_{RD_2}}{2(f_{RD_1} + f_{RD_2})}\Big) \end{aligned} \tag{39}$$

Results

Represent the numerical results of the out theoretical concept discussion. Our aim is to showcase the influence of system parameters on system performance and the optimal value for the relaying parameter. We also investigate the effects of various system parameters on the source maximum stable throughput. We fix the channel success probabilities to be $f_{SR_1} = 0.4$, $f_{SR_2} = 0.3$, $f_{SD} = 0.2$, $f_{RD_1} = 0.4$, $f_{RD_2} = 0.3$, $q_{R_1} = 0.2$, and q_{R_2} =0.3. Figure 8.6 shows the graph between the maximum stable throughput vs the rate at which energy is harvested at the source. The graph illustrates

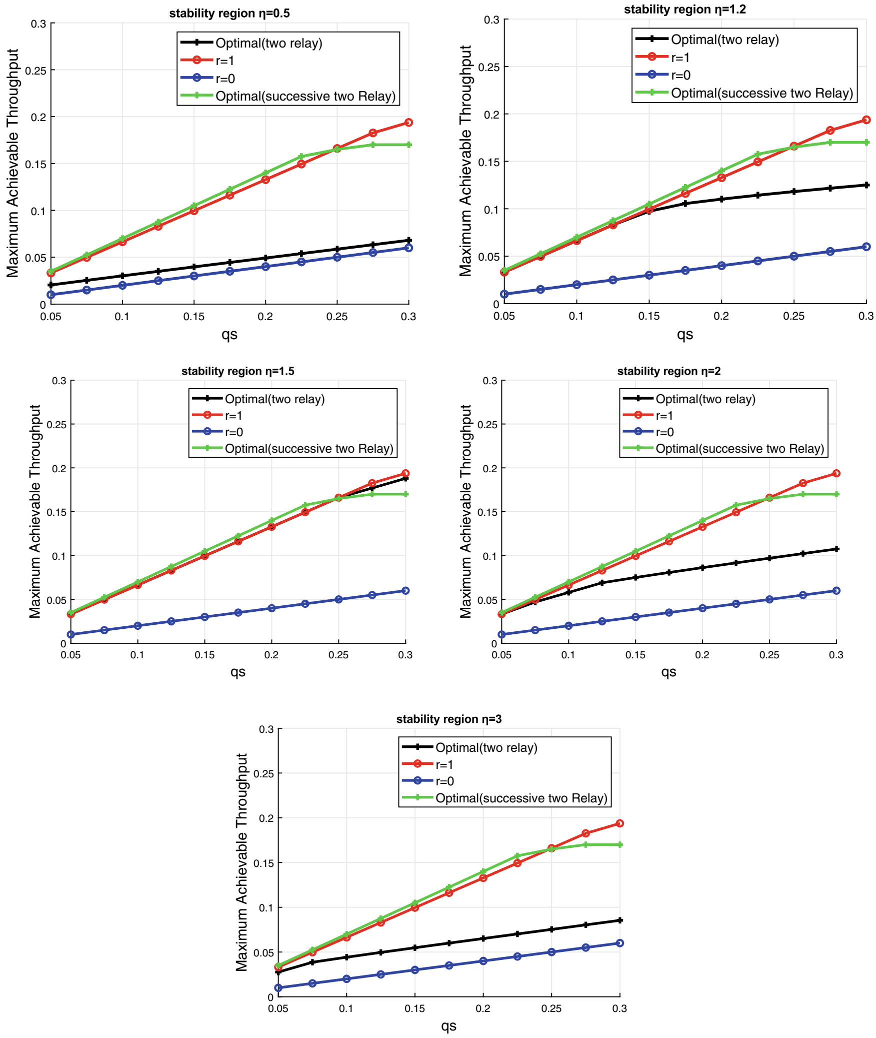


Fig. 8.6 Maximum achievable throughput against the q_S for $\eta = 0.5$, 1.2, 1.5, 2, and 3

how the optimal partial relaying parameters enhance the network's performance. When the source has sufficient energy to transfer its data to the destination, it sends the data directly. In the full relaying case, the source transmits the data through relay nodes, and in the case of successive relaying, data can be transmitted in successive fashion. Figure 8.6 show that if the difference between relaying parameters r_1 and r_2 increases, then the maximum achievable throughput decreases because data is sent through a lower index relay in the simple relay network. If the difference between both relay parameters is very large, it almost works as a single relay network. So, seeing the result in Figs. 8.3, 8.4, 8.5, 8.6, we conclude that maximum stable throughput increases when $\eta \leq 1.5$, and then after that, it decreases when $1.5 < \eta < \infty$.

Case 1 For $\eta = 0.5$, optimal successive two-relay network performs better in comparison to the simple or two-relay, single-relay, and no-relay sensor networks. Single-relay sensor networks perform always better compared to the two- and zero-relay networks. However, optimal successive two-relay network performs better for q_S < 2.5 and after that single-relay network dominant in over the all other networks.

Case 2 For $\eta = 1.2$, optimal simple two-relay maximal throughput is also increased with the η value increase.

Case 3 For $\eta = 1.5$, optimal or simple two-relay maximal throughput is greater than the optimal successive relay sensor networks for the $\eta = 1.5$ value.

Case 4 For $\eta = 2$ and $\eta = 3$, optimal or single two-relay maximal throughput is again decreased with an increase from $\eta = 1.5$.

Conclusion

For energy harvesting networks, we established the concept of partial network-level cooperation. The focus is on controlling the flow from source to relay, allowing for more effective resource utilization. We derive an exact characterization of the maximum achievable throughput in the considered system. The analysis demonstrates that cooperation enhances the maximum achievable rate of the source, resulting in improved overall network performance. Furthermore, the paper investigates the optimal relaying strategies based on data arrival rate at relay. For scenarios where the arrival data rate is small at the relay, employing full relaying is shown to be optimal. Conversely, when the data arrival rate at the source is small, it is optimal to refrain from using relaying altogether. The maximum achievable throughput for a successive relay network is determined and compared to that of a simple relay network. The analysis reveals that a successive relay network outperforms a simple relay network in terms of achievable throughput, suggesting the superiority of the former in certain scenarios. Finally, the resultant model provides security with lightweight communication cost and ensures confidentiality, integrity, and authentication. In summary, the paper presents the concept of partial network-level cooperation, analyzes the maximum achievable throughput, explores optimal relaying strategies based on data arrival rates, and demonstrates the advantages of a successive relay network over a simple relay network.

References

1. Kumar A, Singh J (2021) Optimization of substrate layer material and its mechanical properties for piezoelectric cantilever energy harvesting system. Adv Theory Simul 4(8):2100156
2. Kumar A, Jaiswal A, Joshi RS, Singh J (2022) A novel piezoelectric and electromagnetic energy harvester as a high-pass filter with a low cutoff frequency. IEEE Sens J 22(24):23705–23715
3. Kumar A, Singh J (2022) Interference aware heuristics to optimize power beacons for battery-less WSNS. In: Proceedings of the 25th international ACM conference on modeling analysis and simulation of wireless and mobile systems, pp 197–201
4. Oner O, Elza E (2015) Energy harvesting two-hop communication networks. IEEE J Select Areas Commun 33(12):2658–2670
5. Jeon J, Ephremides A (2011) The stability region of random multiple access under stochastic energy harvesting. In: 2011 IEEE international symposium on information theory proceedings. IEEE, pp 1796–1800
6. Paradiso JA, Starner T (2005) Energy scavenging for mobile and wireless electronics. IEEE Pervasive Comput 4(1):18–27
7. Wentzloff DD, Lee FS, Daly DC, Bhardwaj M, Mercier PP, Chandrakasan AP (2007) Energy efficient pulsed-uwb cmos circuits and systems. In: 2007 IEEE international conference on ultra-wideband. IEEE, pp 282–287
8. Raghunathan V, Kansal A, Hsu J, Friedman J, Srivastava M (2005) Design considerations for solar energy harvesting wireless embedded systems. In: IPSN 2005. Fourth international symposium on information processing in sensor networks 2005. IEEE, pp 457–462
9. Sahoo SS, Menon AR, Chaurasiya VK (2022) Secure blockchain model for vehicles toll collection by GPS tracking: a case study of India. In: 2022 IEEE India council international subsections conference (INDISCON). IEEE, pp 1–6
10. Jiang X, Polastre J, Culler D (2005) Perpetual environmentally powered sensor networks. In: IPSN 2005. Fourth international symposium on information processing in sensor networks. IEEE, pp 463–468
11. Gorlatova Maria, Kinget Peter, Kymissis Ioannis, Rubenstein Dan, Wang Xiaodong, Zussman Gil (2010) Energy harvesting active networked tags (enhants) for ubiquitous object networking. IEEE Wireless Commun 17(6):18–25
12. Sahoo SS, Hosmane MM, Menon AR, Chaurasiya VK (2022) Ethereum compatible faster atomic payment splitting network. In: 2022 IEEE 19th India council international conference (INDICON). IEEE, pp 1–6
13. Kasbekar GS, Bejerano Y, Sarkar S (2009) Lifetime and coverage guarantees through distributed coordinate-free sensor activation. In: Proceedings of the 15th annual international conference on Mobile computing and networking, pp 169–180
14. Himsoon T, Siriwongpairat WP, Han Z, Ray Liu KJ (2007) Lifetime maximization via cooperative nodes and relay deployment in wireless networks. IEEE J Select Areas Commun 25(2):306–317
15. Laneman JN, Wornell GW (2003) Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks. IEEE Trans Inform Theory 49(10):2415–2425
16. Laneman JN, Tse DNC, Wornell GW (2004) Cooperative diversity in wireless networks: efficient protocols and outage behavior. IEEE Trans Inform Theory 50(12):3062–3080
17. Sahoo SS, Chaurasiya VK (2023) Vibe: Blockchain-based virtual payment in IoT ecosystem: a secure decentralized marketplace. Multimedia Tools Appl 1–26

18. Jing Y, Jafarkhani H (2008) Single and multiple relay selection schemes and their diversity orders. In: ICC Workshops-2008 IEEE international conference on communications workshops. IEEE, pp 349–353
19. Sahoo SS, Menon AR, Chaurasiya VK (2023) Blockchain based n-party virtual payment model with concurrent execution. Arab J Sci Eng 1–28
20. Pappas N, Jeon J, Ephremides A, Traganitis A (2012) Wireless network-level partial relay cooperation. In: ISIT, pp 1122–1126
21. Sahoo SS, Hosmane MM, Chaurasiya VK (2023) A secure payment channel rebalancing model for layer-2 blockchain. Internet Things 22:100822
22. Jing L, Roy Y, Larry G (2009) A generic model for optimizing single-hop transmission policy of replenishable sensors. IEEE Trans Wireless Commun 8(2):547–551
23. Vinod S, Utpal M, Vinay J, Shrey G (2010) Optimal energy management policies for energy harvesting sensor nodes. IEEE Trans Wireless Commun 9(4):1326–1336
24. Ho CK, Zhang R (2010) Optimal energy allocation for wireless communications powered by energy harvesters. In: 2010 IEEE international symposium on information theory. IEEE, pp 2368–2372
25. Yang J, Ulukus S (2010) Transmission completion time minimization in an energy harvesting system. In: 2010 44th annual conference on information sciences and systems (CISS). IEEE, pp 1–6
26. Jing Y, Sennur U (2011) Optimal packet scheduling in an energy harvesting communication system. IEEE Trans Commun 60(1):220–230
27. Kaya T, Aylin Y (2012) Optimum transmission policies for battery limited energy harvesting nodes. IEEE Trans Wireless Commun 11(3):1180–1189
28. Omur O, Kaya T, Jing Y, Sennur U, Aylin Y (2011) Transmission with energy harvesting nodes in fading wireless channels: optimal policies. IEEE J Select Areas Commun 29(8):1732–1743
29. Mohamed K, Anthony E (2012) Optimal packet scheduling for energy harvesting sources on time varying wireless channels. J Commun Netw 14(2):121–129
30. Jing Y, Omur O, Sennur U (2011) Broadcasting with an energy harvesting rechargeable transmitter. IEEE Trans Wireless Commun 11(2):571–583
31. Antepli MA, Uysal-Biyikoglu E, Erkal H (2011) Optimal packet scheduling on an energy harvesting broadcast link. IEEE J Select Areas Commun 29(8):1721–1731
32. Sahoo SS, Chaurasiya VK (2023) Proof of location based delivery system using multi-party virtual state channel: a blockchain model. J Supercomput 1–31
33. Ozel O, Yang J, Ulukus S (2011) Broadcasting with a battery limited energy harvesting rechargeable transmitter. In: 2011 international symposium of modeling and optimization of mobile, Ad Hoc, and Wireless Networks. IEEE, pp 205–212
34. Loynes RM (1962) The stability of a queue with non-independent inter-arrival and service times. In: Mathematical proceedings of the cambridge philosophical society, vol 58. Cambridge University Press, pp 497–520
35. Mohanty S, Sahoo SS, Majhi B (2016) Certificateless nominative signature scheme based upon dlp. In: 2016 international conference on electrical, electronics, and optimization techniques (ICEEOT). IEEE, pp 1241–1245
36. Aditia MK, Altaf F, Singh MR, Burra MS, Maurya C, Sahoo SS, Maity S (2019) Optimized cl-pke with lightweight encryption for resource constrained devices. In: Proceedings of the 20th international conference on distributed computing and networking, pp 427–432

Chapter 9
MRMM-Based Keystream Generators for Information Security

Susil Kumar Bishoi and Vashek Matyas

Abstract The central theme of a stream cipher is a keystream generator that produces a pseudorandom bit sequence and is popularly used to ensure privacy over a communication channel. The word-based linear feedback shift register also known as the multiple recursive matrix method (MRMM) is attractive as it possesses most of the randomness properties like linear feedback shift register. It also takes advantage of modern word-based processors and thus increases the throughput. The major drawback of MRMM is that it has low linear complexity. In order to address the low linear complexity drawback in MRMM, the concept of several bit-oriented generators is extended to MRMMs. For this, a generalized form of the feedback function for word-oriented feedback shift registers is discussed, and then study MRMM as a special case.

Keywords Alternative step generators · Cascade generator · Linear complexity · LFG · Multiple-recursive matrix method · Shrinking generator · Stream cipher

Introduction

In the 21st century, information technology and electronic communication play a crucial role in our lives. They have become even more significant in the post-pandemic era due to their contactless nature. The volume of electronically exchanged and stored data is growing rapidly and is expected to continue its upward trend in the future. However, the increase in digital technologies also brings about a rise in online frauds, scams, intrusions, and security breaches. In such a scenario, the challenge is to ensure secure communication over insecure channels and secure storage of digital data.

S. K. Bishoi (✉)
CAIR, DRDO, Bangalore 560093, India
e-mail: skbishoi.cair@gov.in

V. Matyas
Masaryk University, Brno, Czech Republic
e-mail: matyas@fi.muni.cz

D. Gountia et al. (eds.), *Information Technology Security*, Springer Tracts in Electrical and Electronics Engineering, https://doi.org/10.1007/978-981-97-0407-1_9

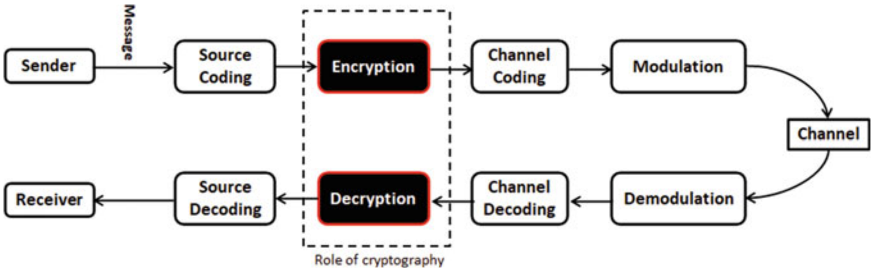


Fig. 9.1 Role of cryptography in a communication system

Cryptography [28, 37, 40] takes a leading role to solve these challenges, having a long and fascinating history, showcasing remarkable advancements in the art of secret communication. Earlier cryptography was mostly restricted to government agencies and the military. Now it has been extensively used by the telecommunication industry, the financial world, the healthcare sectors, the entertainment industry, etc.

Figure 9.1 illustrates the typical path of information flow in an insecure channel, comprising four main stages:

I. **Source Coding**: Redundancy is eliminated from the source, resulting in compressed information. Source encoders like ZIP and RAR are commonly used for this purpose. Source decoding reverses the compression to restore the original data.

II. **Encryption**: The compressed information undergoes encryption, transforming it into ciphertext, which is transmitted across the channel after suitable channel coding. Decryption is the reverse operation of the encryption.

III. **Channel Coding**: Redundancy is added to the data stream to facilitate error detection and correction during transmission. Error-correcting codes, employed in applications such as media compact discs and mobile communication, help ensure data integrity.

IV. **Modulation**: The modulation process converts the encoded information into a signal or light wave to transmit the information effectively through the channel, while demodulation is the process of extracting information from the transmitted signal.

Encryption can be used for secret storage or transmission of a message from one end to another such that any attacker or eavesdropper cannot understand the meaning of the message. Only intended people can read/understand after using the reverse process called decryption. So, encryption safeguards the personal security of millions of people and the national security of countries around the world. After encryption, the ciphertext is transmitted by the sender to the intended receiver over an insecure communication channel. At the receiver end, the plaintext is recovered using the decryption algorithm. The algorithm used for performing encryption and decryption is called the cryptosystem or cipher.

There are two types of cryptosystems given as follows:

I. Symmetric-key cryptosystems (also called secret-key cryptosystems).
II. Asymmetric-key cryptosystems (also known as public-key cryptosystems).

This chapter aims mostly at the area of type 1. For asymmetric-key cryptosystems, refer [28, 40].

Symmetric-Key Cryptosystems

Symmetric-key encryption is important for secure communication and storage. In a symmetric-key cryptosystem, the decryption key K_d is easily derivable from the encryption key K_e. For most of the ciphers, both K_e and K_d are the same key K [28]. Here also both K_e and K_d are considered to be the same. Both sender and receiver share the same secret key and the same symmetric-key cipher before the communication starts. Symmetric-key cryptosystems are commonly classified as being block cipher or stream cipher.

Block Cipher

In a block cipher, the input message M breaks into successive blocks, where each message block is of a fixed length known as block size (typically containing 64–256 bits), and suitable padding is used if the length of the last message block is less than the block size. Suppose the message $M = M_1 \| M_2 \ldots \| M_n$. Then the block cipher takes each message block M_i one by one along with the secret key and outputs a fixed-length ciphertext C_i for $1 \leq i \leq n$. That is, M is encrypted to $C = C_1 \| C_2 \ldots \| C_n$ where $C_1 = E_k(M_1), \ldots, C_n = E_k(M_n)$ and further C is transmitted across the channel to the receiver. At the receiver end, M is recovered using the decryption algorithm as $D_k(C_i) = M_i$ for $1 \leq i \leq n$. The literature on block cipher is extremely rich and one may refer to [28] for more details. Advanced Encryption Standard (AES) [29], DES [30], RC6 [35], SERPENT [2], TWOFISH [38] are some of the popular block ciphers.

Stream Cipher

Stream ciphers are simple block ciphers having block length equal to one. Prior to the eSTREAM project [13], several well-known stream ciphers, such as A5, E0, HELIX [15], FISH, RC4, SNOW [14], were in use. The eSTREAM project, initiated by ECRYPT, spanned multiple years to discover and evaluate new and promising stream ciphers. The project solicited proposals for new stream ciphers started in November 2004. These proposals were intended to satisfy either a hardware-oriented or a software-oriented profile or both. With an announcement of a portfolio of eight stream ciphers [13], four in each profile, the project ended in April 2008. The hardware-oriented cipher, F-FCSR-H v2, was eliminated from this portfolio in September 2008. The other seven ciphers were Grain v1, HC-128, MICKEY 2.0, Rabbit, Salsa20/12, SOSEMANUK, and Trivium.

Another competition that was announced at the Early Symmetric Crypto workshop in January 2013 is CAESAR [9]. This competition was organized by a group of international cryptologic researchers to encourage the design of authenticated encryption schemes. The final portfolio was announced in February 2019. The final CAESAR portfolio is organized into three use cases:

1. Lightweight applications (resource constrained environments).
2. High-performance applications.
3. Defense in depth.

In stream cipher, keystream generator is a crucial component. It generates a sequence of random or pseudorandom bits called the keystream, which is combined with the plaintext using a bitwise XOR operation to produce the ciphertext. If $k_1, k_2, \ldots, k_t$ are the keystream bits and $m_1, m_2, \ldots, m_t$ are the message bits, then the ciphertext bits $c_1, c_2, \ldots, c_t$ are computed as $c_i = m_i \bigoplus k_i$, $1 \leq i \leq t$, where t is the length of the message in bits. The classical example of a stream cipher is the Vernam cipher, also known as the one-time pad. It is theoretically secure if the key material is truly random, used only once, and at least as long as the plaintext [39]. Thus, both key distribution and key management are challenging in the case of one-time pad cipher. This motivates the design of stream ciphers based on the keystream generated by a pseudorandom bit generator. While pseudorandom bit generator-based stream ciphers are not unconditionally secure, their main goal is to achieve computational security. A pseudorandom bit generator is a deterministic algorithm that takes a smaller size random key K and produces a random-looking binary sequence of a large period. Therefore, the problem of designing a good stream cipher is to construct a good pseudorandom bit generator. Figure 9.2 illustrates the typical application of stream cipher in communication.

Stream ciphers can be classified as synchronous or asynchronous based on their synchronization requirements. Synchronous stream ciphers generate the keystream independently of the plaintext and ciphertext, while asynchronous stream ciphers rely on the previous ciphertext or plaintext to generate the keystream. Asynchronous stream ciphers often introduce a delay in the encryption process. This chapter discusses only the synchronous stream ciphers.

In the stream cipher design, feedback shift registers (FSRs, see [19, 23, 28, 40]) serve as one of the important basic building blocks. Both linear FSRs (LFSRs) and nonlinear FSRs (NLFSRs) [16, 19, 28, 37] are quite useful in the design of keystream generators. LFSRs are useful for their simplicity, efficiency, and low cost of hardware implementations. They are often used as the keystream generator in stream ciphers due to their ability to produce long sequences of pseudorandom bits having most of the statistical properties. But, the weakness of LFSRs is that they

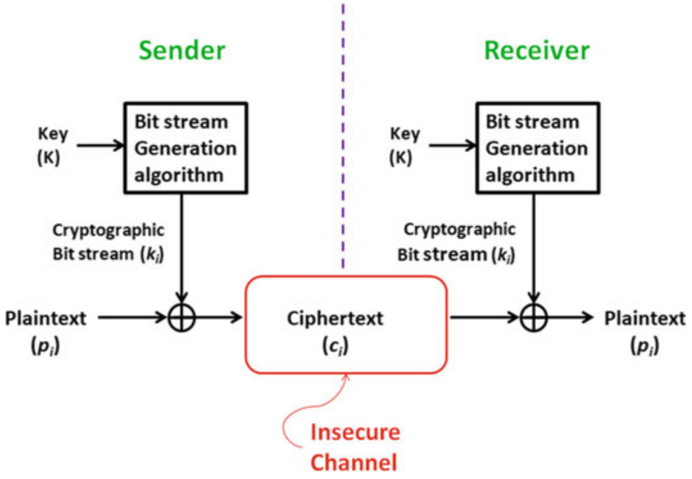

Sender
Receiver
Key (K)
Bit stream Generation algorithm
Cryptographic Bit stream (k_i)
Plaintext (p_i)
Ciphertext (c_i)
Key (K)
Bit stream Generation algorithm
Cryptographic Bit stream (k_i)
Plaintext (p_i)
Insecure Channel

Fig. 9.2 Stream cipher

have low linear complexity due to their linear structure. Due to the Berlekamp-Massey algorithm [26], it is possible to find the feedback polynomial of an n-stage LFSR and its initial states from any $2n$ consecutive output bits. Therefore, it is very much essential that the bitstream must have high linear complexity to counter this attack. The shortest length LFSR which reproduces the given bitstream is called its linear complexity. Several techniques are available in the literature to generate bitstreams with high linear complexity. Some of the well-known techniques are nonlinear combination generators, nonlinear filter generators, nonlinear feedback shift registers, and clock-controlled generators [28] for achieving the high linear complexity. In the case of nonlinear filter generators, only a single LFSR is used whereas several LFSRs are used in nonlinear combination generators. In both cases, the nonlinearity is introduced by a nonlinear Boolean function.

On the other hand, nonlinearity is introduced into the keystream by controlling the output of one LFSR by the other LFSRs, in the case of a clock-controlled generator [12, 27]. In [28, Sect. 6.3.3], two clock-controlled generators, i.e., the shrinking generator [12] and the alternating step generator [20] are described for keystream generation. Two LFSRs are used in the former case proposed by Coppersmith, Krawczyk, and Mansour whereas three LFSRs are used in the latter introduced by C.G. Gunther. In the case of the shrinking generator, both the LFSRs are always clocked together and the keystream bits are produced by shrinking the output sequence of one LFSR under the control of the second LFSR. If the control bit (i.e., the output bit of the first LFSR) is 1, then the output bit of the second LFSR is selected as a keystream bit, else it is discarded. In the case of an alternating step generator, the output bit of the first LFSR controls the clocking of the other two LFSRs. Another clock-controlled generator is the self-shrinking generator [27] proposed by Meier and Staffelbach. In this case, the keystream is generated only from a single LFSR. If the $(2k + 1)$th output bit of the LFSR is 0, then the $(2k+2)$th bit is discarded, otherwise the $(2k+2)$th output bit is added to the keystream bit for $k \geq 0$. It is shown in [27] that the self-shrinking generator can be realized as a shrinking generator and vice versa.

The above mentioned generators use only bit-oriented feedback shift registers (FSR). In the case of a bit-oriented FSR of order n, it needs n shifting and $O(n)$ operations for the feedback value computation in each cycle. But, the cost for shifting 1-bit is the same as that of shifting an m-bit word in the case of processors, where m is the machine processor size in bits which is 16 or 32, or 64 in most modern computer systems. The word-oriented primitives like lagged Fibonacci generators (LFGs) [11], word-based LFSRs called MRMMs [3, 4, 34, 41, 43], and Xorshift RNGs [5]) do take advantage of the available word-based processors. However, LFSRs, shrinking generators, and other bit-oriented pseudorandom number generators (PRNGs) do not exploit this benefit.

In the case of additive LFG (ALFG) of order n with word size m, it uses $O(n)$ modular additions of m-bit numbers to produce an m-bit output. But the complexity of one modular addition is $O(m^2)$ bit operations, thus ALFG needs $O(nm^2)$ bit operations to produce m-bit outputs. However, in the case of efficient MRMM, it takes $O(mn)$ bit operations to produce m-bit output. Thus, MRMM-based keystream generators are useful for resource-constraint environments.

In literature, MRMM is also known as σ-LFSR, and in the following text only MRMM is used. The primitive MRMMs produce bit or word sequences having most of the good statistical properties. Their period is exponential in terms of its order n and word size m. But they have low linear complexity similar to LFSR. Similar methods used in the case of bit-oriented LFSRs can be extended to MRMMs in order to have large linear complexity. Some of those ideas of the bit-oriented generator are extended to the respective word-oriented generator and investigate their periodicity and linear complexity along with their randomness properties.

Chapter Outline

The remainder of this section introduces basic notations. Word-oriented feedback shift registers, in particular MRMMs, are studied in Sect. 9.3. Several MRMM-based keystream generators are presented in Sect. 9.4. Section 9.5 concludes the chapter with references.

Notations

Consider the binary field, i.e., Galois field of two elements denoted by $\mathbb{F}_2 = \{0, 1\}$. Denote by $\mathbb{F}_2^m$ the m-dimensional vector space over $\mathbb{F}_2$. We use the normal letter x as a bit i.e., $x \in \mathbb{F}_2$, whereas the bold letter variable if it belongs to $\mathbb{F}_{2^m}$ i.e., $\boldsymbol{x} \in \mathbb{F}_{2^m}$. The symbol $+$ is the addition in the residue class ring $\mathbb{Z}/2^m\mathbb{Z}$ and the symbol $\oplus$ denotes the addition in $\mathbb{F}_2$. Denote by $\mathbb{M}_m(\mathbb{F}_2)$ the set of all $m \times m$ matrices with entries in $\mathbb{F}_2$ and the set of all $m \times m$ invertible matrices in $\mathbb{M}_m(\mathbb{F}_2)$ is denoted by $\mathbb{GL}_m(\mathbb{F}_2)$. I_m represents the $m \times m$ identity matrix. Denote by $\det(C)$ the determinant of a matrix $C \in \mathbb{M}_m(\mathbb{F}_2)$. The element of $\mathbb{F}_{2^m}$ may be thought of as a column vector of size m over $\mathbb{F}_2$ as two finite dimensional vector spaces $\mathbb{F}_{2^m}$ and $\mathbb{F}_2^m$ are isomorphic [31], and hence, for any $\mathbf{s} \in \mathbb{F}_{2^m}$ and $C \in \mathbb{M}_m(\mathbb{F}_2)$ the matrix-vector multiplication $C\mathbf{s}$ is a well-defined element of $\mathbb{F}_2^m$.

The set of all multivariate polynomial $F(.)$ in n variables $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n-1}$ with coefficients in $\mathbb{M}_m(\mathbb{F}_2)$ such that $F(\mathbf{0}, \ldots, \mathbf{0}) = \mathbf{0}$ and each $\boldsymbol{x}_i \in \mathbb{F}_{2^m}$ is denoted by $\mathcal{MP}_m[\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n-1}] = \mathbb{M}_m(\mathbb{F}_2)[\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n-1}]/(\boldsymbol{x}_0{}^2 \oplus \boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n-1}{}^2 \oplus \boldsymbol{x}_{n-1})$. If $F \in \mathcal{MP}_m[\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n-1}]$, then it can be expressed as follows.

$$F(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n-1}) = \sum_{I \in P(N)} C_I \prod_{k \in I} \boldsymbol{x}_k \tag{1}$$

where $C_I \in \mathbb{M}_m(\mathbb{F}_2)$ and $P(N)$ denotes the power set of $N = \{0, \ldots, n-1\}$. For example, if $N = \{0, 1\}$ then the general expression of $F(\boldsymbol{x}_0, \boldsymbol{x}_1) = C_0\boldsymbol{x}_0 + C_1\boldsymbol{x}_1 + C_{01}\boldsymbol{x}_0\boldsymbol{x}_1$. In Eq. (1), the term $\mathbf{X} = \prod_{k \in I} \boldsymbol{x}_k$ is calculated first, then $C_I \boldsymbol{X}$ is computed using matrix-vector multiplication. Here the sum $\sum$ is either modular integer addition + or bitwise XOR operation $\oplus$. Similarly, the product $\prod$ can be either field multiplication or modular integer multiplication $*$ or bitwise AND operation &. Two multiplication operations $*$ and & are only considered as the field multiplication is expensive compared to the other two. The algebraic degree of F denoted as $deg(F)$ can be defined as $\max\{\mid I \mid : C_I \neq 0\}$, where $\mid I \mid$ denotes the size of I.

Word-Oriented FSR

An n-stage word-oriented FSR (WFSR) is an FSR where each stage contains an m-bit size word. A state of a WFSR is a vector $(\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{n-1})$, where $\boldsymbol{y}_k$ indicates the content of stage k. Consider $F \in \mathcal{MP}_m[\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{n-1}]$ is the feedback function for WFSR and the initial state of WFSR is $\mathbf{S}_0 = (\mathbf{s}_0, \ldots, \mathbf{s}_{n-1})$. At every clock pulse, there is a transition from the state $\mathbf{S}_t = (\mathbf{s}_t, \ldots, \mathbf{s}_{t+n-1})$ to the state $\mathbf{S}_{t+1} = (\mathbf{s}_{t+1}, \ldots, \mathbf{s}_{t+n})$, where $\mathbf{s}_{n+t} = F(\mathbf{S}_t)$ for integer $t \geq 0$. The WFSR outputs a word sequence $[\mathbf{s}] = \{\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_n, \ldots\}$ after consecutive clock pulses. If there exists integers r, n_0 with $r \geq 1$ and $n_0 \geq 0$ such that $\mathbf{s}_{j+r} = \mathbf{s}_j$ for all $j \geq n_0$, then the sequence $[\mathbf{s}]$ is said to be *ultimately periodic*. The sequence $[\mathbf{s}]$ is said to be periodic, if $n_0 = 0$ and in such case, the least positive integer r is called the *period* of the sequence $[\mathbf{s}]$. If $deg(F) = 1$, then WFSR is called a linear WFSR otherwise a nonlinear WFSR.

Example 1 Consider WFSR$_1$ as a three-stage WFSR with word size 2 and its feedback function as $F_1(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2) = C_{12}\boldsymbol{x}_1\boldsymbol{x}_2 + C_0\boldsymbol{x}_0$ where $C_{12} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and

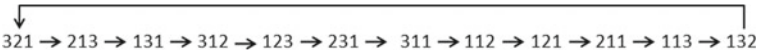


Fig. 9.3 The state diagram of WFSR$_1$ with initial state 321

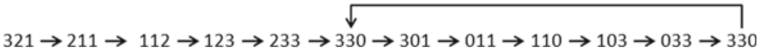


Fig. 9.4 The state diagram of WFSR$_2$ with initial state 321

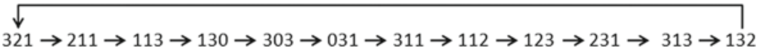


Fig. 9.5 The state diagram of WFSR$_3$ with initial state 321

$C_0 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$. Here modular integer multiplication and modular integer addition are used in the feedback value calculation and $deg(F) = 2$. One can compute the next state of a given state by using the feedback function. The number of all possible states for a 3-stage WFSR with word size 2 is 4^3 states. A part of the state diagram of WFSR$_1$ is shown in Fig. 9.3 with initial state $(3, 2, 1)$. For abuse of notation, we use 321 for the state $(3, 2, 1)$. Each number is converted to a vector during the feedback value computation. For this example, $0 = [0, 0]^t$, $1 = [0, 1]^t$, $2 = [1, 0]^t$ and $3 = [1, 1]^t$. As $F(3, 2, 1) = 3$, the next state of 321 is 213. In this case, the state 321 is called the predecessor of 213 and the state 213 is called the successor of 321.

Consider another three-stage WFSR with word size 2 called WFSR$_2$. The feedback function for WFSR$_2$ is $F_2(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2) = C_0\boldsymbol{x}_0 + C_{12}\boldsymbol{x}_0\boldsymbol{x}_1\boldsymbol{x}_2$ where the coefficient matrices C_{12} and C_0 are same as in WFSR$_1$. The degree of the feedback function for WFSR$_1$ is 3. The part of the state diagram of WFSR$_2$ is given in Fig. 9.4. The same initial state 321 is used in WFSR$_2$. Let WFSR$_3$ be a three-stage WFSR with word size 2 and the feedback function $F_3(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2) = C_{12}(\boldsymbol{x}_1 \,\&\, \boldsymbol{x}_2) + C_0\boldsymbol{x}_0$. Note that the feedback function F is the same as in WFSR$_1$ except for a different operation. Here, the bitwise AND operation & is used in place of the modular integer multiplication $*$. The part of the state diagram of WFSR$_3$ with the same initial state 321 is shown in Fig. 9.5.

In Fig. 9.4, it is observed that distinct vectors do not have distinct successors in the case of WFSR$_2$. The state 330 is the common successor of 233 and 330 which means 330 does not have unique predecessors. If the distinct state in a WFSR has a distinct predecessor, then the WFSR (or its feedback function) is called nonsingular. It is shown in [19] that the bit-oriented FSR is nonsingular if and only if its feedback function is of type

$$f(x_0, \ldots, x_{n-1}) = x_0 \oplus g(x_1, \ldots, x_{n-1}) \tag{2}$$

where the function $g(.)$ does not depend on the variable x_0. Now a natural question arises: Is there a single condition that guarantees both necessity and sufficiency for

WFSR to be nonsingular? If the feedback function of a WFSR is of the form in Eq. (2), then it is straightforward that the WFSR is nonsingular. However, the feedback function of WFSR$_1$ of Example 1 has a different form. In the following theorem, a necessary and sufficient condition for WFSR to be nonsingular is provided.

Theorem 1 *An n-stage WFSR with word size m is nonsingular if and only if its feedback function* $F \in \mathcal{MP}_m[\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n-1}]$ *can be represented as*

$$F(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n-1}) = f_0(\boldsymbol{x}_0) \odot f_1(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}) \tag{3}$$

where f_1 *is an arbitrary function from* $\mathbb{F}_{2^m}^{n-1} \rightarrow \mathbb{F}_{2^m}$ *and* f_0 *is a bijective function. The operation* $\odot$ *is either* $+$ *or* $\oplus$*.*

Proof Suppose WFSR is nonsingular, then distinct vectors have distinct successors. If the feedback function F does not contain any $\boldsymbol{x}_0$ term, then it is straightforward to get a common successor for two distinct vectors. Hence, the term $\boldsymbol{x}_0$ must present in the expression of F. Thus, $F(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n-1})$ can be viewed as $f_0(\boldsymbol{x}_0) \odot \boldsymbol{x}_0 f_2(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}) \odot f_1(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1})$ where $f_0(\boldsymbol{x}_0)$ contains all terms of $\boldsymbol{x}_0$ and the variable $\boldsymbol{x}_0$ is absent in both the functions f_1 and f_2. This implies that f_2 does not have any constant term and so $f_2(0, \ldots, 0) = 0$. Assume that f_0 is not bijective, then there exists $\boldsymbol{x}_0^{'} \neq \boldsymbol{x}_0^{''}$, such that $F(\boldsymbol{x}_0^{'}, 0, \ldots, 0) = F(\boldsymbol{x}_0^{''}, 0, \ldots, 0)$. Therefore, there are distinct vectors having a common successor. This is a contradiction and hence f_0 is bijective. Now the next aim is to show that $f_2 = 0$. Assume $f_2 \neq 0$, then $f_2(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1})$ must contain a term of $\boldsymbol{x}_k$ for some $k \neq 0$. In this case, $F(0, 2, \ldots, 2) = F(2^{m-1}, 2, \ldots, 2)$. This is a contradiction as WFSR is nonsingular and thus $f_2(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}) = 0$. This proves the necessary part.

A WFSR is nonsingular if and only if distinct vectors have distinct successors. If $(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1})$ and $(\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n-1})$ differ in any component other than the first, then their successors $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$ and $(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n)$ will be distinct. Therefore, the necessary and sufficient condition for WFSR to be nonsingular is that $(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1})$ and $(\boldsymbol{y}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1})$ have distinct successors for $\boldsymbol{x}_0 \neq \boldsymbol{y}_0$. Assume WFSR is singular, then there exists $\boldsymbol{x}_0 \neq \boldsymbol{y}_0$ such that $f(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n-1}) = f(\boldsymbol{y}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1})$. This implies that $f_0(\mathbf{x}_0) = f_0(\mathbf{y}_0)$ for $\boldsymbol{x}_0 \neq \boldsymbol{y}_0$. This is not possible as f_0 is bijective. This completes the proof. □

Corollary 1 *Bit-oriented FSRs are nonsingular if and only if the feedback function* $f(x_0, \ldots, x_{n-1}) = x_0 \oplus g(x_1, \ldots, x_{n-1})$*.*

By putting different restrictions in Eq. (1), three well-known FSRs are described as the special case of WFSR in the following.

Bit-Oriented FSRs

The bit-oriented FSRs are the special case of WFSR. In this case, $m = 1$ and the $m \times m$ matrix coefficient C_k of Eq. (1) becomes a scalar $c_k \in \mathbb{F}_2$ and the feedback function $F(x_0, \dots, x_{n-1}) = \bigoplus_{k=0}^{n-1} c_k x_k$ when $deg(F) = 1$. This expression is the feedback function of a well-developed bit-oriented FSR called LFSR [19]. LFSR generates maximal periodic bitstreams for any nonzero initialization and these bitstreams satisfy most of the statistical properties when it is primitive.

The bit-oriented FSR becomes NLFSR when $deg(F) > 1$. There is no efficient way to generate bitstream by an NLFSR with guaranteed large periods. Also, it is hard to determine the periods of NFSR sequences. However, Golomb [19] proved that the sequences produced by an NFSR are periodic if and only if the feedback function of NLFSR is nonsingular. Note that if $deg(F) = 1$, then the feedback function is always nonsingular as the feedback function F in Eq. (1) satisfies the criteria of Theorem 1. Though, it is hard to find NLFSR with a large period, still there are plenty of n-stage NLFSRs having the maximum period 2^n, and in this case, they are known as de Bruijn sequences. Unlike LFSRs, the theory of NFSRs is not well-investigated due to its complexity.

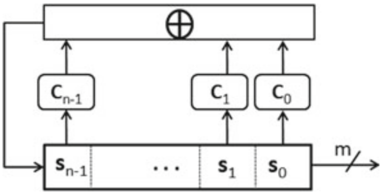
LFG

Another special case of WFSR is LFG. If $deg(F)$ is 1 with $m > 1$ and $+$ is used in place of $\odot$, then $F(\boldsymbol{x}_0, \dots, \boldsymbol{x}_{n-1}) = C_0\boldsymbol{x}_0 + \dots + C_{n-1}\boldsymbol{x}_{n-1}$. If all $C_k \in \{I_m, 0\}$, then $F(\boldsymbol{x}_0, \dots, \boldsymbol{x}_{n-1})$ is the general expression of the feedback function of an additive LFG. Using matrix theory, George Marsaglia et al. [25] prove that for the transition matrix A corresponding to the characteristic primitive polynomial, the recurrence relation has the maximal period $(2^n - 1)2^{m-1}$ for all $m \geq 1$ for every initial state with at least one odd number if and only if the order of A satisfies the following conditions

- order $j = 2^n - 1$ in the group of nonsingular matrices for mod 2,
- order $2j$ for mod 2^2,
- order $4j$ for mod 2^3.

If $f(x) = x^n - c_{n-1}x^{n-1} - \cdots - c_1 x - c_0$ is the corresponding characteristic primitive polynomial of LFG, then it is proved by Brent [8] that the period of the recurrence relation is $2^{m-1}(2^n - 1)$ for all $m \geq 1$ if and only if $f(x)^2 + f(-x)^2 \not\equiv 2f(x^2) \pmod 8$ and $f(x)^2 + f(-x)^2 \not\equiv 2(-1)^n f(-x^2) \pmod 8$. An additive LFG is called primitive if it attains the maximal period $(2^n - 1)2^{m-1}$. It is shown in [11] that an nth order LFG can be visualized as a scrambler of m binary LFSRs. In this case, each LFSR is of length n has the same feedback function, and for $k = 1, \dots, m$; the kth LFSR corresponds to the kth least significant bit (lsb) of each of the m-bit state of the LFG. Except 1st lsb LFSR, all other $(m - 1)$ LFSRs run in scrambler mode, i.e., feedback value is modified by an external bit value.

Fig. 9.6 The nth order MRMM over $\mathbb{F}_{2^m}$



MRMM

The next special case of WFSR is MRMM as depicted in Fig. 9.6. In an n-stage WFSR with word size m, if $deg(F)$ is 1, $m > 1$ and $\oplus$ is used in place of $\odot$, then the expression of F in Eq. (1) becomes $F(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n-1}) = \bigoplus_{k=0}^{n-1} C_k \boldsymbol{x}_k$ where $C_0, C_1, \ldots, C_{n-1} \in \mathbb{M}_m(\mathbb{F}_2)$. This is the general expression for the feedback function of MRMM [31–33]. An n-stage MRMM over $\mathbb{F}_{2^m}$ is denoted by MRMM(m, n). It is always possible for a periodic word sequence $[\mathbf{s}]$ to have a linear recurring relation (LRR) among the elements as

$$\mathbf{s}_{i+n} = \sum_{k=0}^{n-1} C_k \mathbf{s}_{i+k} \quad i \geq 0. \tag{4}$$

It is possible to associate Eq. (4) with a matrix polynomial. It is expressed as $M(x) = I_m x^n - C_{n-1} x^{n-1} - \cdots - C_1 x - C_0$ with matrix coefficients and is called the *matrix polynomial* of the MRMM(m, n). Then, det$(M(x))$ is a polynomial of degree mn over $\mathbb{F}_2$. Again, it is always possible to associate an (m, n)-block companion matrix $T \in M_{mn}(\mathbb{F}_2)$ for $M(x) \in \mathbb{M}_m(\mathbb{F}_2)[x]$ in the following form

$$T = \begin{bmatrix} \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & C_0 \\ I_m & \ldots & \mathbf{0} & \mathbf{0} & C_1 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \ldots & I_m & \mathbf{0} & C_{n-2} \\ \mathbf{0} & \ldots & \mathbf{0} & I_m & C_{n-1} \end{bmatrix}. \tag{5}$$

where $\mathbf{0}$ indicates the zero matrix in $\mathbb{M}_m(\mathbb{F}_2)$, while I_m denotes the $m \times m$ identity matrix over $\mathbb{F}_2$. It is easy to see that the characteristic polynomial of the MRMM is same as the characteristic polynomial of T = det$(T - xI)$.

The following proposition [17, Proposition 4.2] gives some basic facts about the MRMMs.

Proposition 1 *For the sequence* $[\mathbf{s}]$ *generated by the MRMM(m, n),*

(i) $[\mathbf{s}]$ *is ultimately periodic and its period is no more than* $2^{mn} - 1$*;*

(ii) *if C_0 is nonsingular, then* $[\mathbf{s}]$ *is periodic. Conversely, if* $[\mathbf{s}]$ *is periodic whenever the initial state is of the form* $(b, 0, \dots, 0)$*, where* $b \in \mathbb{F}_{2^m}$ *with* $b \neq 0$*, then* C_0 *is nonsingular.*

Let the word sequence $[\mathbf{s}]$ be generated by a primitive MRMM(m, n). For $j = 1, 2, \dots, m$, consider $[s^{(j)}] = \{s_0^{(j)}, s_1^{(j)}, \dots, \}$ be the bit sequence, where $s_k^{(j)}$ is the jth least significant bit (lsb) of m-bit word $\mathbf{s}_k$. Then, $\mathbf{s}_k = (s_k^{(m)}, \dots, s_k^{(2)}, s_k^{(1)}) \in \mathbb{F}_2^m$, for $k = 0, 1, \dots$, and each $s_k{}^{(j)} \in \mathbb{F}_2$. Now, the following result is due to Niederreiter [31, Lemma-1].

Lemma 1 *Let* $[\mathbf{s}]$ *be an arbitrary recursive vector sequence with the characteristic polynomial* $g(x) \in \mathbb{F}_2[x]$ *of degree mn and each vector is m-bit wide. Then, the bit sequence* $[s^{(j)}]$ *for* $1 \le j \le m$ *will be a linear recurring sequence in* $\mathbb{F}_2$ *with the same characteristic polynomial g(x). So if* $g(x)$ *is primitive, then the word sequence* $[s]$ *and each bit sequence* $[s^{(j)}]$ *attain the maximal period i.e.,* $(2^{mn} - 1)$.

Each bit sequence $[s^{(j)}]$ has period $(2^{mn} - 1)$ in the case of a primitive MRMM(m, n) and is a circularly shifted version of any other bit sequence $[s^{(k)}]$ for $1 \le j, k \le m$.

The following corollary follows from Lemma 1 and gives the component-wise linear complexity of the sequences generated by primitive MRMM.

Corollary 2 *Let*

$$\mathbf{s}_i = \left(s_i^{(m)}, \dots, s_i^{(1)}\right) \in \mathbb{F}_2^m \simeq \mathbb{F}_{2^m} \quad i = 0, 1, \dots,$$

be a sequence of words generated by a primitive MRMM of order n over $\mathbb{F}_{2^m}$*. Then for each* $1 \le j \le m$*, the linear complexity of the* j^{th} *coordinate sequence* $[s^{(j)}] = \{s_0^{(j)}, s_1^{(j)}, \dots, \}$ *over* $\mathbb{F}_2$ *is mn.*

Algorithms for Efficient Primitive MRMM

From Eq. (4), it is obvious that the MRMM needs n state shifting operations along with a feedback value computation to produce an m-bit word. Thus, the feedback value calculation needs several matrix-vector multiplications and XOR operations. The computational complexity for matrix-vector multiplication of order m is $O(m^2)$. However, in the search algorithm [43] and the construction algorithm [3] for the generation of primitive MRMMs, a special kind of matrices are used. For those special matrices, the matrix-vector multiplication complexity is $O(m)$ instead of $O(m^2)$ and thus the MRMMs generated by both the search algorithm and construction algorithm are efficient. The search algorithm for efficient MRMM is presented below.

Search algorithm: The search algorithm [43, Algorithm 1] proposed by Zeng et al. produces primitive MRMMs and they have a very efficient and fast software implementation. Some special word-oriented linear transformations provided by modern processors are used in those MRMMs. The list of those operations is bitwise AND, OR, and XOR operations, left (right) shift operation, circular rotation, and a combination of left shift and right shift operations.

1. AND operation Λ_γ: Let $\{\alpha_i\}_{i=0}^{m-1}$ be the basis for $\mathbb{F}_{2^m}$, then each $\gamma \in \mathbb{F}_{2^m}$ can be expressed as $\gamma = \sum_{i=0}^{m-1} c_i\alpha_i$, where $c_i \in \mathbb{F}_2$. Then, for each $x = (x_0, x_1, \ldots, x_{m-1}) \in \mathbb{F}_{2^m}$, Λ_γ is defined as $\Lambda_\gamma(x) = \sum_{i=0}^{m-1} c_i\alpha_i x_i$. The matrix representation of Λ_γ is given below.
$$\Lambda_\gamma = \begin{pmatrix} c_0 & 0 & \cdots & 0 & 0 \\ 0 & c_1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & c_{m-1} \end{pmatrix}_{m\times m}$$

2. Left shift operation $\mathbf{L}$ and right shift operation $\mathbf{R}$: For $\boldsymbol{x} = (x_0, \ldots, x_{m-1})$, the left shift operator $\mathbf{L}$ is defined as $\mathbf{L}(\boldsymbol{x}) = (x_1, \ldots, x_{m-1}, 0)$ and the matrix representation of the operator $\mathbf{L}$ is as follows
$$\mathbf{L} = \begin{pmatrix} 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix}_{m\times m}$$
Similarly, the right shift operator $\mathbf{R}$ is defined as $\mathbf{R}(\boldsymbol{x}) = (0, x_0, \ldots, x_{m-2})$ and the matrix representation of $\mathbf{R}$ is the transpose of the matrix $\mathbf{L}$, i.e.,
$$\mathbf{R} = \mathbf{L}^T = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}_{m\times m}$$

3. Circular rotation operation σ and β: The right circular rotation σ is defined as $\sigma(x) = \sigma(x_0, \ldots, x_{m-1}) = (x_{m-1}, x_0, \ldots, x_{m-2})$. The matrix representation of σ is as follows
$$\sigma = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}_{m\times m}$$

Similarly, the left circular rotation β is defined as the transpose of σ , i.e., $\beta(x) = \beta(x_0, \dots, x_{m-1}) = (x_1, \dots, x_{m-1}, x_0)$ and the matrix form of β is

$$\beta = \sigma^T = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{pmatrix}_{m \times m}$$

It can be verified that $\sigma\beta = \beta\sigma = I_m$.

4. Combination of left shift and right shift operation $\bigsqcup_{s,t}$: For given positive integers $0 < s, t < m$, the operator $\bigsqcup_{s,t} = \mathbf{L}_s + \mathbf{R}_t$, i.e., it is an $m \times m$ matrix having all entries are 1 in sth upper sub-diagonal and tth lower sub-diagonal. For example

$$\bigsqcup_{m-2,1} = \begin{pmatrix} 0 & 0 & \cdots & 1 & 0 \\ 1 & 0 & \cdots & 0 & 1 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}_{m \times m}$$

Now, define two sets W and V as follows by using the above operators followed by the search algorithm [43].

$W = \{\Lambda_\gamma, \mathbf{L}^j, \mathbf{R}^k, \sqcup_{s,t} \mid \gamma \in \mathbb{F}_{2^m}, \gamma \neq 0 \text{ or } 2^m - 1; 0 < j, k, s, t < m\}$
$V = \{\sigma^k, \sqcup_{s,t} \mid -m < k < m; 0 < s, t < m, (s + t)|m\}$

Algorithm 1 Search algorithm for primitive MRMM

```
Require: Order of the MRMM $n$ and the word size $m$.
Ensure: An efficient primitive MRMM($m, n$).
1: Choose $C_0 \in V$ and $C_1, C_2, \dots, C_{n-1} \in V \bigcup W$ randomly and compute the matrix polynomial
   $M(x) = I_m X^n + \cdots + C_1 X + C_0$.
2: Compute the determinant $g(x) = det(M(x))$.
3: $M(x)$ returns a primitive MRMM if $g(x)$ is primitive, otherwise go to Step 1.
```

The MRMMs obtained from the search algorithm have an efficient implementation in both software and hardware. Yet the search algorithm becomes sluggish for larger values of m and n as the search space size increases exponentially [5].

Construction of efficient primitive MRMM: The search algorithm [43, Algorithm 1] for generating efficient primitive MRMM first constructs the matrix polynomial by choosing some matrices randomly from a specific set. Then test the primitivity of a polynomial obtained by computing the determinant of a matrix polynomial. The idea behind the search algorithm is, that it first constructs efficient MRMM and then checks for its primitiveness. Can the reverse method is possible, i.e., from a given primitive polynomial, can an efficient MRMM be constructed? Yes, it is possible, An algorithm called the construction algorithm for primitive MRMM is given for the same [3]. This algorithm constructs a primitive MRMM(m, n) from a primitive polynomial $f(x)$ of degree mn. The sequential steps of this algorithm are described in [3]. Algorithm 2 describes a brief description of this algorithm.

Algorithm 2 Construction of efficient primitive MRMM(m, n)

```
Require: Two positive integers $m$ and $n$.
Ensure: An efficient primitive MRMM$(m, n)$.
1: Generates a primitive polynomial $f(x)$ of degree $mn$.
2: Compute $(m, n)$-Horner's form of $f(x)$ (as explained in [3]).
3: Construct $(m, n)$-Horner's matrix $H_m(n, f)$ of $f(X)$ and generates the matrix coefficients $C_i$.
```

The class of xorshift random number generators (RNGs) [24] are RNGs that use only XOR and shift operations. The MRMMs constructed by the construction algorithm use only XOR and shift operations for feedback computations, thus they fall in the class of xorshift RNGs.

It is shown in [5] that the initial states of the MRMMs generated by the construction algorithm need to be initialized carefully for the generation of good pseudorandom numbers. To avoid this weakness, it is suggested that the initial states of such MRMMs be initialized with odd numbers.

MRMM-Based Keystream Generators

MRMMs produce sequences having large periods, good randomness properties and are well-suited for both hardware and software implementation. They are readily analyzed using algebraic techniques. Again, a bit sequence generated by an LFSR of order mn can be reproduced by an MRMM(m, n). Thus, MRMMs can be used like LFSRs in the design of keystream generators. Unfortunately, the output sequences of MRMMs are also easily predictable like in LFSR due to their linear structure. Though the period of a primitive MRMM(m, n) is $(2^{mn} - 1)$, its linear complexity (LC) is mn (by Corollary 2). Then the natural question comes to mind: how to increase the LC in the case of MRMM-based keystream generators. A method using the Langford arrangement is proposed in [3], to achieve a higher LC with complexity $O((mn)^2)$. This subsection summarizes some of the techniques for destroying the linearity properties of MRMMs and the results pertaining to them.

Word-Oriented Nonlinear Filter Generators

In the case of LFSRs, one technique for removing the linearity is the use of a nonlinear filter generator. This nonlinear filter generator [36] consists of a single LFSR which is filtered by a nonlinear function. The keystream is the output of a nonlinear function which takes some stages of a single LFSR as inputs.

Fact 2 describes the linear complexity of the output sequence of an LFSR-based nonlinear filter generator [28, Fact 6.55].

Fact 2 *Suppose that a nonlinear filter generator is constructed using a primitive LFSR of length n and a filtering function f of nonlinear order m.*

1. *The upper bound for linear complexity of the keystream is $n_m = \sum_{i=1}^{m} \binom{n}{i}$.*
2. *For large n, most of the generators produce sequences whose linear complexity meets the upper bound n_m.*

Therefore, the algebraic degree of the filtering function with an n-stage LFSR should be large enough to achieve a high linear complexity. Again, in order to obtain a keystream sequence with good statistical properties [1], the LFSR should be primitive and the filtering function should be balanced (i.e., output bits are uniformly distributed). The concept of the bit-oriented filter generator is extended to the word-oriented filter generator in [21].

Shrinking MRMMs

The bit-oriented shrinking generator was introduced by Coppersmith et al. [12]. The following result is due to [12, Theorem 1].

Fact 3 *Let A and S form a shrinking generator Z. Denote by T_A, T_S, the periods of the A- and S-sequences, respectively. If*

- *A and S are of maximal length*
- *$gcd(T_A, T_S) = 1$*

then Z has period $T_A \cdot 2^{|S|-1} = (2^{|A|} - 1) \cdot 2^{|S|-1}$.

The bit-oriented shrinking generator concept is extended to that of a word-oriented shrinking generator in [6]. Let $[\mathbf{a}] = \{\mathbf{a}_0, \mathbf{a}_1, \ldots\}$ and $[\mathbf{s}] = \{\mathbf{s}_0, \mathbf{s}_1, \ldots\}$ be two word sequences. A new word sequence $[\mathbf{z}] = \{\mathbf{z}_0, \mathbf{z}_1, \ldots\}$ is generated by using the sequences $[\mathbf{a}]$ and $[\mathbf{s}]$ as follows. For $i = 0, 1, \ldots, \mathbf{a}_i$ is added into the sequence $[\mathbf{z}]$ if $\mathbf{s}_i$ is odd, otherwise discard $\mathbf{a}_i$. Thus, $\mathbf{z}_i = \mathbf{a}_{i_k}$, where i_k is the position of the kth odd number in the sequence $[\mathbf{s}]$ and hence the resultant word sequence $[\mathbf{z}]$ is a shrunken version of the sequence $[\mathbf{a}]$. Let $\text{MRMM}_1(m_1, n_1)$ and $\text{MRMM}_2(m_2, n_2)$

be two primitive MRMMs such that the nonzero word sequences [**a**] and [**s**] are generated by MRMM$_1$ and MRMM$_2$, respectively. Then, $T_A = 2^{m_1 n_1} - 1$ is the period of [**a**], whereas the period of [**s**] is $T_S = 2^{m_2 n_2} - 1$.

The following result due to [6, Theorem 3.3] tells about the periodicity and linear complexity of [**z**].

Fact 4 *Let* $[\boldsymbol{a}]$ *and* $[\boldsymbol{s}]$ *be two word sequences generated by two primitive MRMMs and form a shrinking generator* $[\boldsymbol{z}]$ *as defined above. If* T_A *and* T_S *be the period of* $[\boldsymbol{a}]$ *and* $[\boldsymbol{s}]$*, respectively, then the shrunken word sequence* $[\boldsymbol{z}]$ *has period* $\frac{T_A}{gcd(T_A,T_S)} \cdot 2^{m_2 n_2 - 1}$.

The word size m of most of the computer systems is of the form 2^k for some positive integer k. Assume $m_1 = 2^{k_1}$ and $m_2 = 2^{k_2}$ for some positive integers k_1 and k_2. Then, the following fact provides both the bounds for linear complexity of [**z**].

Fact 5 *Under the conditions of Fact 4 and if* $gcd(T_A, T_S) = 1$*, then the LC of* $[\boldsymbol{z}]$ *satisfies the following inequality,*

$$m_1{}^2 n_1 2^{m_2 n_2 - 2} < LC \leq m_1{}^2 n_1 2^{m_2 n_2 - 1}$$

Self-shrinking MRMMs

Another MRMM-based keystream generator with large linear complexity is a word-oriented self-shrinking generator [6]. This is similar to the bit-oriented self-shrinking generator (SSG) [27]. In case of bit-oriented n-stage SSG, the period $P \geq 2^{\lfloor \frac{N}{2} \rfloor}$ and P divides 2^{n-1} [27]. Later it is shown in [6] that the period is exactly equal to 2^{n-1}.

More formally, let $[\mathbf{s}] = \{\mathbf{s}_0, \mathbf{s}_1, \ldots\}$ be the word sequence with period P. Then, a new word sequence $[\mathbf{z}] = \{\mathbf{z}_0, \mathbf{z}_1, \ldots\}$ is generated from [**s**] using similar technique used in bit-oriented SSG. For integer $i \geq 0$, if $\mathbf{s}_{2i}$ is an odd number, then $\mathbf{s}_{2i+1}$ is added into the word sequence [**z**] otherwise $\mathbf{s}_{2i+1}$ is discarded. As MRMM with a large period is interested, only primitive MRMMs are considered. The following result is due to [6, Theorem 4.2].

Fact 6 *The word sequence produced by a self-shrinking primitive MRMM*(m, n) *is periodic and its period divides* 2^{mn-1}*. Moreover, the generated shrunken sequence is balanced.*

Let $[z^{(j)}] = \{z_0^{(j)}, z_1^{(j)}, \ldots, \}$ be the jth bit sequence, where $z_k^{(j)}$ is the jth bit of m-bit word $\mathbf{z}_k$. So, $\mathbf{z}_k = (z_k^{(m)}, \ldots, z_k^{(1)}) \in \mathbb{F}_2^m$. Then, the following facts give a lower bound for the period and linear complexity of [**z**].

Fact 7 *In case of a primitive MRMM*(m, n)*, the period* P *of* $[z^{(j)}]$ *satisfies*

$$P \geq 2^{m \lfloor \frac{n}{2} \rfloor}. \tag{6}$$

Fact 8 *The linear complexity LC of the bit sequence produced by a self-shrunken maximum length MRMM(m, n) with $m = 2^k$ satisfies*

$$LC > 2^{m\lfloor \frac{n}{2} \rfloor + k - 1}. \tag{7}$$

Cascade Connection of WFSRs

The cascade connection of two FSRs was first introduced in [18]. Consider f_1 and f_2 are the characteristic functions of FSR$_1$ and FSR$_2$, respectively. Then, FSR$(f_1; f_2)$ denotes the cascade connection of the FSR$_1$ into the FSR$_2$. Then it was shown in [18] that the FSR with a characteristic function $f_1 * f_2$ and the FSR$(f_1; f_2)$ generate the same family of bit sequences. Here the operator $*$ is defined as follows

$$f_1 * f_2(z_0, z_1, \dots, z_{n+m}) = f_1(f_2(z_0, \dots, z_m), \dots, f_2(z_n, \dots, z_{n+m})), \tag{8}$$

for two Boolean functions $f_1(x_0, \dots, x_n)$ and $f_2(y_0, \dots, y_m)$. It is easy to check that $f_1 * f_2$ is a Boolean function of $(n + m + 1)$-variables and $f_1 * f_2 = f_2 * f_1$ when both f_1 and f_2 are linear. In FSR$(f_1; f_2)$, the feedback value of FSR$_1$ depends on the states of FSR$_1$ only whereas the feedback value of FSR$_2$ depends on the states of FSR$_2$ and the first state of FSR$_1$. In this case, it is called that FSR$_1$ runs in free-running mode whereas FSR$_2$ runs in scrambler mode. Usually, the free-running mode of FSR$_1$ ensures a large period and most of the statistical properties and scrambler mode of FSR$_2$ introduces nonlinearity. There are several examples of bit-oriented cascade systems (CSs) in literature. One example of a cascade connection of FSRs is Grain-128 [13], one finalist of the CAESAR competition. In this case, FSR$_1$ and FSR$_2$ are taken as a primitive LFSR and an NFSR, respectively. Hu et al. [22] have showed that the period of FSR$_1$ divides the period of FSR$(f_1; f_2)$.

There is another variant of cascade connection of FSRs where all the FSRs run in scrambler mode. ACORN [42] and TRIVIUM [13] ciphers use this type of CS. In ACORN, six LFSRs are used whereas three NFSRs are used in TRIVIUM. The above three ciphers fall in bit-oriented FSRs and in the following, the concept of word-oriented cascade connection is discussed. Here the first WFSR runs in free-running mode and the remaining runs in scrambler mode. Then, their period and different randomness properties are studied.

Suppose WFSR$_1$ and WFSR$_2$ are two WFSRs and WFSR$_1$ is cascaded into WFSR$_2$ as illustrated in Fig. 9.7. Suppose the orders of WFSR$_1$ and WFSR$_2$ are n_1 and n_2, respectively with common word size m. Let $\mathbf{S}_0 = (\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{n_1-1})$ and $\mathbf{T}_0 = (\mathbf{t}_0, \mathbf{t}_1, \dots, \mathbf{t}_{n_2-1})$ be the initial states of WFSR$_1$ and WFSR$_2$, respectively. Consider $F_1(\boldsymbol{x}_0, \dots, \boldsymbol{x}_{n_1-1})$ is the feedback function of WFSR$_1$ and $F_2(\boldsymbol{x}_0, \dots, \boldsymbol{x}_{n_2-1})$ is the feedback function of WFSR$_2$. Then, $F_1 : \mathbb{F}_{2^m}^{n_1} \to \mathbb{F}_{2^m}$ and $F_2 : \mathbb{F}_{2^m}^{n_2} \to \mathbb{F}_{2^m}$. Let $g : \mathbb{F}_{2^m} \to \mathbb{F}_{2^m}$ be a bijective function. Then, the feedback value of WFSR$_2$ is calculated as follows

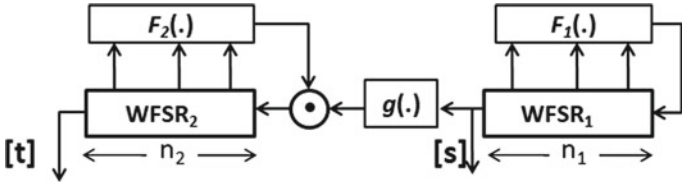


Fig. 9.7 The cascade connection of WFSR_1 into WFSR_2

$$\mathbf{t}_{i+n_2} = g(\mathbf{s}_i) \odot F_2(\mathbf{t}_i, \mathbf{t}_{i+1}, \ldots, \mathbf{t}_{i+n_2-1}), \;\; for\ i \geq 0, \tag{9}$$

where the operations $\oplus$ or modular addition can be used in place of $\odot$. Consider [**s**] and [**t**] are generated by WFSR_1 and WFSR_2, respectively.

Theorem 9 *If $WFSR_1$ is periodic and $WFSR_2$ is nonsingular, then [$\boldsymbol{t}$] is periodic. Let P_s and P_t be the period of [$\boldsymbol{s}$] and [$\boldsymbol{t}$], respectively, then P_s divides P_t.*

Proof It is obvious that the sequence [**t**] will be *ultimately* periodic and let Q_t be the preperiod of [**t**]. Then, $\mathbf{t}_i = \mathbf{t}_{i+P_t}$ for any $i \geq Q_t$ therefore, $\mathbf{t}_{i+n_2} = \mathbf{t}_{i+n_2+P_t}$. So by Eq. (9), $g(\mathbf{s}_i) + F_2(\mathbf{t}_i, \ldots, \mathbf{t}_{i+n_2-1}) = g(\mathbf{s}_{i+P_t}) + F_2(\mathbf{t}_{i+P_t}, \ldots, \mathbf{t}_{i+P_t+n_2-1})$. But P_t is the period of [**t**] and thus, $g(\mathbf{s}_i) = g(\mathbf{s}_{i+P_t})$ for all $i \geq Q_t$. As g is bijective, $\mathbf{s}_i = \mathbf{s}_{i+P_t}$. This implies that $\mathbf{s}_i = \mathbf{s}_{i+P_t}$ for any $i \geq 0$ as [**s**] is periodic. But, the period of [**s**] is P_s and hence P_s divides P_t.

Suppose, [**t**] is not periodic i.e., $Q_t > 0$. Then, $\mathbf{t}_{Q_t+n_2-1} = \mathbf{t}_{Q_t+n_2-1+P_t}$ as P_t is the period of [**t**] and $Q_t + n_2 - 1 \geq Q_t$. This implies, $g(\mathbf{s}_{Q_t-1}) \odot F_2(\mathbf{t}_{Q_t-1}, \ldots, \mathbf{t}_{Q_t+n_2-2}) = g(\mathbf{s}_{Q_t-1+P_t}) \odot F_2(\mathbf{t}_{Q_t-1+P_t}, \ldots, \mathbf{t}_{Q_t+n_2-2+P_t})$. As P_s divides P_t, it implies that $\mathbf{s}_{Q_t-1} = \mathbf{s}_{Q_t-1+P_t}$ therefore, $g(\mathbf{s}_{Q_t-1}) = g(\mathbf{s}_{Q_t-1+P_t})$. Thus, $F_2(\mathbf{t}_{Q_t-1}, \ldots, \mathbf{t}_{Q_t+n_2-2}) = F_2(\mathbf{t}_{Q_t-1+P_t}, \ldots, \mathbf{t}_{Q_t+n_2-2+P_t})$. Again, FSR_2 is nonsingular and by Theorem 1, F_2 can be expressed as $F_2(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1}) = f_{20}(\mathbf{x}_0) \odot f_{21}(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1})$ with f_{20} is one-one. Thus $f_{20}(\mathbf{t}_{Q_t-1}) = f_{20}(\mathbf{t}_{Q_t-1+P_t})$ and $\mathbf{t}_{Q_t-1} = \mathbf{t}_{Q_t-1+P_t}$. Thus, $\mathbf{t}_i = \mathbf{t}_{i+P_t}$ for any $i \geq Q_t - 1$. This is a contradiction to the fact that Q_t is the preperiod of [**t**]. Hence $Q_t = 0$ and this completes the proof. □

Now onwards, we study only the CS of two WFSRs. First, the CS of two MRMMs is discussed in the following.

Cascade connection of two MRMMs: Consider the CS comprising of two primitive MRMMs: $\text{MRMM}_1(m_1, n_1)$ and $\text{MRMM}_2(m_2, n_2)$. The CS can be expressed in terms of a matrix. Consider $M_1(x) = x^{n_1} - C_{n_1-1}x^{n_1-1} - \cdots - C_0$ is the matrix polynomial of MRMM_1 where each C_i is an $m_1 \times m_1$ matrix. Similarly, let $M_2(x) = x^{n_2} - D_{n_2-1}x^{n_2-1} - \cdots - D_0$ be the matrix polynomial of MRMM_2 where each D_i is an $m_2 \times m_2$ matrix. If the companion matrices of MRMM_1 and MRMM_2 are T_1 and T_2, respectively, then

$$(T_1)_{m_1n_1\times m_1n_1}=\begin{bmatrix}\mathbf{0} & I & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & I\\ C_0 & C_1 & \cdots & C_{n_1-1}\end{bmatrix}_{n_1\times n_1}, \tag{10}$$

and

$$(T_2)_{m_2n_2\times m_2n_2}=\begin{bmatrix}\mathbf{0} & I & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & I\\ D_0 & D_1 & \cdots & D_{n_2-1}\end{bmatrix}_{n_2\times n_2}. \tag{11}$$

Here I and $\mathbf{0}$ are Identity and zero matrices of dimension $m_1 \times m_1$ in T_1 and, $m_2 \times m_2$ in T_2. Define an $(m_1n_1 + m_2n_2) \times (m_1n_1 + m_2n_2)$ matrix $\widetilde{T}$ as follows

$$\widetilde{T}=\begin{bmatrix}T_1 & 0\\ B & T_2\end{bmatrix}, \tag{12}$$

where the matrix B is given as

$$(B)_{m_2n_2\times m_1n_1}=\begin{bmatrix}\mathbf{0}\,\mathbf{0}\cdots\mathbf{0}\\ \mathbf{0}\,\mathbf{0}\cdots\mathbf{0}\\ \vdots\;\vdots\;\ddots\;\vdots\\ \widetilde{I}\,\mathbf{0}\cdots\mathbf{0}\end{bmatrix}_{n_2\times n_1}.$$

Here the entries of B are $m_2 \times m_1$ matrices and the entries of the matrix $\widetilde{I}$ are 0 except $min(m_1, m_2)$ entries as 1 s along the diagonal. It is straightforward that the product of the characteristic polynomial of the matrices T_1 and T_2 is the characteristic polynomial of the matrix $\widetilde{T}$. It can be checked that $\widetilde{T}$ is the matrix representation of the CS of two MRMMs. Now, for $j \geq 1$, $\sum_{k=0}^{j-1} T_2^k B T_1^{j-k-1}$ is well defined and it can be verified that

$$\widetilde{T}^j=\begin{bmatrix}{T_1}^j & \mathbf{0}\\ \sum_{k=0}^{j-1} T_2^k B T_1^{j-k-1} & {T_2}^j\end{bmatrix}. \tag{13}$$

For any $m_2n_2 \times m_1n_1$ order matrix $B^{'}$, if $\widetilde{T}^{'} = \begin{bmatrix}T_2 & 0\\ B^{'} & T_1\end{bmatrix}$, then both $\widetilde{T}$ and $\widetilde{T}^{'}$ have same the characteristic polynomials. Before investigating the periodicity of the matrix $\widetilde{T}$, let us see the following trivial result.

Lemma 2 *For relatively prime integers m, n, the set $\{kn \ mod \ m\}_{k=0}^{m-1}$ is equal to* $\{0, 1, \ldots, m-1\}$.

Theorem 10 *If the period of MRMM$_1$ and MRMM$_2$ are P_1 and P_2, respectively with $gcd(P_1, P_2) = 1$, then the period of $\widetilde{T}$ is P_1P_2.*

Proof Let $\widetilde{T}$ has period $j = kP_1P_2 + r$ where $0 < r < P_1P_2$. Then both P_1 and P_2 do not divide r and so, either $T_1^r \neq I$ or $T_2^r \neq I$. Since $\widetilde{T}^j = I$, this implies $\begin{bmatrix} T_1^j & 0 \\ \sum_{i=0}^{j-1} T_2^i B T_1^{j-i-1} & T_2^j \end{bmatrix} = I$. This leads to a contradiction as neither T_1^r nor T_2^r is equal to I. Therefore, $j = kP_1P_2$. Now using Lemma 2 with $j = P_1P_2$,

$$\begin{aligned}
\sum_{i=0}^{j} T_2^i B T_1^{j-i} &= BT_1^j + T_2BT_1^{j-1} + \ldots + T_2^{P_2-1}BT_1^{j-(P_2-1)} \\
&\quad + T_2^{P_2}BT_1^{j-P_2} + \ldots + T_2^{2P_2-1}BT_1^{j-(2P_2-1)} \\
&\quad \vdots \\
&\quad + T_2^{(P_1-1)P_2}BT_1^{j-(P_1-1)P_2} + \ldots + T_2^{P_1P_2-1}BT_1 + T_2^{P_1P_2}B \\
&= BT_1^j + T_2BT_1^{j-1} + \ldots + T_2^{P_2-1}BT_1^{j-(P_2-1)} \\
&\quad + BT_1^{j-P_2} + T_2BT_1^{j-(P_2+1)} + \ldots + T_2^{P_2-1}BT_1^{j-(2P_2-1)} \\
&\quad \vdots \\
&\quad + BT_1^{j-(P_1-1)P_2} + \ldots + T_2^{P_2-1}BT_1 + B \\
&= \left(B + T_2BT_1^{-1} + \ldots + T_2^{P_2-1}BT_1^{-(P_2-1)}\right) \\
&\quad \left(T_1^j + T_1^{j-P_2} + \ldots + T_1^{j-(P_1-1)P_2}\right) + B \\
&= \left(\sum_{i=0}^{P_2-1} T_2^i B T_1^{-i}\right)\left(\sum_{i=0}^{P_1-1} T_1^{j-i}\right) + B \\
&= B.
\end{aligned}$$

Therefore, $\widetilde{T}^{j+1} = \begin{bmatrix} {T_1}^{j+1} & 0 \\ \sum_{i=0}^{j} T_2^i B T_1^{j-i-1} & {T_2}^{j+1} \end{bmatrix} = \begin{bmatrix} T_1 & 0 \\ B & T_2 \end{bmatrix} = \widetilde{T}$. This implies P_1P_2 is the period of $\widetilde{T}$. □

Theorem 11 *Consider MRMM*$_1(m_1, n_1)$ *and MRMM*$_2(m_2, n_2)$ *are two primitive MRMMs such that* m_1n_1, m_2n_2 *are relatively prime. If the cascade connection of MRMM*$_1$ *into MRMM*$_2$ *generates the nonzero word sequence* $[\boldsymbol{t}]$*, then* $(2^{m_1n_1} - 1)(2^{m_2n_2} - 1)$ *and* $m_1n_1 + m_2n_2$ *are the period and linear complexity of* $[t^{(j)}]$*, respectively.*

Proof As m_1n_1, m_2n_2 are relatively prime, the periods of MRMM$_1$ and MRMM$_2$ are also relatively prime. Thus by Theorem 10, $(2^{m_1n_1} - 1)(2^{m_2n_2} - 1)$ is the period of each component bit sequence. Again the characteristic polynomials of T_1 and T_2 are different and primitive as both the primitive MRMMs are different. As the product of characteristic polynomials of T_1 and T_2 is the characteristic polynomial of $\widetilde{T}$, the characteristic polynomial and minimal polynomials of $\widetilde{T}$ are the same. Thus, by [31, Lemma 1], $m_1n_1 + m_2n_2$ is the linear complexity of each component bit sequence. This completes the proof. □

Cascade connection of MRMM and LFG: The CSs comprised of two MRMMs only are discussed in the previous section. Two more CSs are described in this section. One consists of two LFGs whereas the second CS comprises one MRMM and one LFG. Suppose LFG$_1$ is cascaded into LFG$_2$ in the former case and both LFGs are additive primitive LFGs. If the nonzero word sequence generated by this CS is [$\mathbf{t}$], then the following result follows.

Theorem 12 *Let n_1 and n_2 be the order of LFG$_1$ and LFG$_2$, respectively with common word size m. If n_1, n_2 are relatively prime, then $(2^{n_1} - 1)(2^{n_2} - 1)2^{m-1}$ is the period of* [$\boldsymbol{t}$].

Proof As LFG$_1$ is primitive and the period of the sequence generated by LFG$_1$ is $(2^{n_1} - 1)2^{m-1}$. By Theorem 9, $(2^{n_1} - 1)2^{m-1}$ divides the period of [$\mathbf{t}$] as additive LFG is always nonsingular. Let the bit sequence consisting of the first least significant bit (LSB) of each word of [$\mathbf{t}$] be [$t^{(1)}$]. Then, it can be checked that [$t^{(1)}$] is generated by a CS of two primitive LFSRs. Thus, the period of [$t^{(1)}$] is $(2^{n_1} - 1)(2^{n_2} - 1)$ by [7, Corollary 14] and hence $(2^{n_1} - 1)(2^{n_2} - 1)2^{m-1}$ divides period of [$\mathbf{t}$] as period of [$t^{(1)}$] divides period of [$\mathbf{t}$].

As both the LFGs are primitive, it is obvious that the period of [$\mathbf{t}$] divides $lcm((2^{n_1} - 1)2^{m-1}, (2^{n_2} - 1)2^{m-1})$. But n_1, n_2 are relatively prime, hence $lcm((2^{n_1} - 1)2^{m-1}, (2^{n_2} - 1)2^{m-1}) = (2^{n_1} - 1)(2^{n_2} - 1)2^{m-1}$. This completes the proof. □

By using similar arguments, the periodicity of a CS comprising an LFG and an MRMM can be derived. The theorem is as follows.

Theorem 13 *Let n_1 and n_2 be the order of LFG and MRMM, respectively with common word size m. If n_1 and mn_2 are relatively prime, then $(2^{n_1} - 1)(2^{mn_2} - 1)2^{m-1}$ is the period of* [$\boldsymbol{t}$].

Cryptanalysis of word-based CSs: Till now, different CSs comprised of MRMMs and LFGs are discussed along with their periodicity and linear complexity. In this section, a cryptanalytic attack on those CSs is described with its computational complexity. The aim is to find a general method for reconstructing the initial states of FSRs of the CS from a given portion of the output word sequence. Consider the output word sequence of the CS is [$\mathbf{t}$]. If $g(.)$ is considered as the Identity function in Eq. (9), then $\mathbf{t}_{i+n_2} = \mathbf{s}_i \odot F_2(\mathbf{t}_i, \mathbf{t}_{i+1}, \ldots, \mathbf{t}_{i+n_2-1})$, for $i \geq 0$. Consider the CS comprising of two primitive LFGs where LFG$_1$ is cascaded into LFG$_2$ and n_1 and n_2 are the order of LFG$_1$ and LFG$_2$, respectively. If [$t^{(1)}$] is the bit sequence consisting of the first LSB of each output word of the CS, then [$t^{(1)}$] is a bit sequence generated by a CS of two primitive LFSRs. Then from any $2(n_1 + n_2)$ consecutive bits of [$t^{(1)}$], the initial states and respective feedback functions of both LFSRs can be recovered by Berlekamp-Massey algorithm [26]. But in this scenario, the feedback functions of the LFSRs and LFGs are same. Thus, the initial states $\mathbf{s}_0, \ldots, \mathbf{s}_{n_1-1}$ can be computed using Eq. (9), $\mathbf{s}_i = (\mathbf{t}_{n_2+i} - F_2(\mathbf{t}_i, \ldots, \mathbf{t}_{n_2-1+i}))\, mod\, 2^m$, for $i \geq 0$, once [$\mathbf{t}$] is known.

The words $\mathbf{t}_0, \ldots, \mathbf{t}_{n_2-1}$ should not be used as keystream words as they are assumed to be part of the secret key. In this case, assume the keystream words

$\{\mathbf{t}_{n_2+i}\}_{i\geq 0}$ are known. Then, it is again possible to find the feedback polynomials of both the LFGs using the Berlekamp-Massey algorithm on $[t^{(1)}]$. Once the function F_2 is known, the word $\mathbf{s}_{n_2+i}$ can be computed using $\mathbf{s}_{n_2+i} = (\mathbf{t}_{2n_2+i} - F_2(\mathbf{t}_{n_2+i}, \ldots, \mathbf{t}_{2n_2-1+i}))\ mod\ 2^m$, for $i \geq 0$. Again, $F_1(.)$ is linear and thus $\mathbf{s}_{n_2-1}$ can be calculated from $\mathbf{s}_{n_2+n_1-1} = F_1(\mathbf{s}_{n_2-1}, \mathbf{s}_{n_2}, \ldots, \mathbf{s}_{n_2+n_1-2})$. Continuing this procedure, all initial states of LFG$_1$ can be retrieved. To mount this attack in this scenario, $2(n_1 + n_2)$ number of consecutive words of $[\mathbf{t}]$ is sufficient. The following algorithm summarizes this attack.

Algorithm 3 : Attack on LFG-based CSs

```
Require: The output word sequence $[\mathbf{t}]$.
Ensure: The initial states of LFGs.
If $[\mathbf{t}]$ is a zero sequence, then return all states of LFGs as $\mathbf{0}$.
Else, collect $[t^{(1)}]$, the first LSB of each word of $[\mathbf{t}]$. Then, find the feedback polynomial of LFGs
    by applying the Berlekamp-Massy algorithm on $[\mathbf{t}]$. Using Eq. (9), compute the initial states of
    LFGs.
```

This attack is applicable when the linear complexity of $[t^{(1)}]$ is $(n_1 + n_2)$ or less. To counter this attack, one easy solution is to destroy the linear structure of $[t^{(1)}]$ by applying a nonlinear bijective function $g(.)$, for example, the use of an S box. The linear relation in the first LSB of $[\mathbf{t}]$ in the case of an LFG-based CS can be destroyed simply by using a rotation by a nonzero value in the feedback calculation of F_2.

Conclusion

A cryptographically secure bitstream generator must have good statistical properties along with a large period and high linear complexity. This chapter discusses only FSR-based keystream generators. Bit-oriented FSRs are extensively used in several stream ciphers including the finalists of recent competitions like eSTREAM competition and CAESAR competition. It is shown that these bit-oriented FSR-based keystream generators are efficient in hardware platforms. However, they do not take advantage of modern processors whereas WFSRs exploit this advantage and significantly improve the throughput. In this chapter, a general expression for WFSRs is described and then a necessary and sufficient condition for a nonsingular WFSR is provided. Table 9.1 shows a list of some well-known FSRs as the special case of WFSR. Columns 1 and 2 of Table 9.1 tell the name of the FSR and the value of the degree of feedback function F, respectively. Columns 3 and 4 give the value of the word size m and the operation that is used for $\sum$ in Eq. (1). Column 5 says the multiplication operation that is used when $deg(F) > 1$.

Several word-oriented FSR-based keystream generators with a large period and linear complexity are discussed. Those generators are word-oriented versions of filter generators, shrinking and self-shrinking generators, and cascade generators.

Table 9.1 Special cases of different WFSRs

FSR	$deg(F)$	m	$\sum$	$\prod$
LFSR	1	1	$\oplus$	
NLFSR	> 1	1	$\oplus$	&
MRMM	1	> 1	$\oplus$	
LFG	1	> 1	$+$	

It is expected that other word-oriented nonlinear FSR-based keystream generators like nonlinear combination generators, alternating stop generators, and others will be available in the literature soon. Since both period and linear complexity are large in the case of most of the word-oriented generators, they could be used as one of the primitives in the design of symmetric-key crypto algorithm, especially when the crypto algorithm is to be implemented in processors.

References

1. Bassham L, Rukhin A, Soto J, Nechvatal J, Smid M, Barker E, Leigh S, Levenson M, Vangel M, Banks D, Heckert N, Dray J (2010) A statistical test suite for random and pseudo random number generators for cryptographic applications. In: Special publication (NIST SP) - 800-22 Rev 1a. https://csrc.nist.gov/publications/detail/sp/800-22/rev-1a/final
2. Biham E, Anderson RJ, Knudsen LR (1998) Serpent: a new block cipher proposal. In: Fast software encryption, FSE. LNCS, vol 1372. Springer-Verlag, pp 222–238
3. Bishoi SK, Haran HK, Hasan SU (2017) A note on the multiple-recursive matrix method for generating pseudorandom vectors. Discret Appl Math 222:67–75
4. Bishoi SK, Matyas V (2018) Investigating results and performance of search and construction algorithms for word-based LFSRs, σ-LFSR s. Discret Appl Math 243:90–98
5. Bishoi SK, Maharana SN (2017) Xorshift RNG from primitive polynomial. Theoret Appl Inf 29:1–13
6. Bishoi SK, Senapati K, Shankar BR (2020) Shrinking generators based on σ-LFSR s. Discret Appl Math 285:493–500
7. Bishoi SK, Senapati K, Shankar BR (2022) Bitstream generators using multiple-recursive matrix methods. https://doi.org/10.21203/rs.3.rs-2105578/v1
8. Brent RP (1994) On the periods of generalized Fibonacci recurrences. Math Comput 63(207):389–401
9. CAESAR: competition for authenticated encryption: security, applicability, and robustness. https://competitions.cr.yp.to/caesar.html
10. Cantor D, Kaltofen E (1991) On fast multiplication of polynomials over arbitrary algebras. Acta Inform 28:693–701
11. Chetry MK, Bishoi SK, Matyas V (2019) When lagged Fibonacci generators jump. Discret Appl Math 267:64–72
12. Coppersmith D, Krawczyk H, Mansour Y (1994) The shrinking generator. In: Advances in cryptology - CRYPTO'93. LNCS, vol 773, pp 22–39
13. Cryptographic competitions eSTREAM: the ECRYPT stream cipher project. http://Competitions.cr.yp.to/estream.html
14. Ekdahl P, Johansson T (2003) A new version of the stream cipher SNOW. In: Selected areas in cryptography, SAC 2002. LNCS, vol 2595. Springer-Verlag, pp 47–61

15. Ferguson N, Whiting D, Schneier B, Kelsey J, Lucks S, Kohno T (2003) Helix: fast encryption and authentication in a single cryptographic primitive. In: Fast software encryption, FSE 2003. LNCS, vol 2887, pp 330–346
16. Fredricksen H (1982) A survey of full length nonlinear shift register cycle algorithms. SIAM Rev 24(2):195–221
17. Ghorpade SR, Hasan SU, Kumari M (2011) Primitive polynomials, singer cycles, and word-oriented linear feedback shift registers. Des Codes Cryptogr 58(2):123–134
18. Green DH, Dimond KR (1970) Nonlinear product-feedback shift registers. Proc Inst Electr Eng 117(4):681–686
19. Golomb SW (1982) Shift register sequences, Revised. Aegean Park Press, Laguna Hills
20. Gunther CG (1988) Alternating step generators controlled by de Bruijn sequences. In: Advances in cryptology-EUROCRYPT'87. LNCS, vol 304, pp. 5–14
21. Hasan SU, Panario D, Wang Q (2018) Nonlinear vectorial primitive recursive sequences. Cryptogr Commun 10:1075–1090
22. Hu HG, Gong G (2011) Periods on two kinds of nonlinear feedback shift registers with time varying feedback functions. Int J Found Comput Sci 22(06):1317–1329
23. Lidl R, Niederreiter H (1997) Finite fields, encyclopedia of mathematics and its applications 20. Cambridge University Press, Cambridge
24. Marsaglia G (2003) Xorshift RNGs. J Stat Softw 8(14):1-9. http://www.jstatsoft.org
25. Marsaglia G, Tsay L (1985) Matrices and the structure of Random number sequences. Linear Algebra Appl 67:147–156
26. Massey JL (1969) Shift register synthesis and BCH decoding. IEEE Trans Inf Theory (IT-15):122–127
27. Meier W, Staffelbach O (1998) The self-shrinking generator. In: Proceedings of advances in cryptology, EuroCrypt. Springer-Verlag, pp 205–214
28. Menezes AJ, van Oorschot PC, Vanstone SA (1997) Handbook of applied cryptography. CRC Press
29. National Institute of Standards and Technology (2001) https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf
30. National Institute of Standards and Technology (1999) https://csrc.nist.gov/CSRC/media/Publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf
31. Niederreiter H (1995) The multiple-recursive matrix method for pseudorandom number generation. Finite Fields Appl 1:3–30
32. Niederreiter H (1995) Pseudorandom vector generation by the multiple-recursive matrix method. Math Comp 64:279–294
33. Niederreiter H (1996) Improved bound in the multiple-recursive matrix method for pseudorandom number and vector generation. Finite Fields Appl 2:225–240
34. Preneel B (1995) Introduction to the proceedings of the second workshop on fast software encryption. Lecture notes in computer science, vol 1008. Springer, Berlin, pp 1–5
35. Rivest RL, Robshaw MJB, Yin YL (2000) RC6 as the AES. In: AES candidate conference, pp 337–342
36. Rueppel RA (1986) Analysis and design of stream ciphers. Springer
37. Schneier B (1996) Applied cryptography. Willey, New York
38. Schneier B, Kelsey J, Whiting D, Wagner D, Ferguson N (2000) Comments on Twofish as an AES candidate. In: AES candidate conference, pp 355–356
39. Shannon CE (1949) Communication theory of secrecy systems. Bell Syst Tech J 28(4):656–715
40. Stinson DR (2006) Cryptography theory and practice, 3rd ed. Chapman & Hall/CRC
41. Tsaban B, Vishne U (2002) Efficient feedback shift registers with maximal period. Finite Fields Appl 8:256–267
42. Wu H (2020) ACORN: a lightweight authenticated cipher (v3). Candidate for the CAESAR Competition (2016)
43. Zeng G, Han W, He K (2007) Word-oriented feedback shift register: σ-LFSR. http://eprint.iacr.org/2007/114