

# **DESIGN OF A MAC ADDRESS BASED SECURE COMMUNICATION SCHEME**



**Zubaria Zafar Qurashai**

**Fall 2019-MS (IS) - 00000318342**

Supervisor

**Dr. Hasan Tahir Butt**

**Department of Computing**

**A thesis submitted in partial fulfillment of the requirements for the degree of Master of  
Science in Information Security (MS IS)**

**In**

**School of Electrical and Computer Science,  
National University of Sciences and Technology (NUST),  
Islamabad, Pakistan.**

**(June 2023)**

## THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled "Design of a MAC Address Based Secure Communication Scheme" written by ZUBARIA QURASHAI, (Registration No 00000318342), of SEECs has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: \_\_\_\_\_  \_\_\_\_\_

Name of Advisor: Dr. Hasan Tahir

Date: 08-Jun-2023

HoD/Associate Dean: \_\_\_\_\_

Date: \_\_\_\_\_


Signature (Dean/Principal): \_\_\_\_\_

Date: \_\_\_\_\_

## Approval


It is certified that the contents and form of the thesis entitled "Design of a MAC Address Based Secure Communication Scheme" submitted by ZUBARIA QURASHAI have been found satisfactory for the requirement of the degree

Advisor : Dr. Hasan Tahir

Signature:  \_\_\_\_\_

Date: 08-Jun-2023  
\_\_\_\_\_

Committee Member 1:Dr. Sidra Sultana

Signature:  \_\_\_\_\_

09-Jun-2023

Committee Member 2:Mr. Jaudat Mamoon

Signature:  \_\_\_\_\_  
Jaudat Mamoon

Date: 10-Jun-2023  
\_\_\_\_\_

# Dedication

I dedicate this thesis to my **Baba Jaan**, whose unwavering support and prayers have been the guiding light in my scholarly pursuit. Your constant belief in me has fueled my determination to achieve my objectives.

To my **Mama Jaan**, who watches over me from the heavens, I dedicate this thesis with immense pride. I can only imagine the joy and pride you would feel seeing my accomplishments today. Your love and encouragement continue to inspire me, and I dedicate this work to honor your memory.

## Certificate of Originality

I hereby declare that this submission titled "Design of a MAC Address Based Secure Communication Scheme" is my own work. To the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics, which has been acknowledged. I also verified the originality of contents through plagiarism software.

Student Name: ZUBARIA QURASHAI

Student Signature:  \_\_\_\_\_

# Acknowledgement

I want to start by expressing my heartfelt gratitude to the Almighty Allah for empowering me with strength, resilience, and valuable opportunities to complete my studies and embark on this research endeavor. His blessings and guidance have been instrumental in my academic journey.

I am immensely thankful to my supervisor, **Dr. Hasan Tahir Butt**, for his unwavering guidance and support, which have played a pivotal role in the successful completion of this endeavor. His mentorship have been invaluable to me. His expertise in the field has not only shaped the direction of my study but also nurtured my intellectual growth. I am truly fortunate to have had the privilege of working under his mentorship.

I am also indebted to my esteemed committee members, Dr. Sidra Sultana and Mr. Jaudat Mamoon. Their constructive feedback, meticulous attention to detail, and dedication to academic excellence have significantly strengthened the quality of this thesis. I am grateful to their commitment and assistance.

# Table of Contents

Chapter 1 : INTRODUCTION .....	1
1.1 Overview .....	1
1.2 Problem Statement .....	6
1.3 Solution Definition/Description .....	7
1.4 Thesis Motivation .....	9
1.5 Thesis Contribution .....	10
1.6 Thesis Organization .....	10
1.7 Summary .....	10
Chapter 2 : LITERATURE REVIEW .....	12
2.1 Cryptographic Key Storage Issues .....	13
2.2 Key Derivation Functions and Their Usage .....	17
2.3 Symmetric Key Cryptography .....	25
2.4 MAC Address Authentication Schemes .....	28
2.5 Summary .....	33
Chapter 3 : SYSTEM DESIGN .....	34
3.1 One-to-One Communication .....	35
3.2 Group Communication .....	42
3.3 Design Assumptions .....	45
3.4 Summary .....	46
Chapter 4 : IMPLEMENTATION, TESTING AND SECURITY ANALYSIS .....	47
4.1 Implementation .....	47
4.2 Testing .....	52
4.3 Security Analysis .....	57
4.4 Summary .....	63
Chapter 5 : CONCLUSION AND FUTURE WORKS .....	65
5.1 Conclusion .....	65
5.2 Future Works .....	66
REFERENCES .....	69

# List of Abbreviations

<b>MAC address</b>	Media Access Control address
<b>PBKDF2</b>	Password Based Key Derivation Function 2
<b>KDF</b>	Key Derivation Function
<b>PUF</b>	Physically Unclonable Function
<b>TPM</b>	Trusted Platform Module
<b>IoT</b>	Internet of Things
<b>HMAC</b>	Hash-Based Message Authentication Code
<b>MITM</b>	Man In The Middle
<b>ARP</b>	Address Resolution Protocol
<b>NIC</b>	Network Interface Controller
<b>OUI</b>	Organizationally Unique Identifier
<b>OSI</b>	Open Systems Interconnection
<b>PRNG</b>	Pseudo Random Number Generator
<b>HKDF</b>	HMAC-based Extract-and-Expand Key Derivation Function
<b>SK</b>	Secret Key
<b>Km</b>	Master Key
<b>KDC</b>	Key Distribution Center
<b>SCADA</b>	Supervisory Control And Data Acquisition
<b>TPA</b>	Third Party Auditor
<b>NVM</b>	Non Volatile Memory
<b>AES</b>	Advanced Encryption Standard
<b>GPU</b>	Graphics Processing Unit



# List of Tables

Table 1: Libraries and Their Usage.....	48
Table 2: Key Generation Testing For One-to-One Communication.....	53
Table 3: Key Generation Testing For Group Communication .....	54
Table 4: Input Size Testing For One-to-One Communication.....	56
Table 5: Input Size Testing For Group Communication .....	57

# List of Figures

Figure 1 : KDF.....	3
Figure 2 : Literature Review Division.....	13
Figure 3: PBKDF2 Overview.....	20
Figure 4 : HKDF .....	23
Figure 5 : Types of Communication in the Proposed System .....	34
Figure 6 : One-to-One Communication Overview .....	36
Figure 7 : Key Generation.....	37
Figure 8 : Seed Generation.....	38
Figure 9 : PRNG .....	39
Figure 10 : Symmetric Encryption .....	39
Figure 11: Symmetric Decryption.....	40
Figure 12 : Secure Communication between Two Parties using Secret Key.....	41
Figure 13: Group Communication.....	43
Figure 14 : Secure Communication between Group using Master Key.....	44
Figure 15 : Varying Input Sizes for One-to-One Communication - Encryption Time(s) and Decryption Time(s).....	55
Figure 16 : Varying Input Sizes for Group Communication - Encryption Time(s) and Decryption Time(s) .....	56

# Abstract

The storage of cryptographic keys is a significant concern in security systems, as keys can become a target for adversaries. The centralized storage of keys brings about the concern of a single point of failure, where if an adversary gains access to the central storage location, all keys stored within it could be compromised. Additionally, storing keys on devices poses a risk of key theft through physical attacks on the device or through remote attacks where an adversary gains access to the device through a vulnerability in the system. Modern solutions, including PUF and TPM, give rise to inherent problems like cost, stability and reproducibility. Moreover, with the proliferation of keys, the management complexity escalates, which can make it difficult to maintain the security and integrity of cryptographic systems. In addition to the costs associated with key storage, there are also costs associated with the maintenance and management of keys. This includes the cost of personnel who are responsible for managing the keys, as well as the cost of systems and processes that are used to guarantee protection and soundness of the keys. Symmetric keys, in particular, are susceptible to theft since they are typically stored in plaintext form, making them easy to identify and use for unauthorized access. The theft of symmetric keys can have severe consequences, such as the unauthorized disclosure of confidential information or the manipulation of data. To tackle these issues, this research presents a novel MAC address based key extraction and secrecy scheme that eliminates the need for stored keys and improves the security of cryptographic systems by utilizing a device's unique MAC address to derive a symmetric key. This scheme can be used for both one-to-one and group communication and can also be integrated with modern symmetric key algorithms.

## INTRODUCTION

Security is considered a necessary component of modern systems. Engineering and IT teams make great efforts to ensure that security is delivered holistically and reliably. Even though guidelines are followed and algorithms are correctly implemented there are side channels which are exploited by the adversaries thus compromising the safety and security of the system.

This preliminary chapter presents the research route map, an overview of the study, and discusses the contributions the research makes towards resolving issues related to key theft and secure communications. Hence the chapter presents a problem statement, which outlines the key challenges that are addressed by the research. The research delves into the motivation behind the study and elucidates its principal goals and objectives. It concludes with an outline of the structure of the thesis, which provides a roadmap for the reader to navigate the content and understand the organization of the research.

### 1.1 Overview

In this fast era of emerging technologies, the number of connected devices, including but not limited to IoT devices, wearables, edge computing devices, and cloud computing devices [33], is increasing rapidly. With the prevalence of these devices, cyber-attacks are also increasing as the devices are communicating in real time [15] [23] [24]. The robustness of

## *CHAPTER 1: INTRODUCTION*

digital systems heavily dependent on the security and resilience of cryptographic keys utilized to protect data [37] [20]. However, the storage of these keys poses a significant security risk, as adversaries can target and steal stored keys, compromising the entire system [51] [31] [32].

The proliferation of digital devices has given rise to the growing concern of key storage issues [20] [19], as the scale of data that needs to be stored and managed continues to expand [1]. This phenomenon has resulted in a series of challenges, including cost-related issues, such as those associated with maintaining and upgrading hardware infrastructure, as well as hardware management issues that arise due to the complex and heterogeneous nature of modern digital devices.

In the realm of cyber physical systems, advanced techniques such as Physical Unclonable Functions (PUFs) [2] and Trusted Platform Module (TPM) have emerged as crucial components in modern security solutions. While these techniques offer significant potential benefits, they are not without their own limitations and issues, including concerns around reproducibility, stability, and cost. These factors present significant challenges for the practical implementation and deployment of PUF [29] [30] and TPM based systems, and must be carefully considered in any comprehensive approach to cybersecurity.

The aforementioned challenges pose a substantial risk to the security of cyber systems and digital devices, necessitating the development of a cost-effective, easily-manageable, and secure solution. In light of this, we introduce a pioneering approach to key generation that eliminates the necessity of storing keys within digital devices [35] [36]. The approach is based on the use of unique identifiers, such as MAC addresses, and Key Derivation Functions (KDFs), which are linked to symmetric key cryptography for both one-to-one and group communication. By leveraging these techniques, the research effectively eliminates the need for key storage, thereby reducing the risk of key exposure and enabling secure communication without compromising performance or scalability.

## CHAPTER 1: INTRODUCTION

A Key Derivation Function (KDF) is a cryptographic function designed to derive one or multiple secret keys from a provided secret value, which can be a password or a shared secret. It transforms a given input into a key or a set of keys suitable for cryptographic operations. KDFs typically accept multiple inputs, including a master key or secret, optional salt, and contextual information. The master key serves as the primary input and is often a high-entropy value generated by a secure random number generator. The salt, if provided, adds randomness to the key derivation process and enhances security. Contextual information may include additional parameters or metadata related to the specific cryptographic application.

The output of a KDF is the derived key or keying material that can be used for various cryptographic purposes, such as encryption, decryption, or authentication. The derived keys possess certain desirable properties, including sufficient length, randomness, and resistance against cryptographic attacks. Additionally, KDFs aim to ensure that the derived keys are independent of the input, meaning that the knowledge of one derived key does not compromise the security of other derived keys or the original input. For more information, see the below Figure 1:

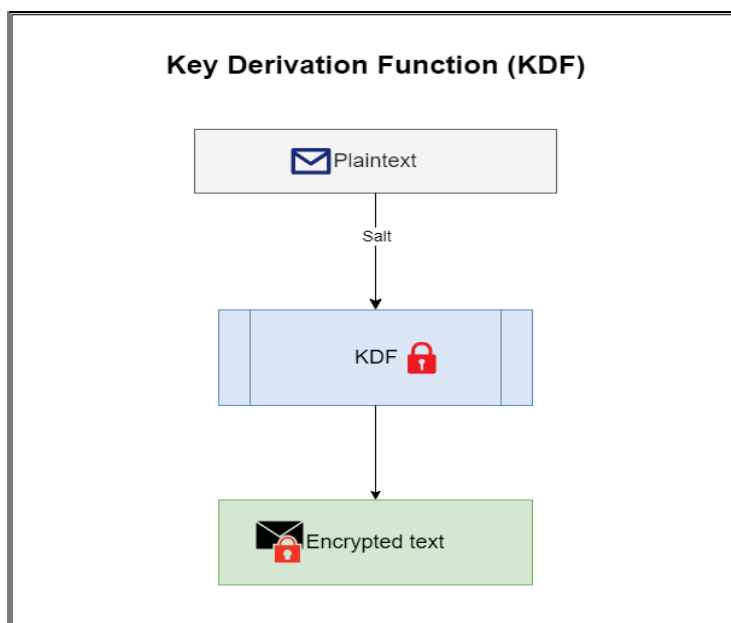


Figure 1 : KDF

## *CHAPTER 1: INTRODUCTION*

KDFs employ various techniques to derive keys, such as hashing, pseudorandom function (PRF) constructions, or employing other cryptographic primitives. These techniques ensure that the derived keys meet the necessary security requirements for the intended cryptographic applications. KDFs play a critical role in key management and cryptographic protocols, enabling secure and reliable generation of keys from a given input, thereby enhancing the overall security of the system.

KDFs are designed to address the need for deriving keys with specific properties from an initial secret or master key. These properties typically include key length, randomness, security, and uniqueness. KDFs ensure that the derived keys are suitable for use in various cryptographic operations, such as encryption, decryption, digital signatures, and key agreement protocols. One important aspect of KDFs is key diversification, which allows for the generation of multiple derived keys from a single master key. This feature is particularly useful in scenarios where different keys are needed for different purposes or when keys are used in multiple independent cryptographic contexts.

KDFs often incorporate additional security measures, such as key stretching or key strengthening, to protect against brute-force attacks and improve resistance against various cryptographic threats. These techniques involve applying additional computational iterations to the key derivation process, effectively increasing the time and resources required to derive keys, thus making them more resistant to exhaustive search attacks. Another important consideration in KDF design is the computational efficiency. While security is paramount, KDFs should also be designed to be computationally efficient to ensure that key derivation can be performed in a reasonable time frame, even on resource-constrained devices.

Standardization bodies, such as the National Institute of Standards and Technology (NIST), provide specifications and guidelines for KDFs. Examples include NIST SP 800-108 and NIST SP 800-56C, which define KDFs suitable for various cryptographic applications, including key establishment protocols. It's worth noting that the choice of an appropriate KDF depends on the specific security requirements and constraints of the cryptographic system at hand. Factors such as the desired level of security, computational resources, and the specific

cryptographic algorithms being used should all be taken into account when selecting a KDF.

By incorporating a well-designed and properly implemented KDF into a cryptographic system, the overall security and integrity of the system's cryptographic keys can be significantly enhanced, reducing the risk of unauthorized access or cryptographic attacks. KDFs are often used in conjunction with password-based key derivation functions (PBKDFs) to generate a cryptographic key from a user password [25] [39] [59]. A commonly used PBKDF is the Password-Based Key Derivation Function 2 (PBKDF2) [7] [54], which leverages a hash function and a salt value to produce a robust key from a user password. The proposed system uses a KDF, specifically PBKDF2, to derive a symmetric key from the device's unique MAC address.

A MAC address, an abbreviation for Media Access Control address, refers to a unique hardware address allocated by the manufacturer to a network interface controller (NIC). The MAC address functions as an exclusive identifier for the network interface controller at the data link layer of the OSI model. It plays a vital role in enabling communication between devices within a local network segment. MAC addresses are crucial for the functioning of many network protocols, including ethernet, as they enable devices to identify and communicate with each other using their assigned hardware addresses [55] [57]. MAC addresses are usually represented as a series of hexadecimal digits, and can be either statically assigned by the manufacturer or dynamically assigned using protocols such as ARP (Address Resolution Protocol). Within its six bytes (48 bits), the MAC address is structured such that the initial three pairs represent the Organizationally Unique Identifier (OUI), which identifies the manufacturer of the NIC. The remaining three pairs constitute the unique identifier assigned by the manufacturer [56].

MAC address-based authentication is a widely employed security mechanism in numerous network-based systems. It is favored for its capability to limit network access exclusively to designated devices, relying on the uniqueness of their MAC addresses [58]. This technique is relatively simple yet effective and is commonly used in wireless network security protocols to control network access by allowing only authorized devices to connect. Additionally, it is utilized in wired networks to restrict access to specific network ports [60]. MAC address-



## *CHAPTER 1: INTRODUCTION*

based authentication is implemented in a variety of security systems, including firewalls, intrusion detection systems, and network access control systems [4] [5] [6]. Despite its limitations, it is a popular security measure. Its ease of implementation and contribution to a multi-layered security approach make it a favored choice among many organizations. It remains a key component of many network security systems and is anticipated to maintain its significance in safeguarding networks in the foreseeable future.

By leveraging the unique properties of MAC addresses and KDFs, the proposed system provides a secure and efficient method of generating symmetric keys on the fly, thus eliminating need for key storage. This approach reduces the risk of key theft and lowers the cost and maintenance associated with key storage. The system can be applied to a wide range of digital devices and offers a robust solution for securing data in both one-to-one and group communication scenarios. It has the potential to be used in a variety of digital devices, including smartphones, laptops, and IoT devices. The use of MAC addresses to generate keys ensures that each device has a unique key, enhancing the security of the system. The utilization of a PRNG in the symmetric encryption function adds another layer of security to the system.

### 1.2 Problem Statement

Cryptographic schemes are widely used to provide security for various types of communications, such as emails, online transactions, and data transfers. These schemes rely on the use of cryptographic keys, which serve the purpose of encrypting and decrypting data. However, it is crucial to recognize that the security of these schemes is intricately tied to the robustness and confidentiality of the cryptographic keys employed. Unfortunately, storing cryptographic keys securely has proven to be a significant challenge, and despite extensive research, no single solution has been found in the literature. The challenge with securely storing keys arises from their inherent vulnerability to various forms of attacks [46]. If an adversary gains access to a stored key, they can use it to decrypt any encrypted data that was encrypted using the key. Thus, protecting stored keys is critical to maintaining the security of the cryptographic scheme. Several techniques have been proposed in the literature to protect stored keys. These include using secure key storage devices, such as smart cards or hardware

## CHAPTER 1: INTRODUCTION

security modules, and implementing strong access control policies. However, these approaches have their limitations [40]. For example, smart cards can be lost or stolen, and access control policies can be bypassed if an adversary gains access to the system through other means.

Given the challenges associated with storing cryptographic keys securely, the current study aims to articulate a problem statement and provide a corresponding solution. Specifically, the study proposes a MAC address based key extraction and secrecy scheme that is applicable to both one- to-one and group communications [37]. This approach enhances cryptographic service provisioning and obviates the need for storing keys. Thus the following problem statement is formulated:

"Cryptographic schemes rely on stored keys to provide security. Owing to their stored nature the keys present an attractive target for adversaries to pursue. This research seeks to formulate a MAC address based key extraction and secrecy scheme that is applicable to both one-to-one and group communications. This approach enhances cryptographic service provisioning and obviates the need for storing keys".

### 1.3 Solution Definition/Description

The proposed solution includes the following features:

1. The generation of a secret key facilitated by the utilization of the MAC address.
2. The elimination of key storage, thus reducing issues with stored keys.
3. The provision of a highly randomized Pseudo Random Number Generator (PRNG).
4. The generation of a seed using MAC address thus eliminating the need to store seed value separately.
5. The provision of symmetric key encryption for one-to-one communication that works best for both low resourced devices and high resourced devices.
6. The generation of master key by introducing a trusted third party for group communications.

## *CHAPTER 1: INTRODUCTION*

In the proposed system, MAC addresses are utilized as a source of entropy for the generation of symmetric keys for encryption and decryption. Since each device has its own unique MAC address, it can be employed as an input akin to a password, enabling the generation of a key using the PBKDF2 algorithm. By using the MAC address as a source of entropy, the requirement for key storage is obviated, as keys can be generated dynamically whenever necessary, relying solely on the device's MAC address.

After generation, the symmetric key can be employed for both one-to-one and group communication, as well as for symmetric encryption. Symmetric keys belong to a class of cryptographic keys used in symmetric key cryptography wherein a single key serves the dual purpose of encrypting and decrypting data. This approach is widely regarded as one of the most secure cryptography protocols for protecting devices against a diverse array of attacks [3]. In the proposed system, symmetric keys are generated on-demand using PBKDF2 and the MAC address of the device, and are discarded after use, negating the need for key storage. This approach results in a more secure system, since the keys are never stored and are only generated when required, reducing the risk of key exposure [48].

To facilitate one-to-one communication, the use of multiple private keys and a highly robust pseudo-random number generator (PRNG) [3] has been incorporated. The seed value of this PRNG is generated using the MAC address of the device, making it a novel approach within the existing literature. For the purpose of encryption, symmetric encryption [34] has been used which provides equivalent levels of security for both high and low resourced devices [44]. This approach is highly efficient and enables secure communication while minimizing computational overhead.

In contrast, for group communication, the proposed system employs a trusted third party known as a Key Distribution Center (KDC) to facilitate the provision of a master key to the communicating parties. The master key is derived by employing the HKDF (HMAC-based Extract-and-Expand Key Derivation Function) when the communicating parties share their own MAC address based secret key via a secure communication channel [49] [52]. Once the master key has been established, the communicating parties can securely communicate with

each other using the modern symmetric encryption algorithm AES (Advanced Encryption Standard) [12]. This approach is highly efficient and enables secure group communication by minimizing the risk to stored key, exposure or interception.

## 1.4 Thesis Motivation

The motivation for conducting this research lies in the security vulnerabilities associated with the storage of cryptographic keys. Stored keys are prone to compromise, which can result in unauthorized access and exposure of sensitive data. Additionally, key storage and management can be complex and costly, especially in large-scale systems. Therefore, the proposed MAC address-based key extraction and secrecy scheme seeks to eliminate the need for key storage, providing a more secure and flexible approach to cryptographic security. Furthermore, the scheme has the potential to become a standardized solution, leading to greater interoperability and ease of integration between different cryptographic systems. Thus following is a point wise motivation for the resolution of the problem statement:

- **Enhanced security** - storing keys in a system presents the risk of exposure and compromise, which can lead to unauthorized access and manipulation of data. By eliminating key storage, the proposed scheme reduces the risk of key exposure, thereby enhancing the overall security of the system.
- **Ease of implementation** - key storage and management can pose complexities and consume significant time, particularly in large-scale systems. By eliminating the need for key storage, the proposed scheme simplifies the implementation process and reduces the associated costs.
- **Flexibility** - storing keys can limit the flexibility of a cryptographic system, as keys may need to be updated or changed frequently. The proposed scheme, which is based on MAC addresses, provides a more flexible approach that can adapt to changing system requirements.
- **Standardization** - the proposed scheme has the potential to become a standardized approach to cryptographic security, as it eliminates the need for proprietary key storage and management systems. This can lead to greater interoperability and ease of integration between different cryptographic systems.

## 1.5 Thesis Contribution

Based on the problem statement a novel solution has been proposed that studies a detailed system which provisions high levels of security along with communication schemes for one to one and group communications.

1. A novel MAC address-based key extraction and secrecy scheme eliminating the need for stored cryptographic keys and associated vulnerabilities.
2. An adaptable security implementation that allows flexibility in choice of parameters to the system designers.
3. A secure practical solution for targeting one-to-one and group communications.
4. Improves the overall efficiency and effectiveness of cryptographic services.

## 1.6 Thesis Organization

Upon introduction of the proposed system, the organization of this thesis is structured as follows: Chapter 2 provides an in-depth study of the key topics in the literature, which include cryptographic systems, key management techniques etc. Chapter 3 explains the research methodology, which outlines the approach taken to design, develop, and evaluate the proposed MAC address-based key extraction and secrecy scheme. Chapter 4 presents the proposed scheme and all of its features, which include the algorithm for key extraction, the encryption and decryption processes, and the protocol for secure communication. Chapter 5 covers the testing and evaluation of the proposed scheme, which includes the performance analysis, security analysis, and comparison with existing schemes. Chapter 6 presents the conclusion of the research, summarizing the main contributions and findings. Additionally, future works and recommendations for further research are presented to enhance the proposed scheme and address its limitations.

## 1.7 Summary

This chapter presented a comprehensive overview of the proposed MAC address based key extraction and secrecy scheme. The problem statement of the vulnerabilities and

## *CHAPTER 1: INTRODUCTION*

complexities associated with key storage in cryptographic systems has been introduced, and the proposed solution is also discussed. The motivation behind the thesis is explained, which aims to enhance the security and effectiveness of cryptographic services. Additionally, the contributions of the devised framework are discussed, which include its adaptability to changing system requirements and potential for standardization. The organization of the thesis is outlined at the end. The next chapter presents a comprehensive literature review.

### LITERATURE REVIEW

This chapter presents an extensive literature review encompassing the pertinent topics associated with the research problem at hand. It begins by discussing the importance of cryptographic schemes and the challenges associated with storing cryptographic keys securely. It then reviews various hashing algorithms, key derivation functions (KDFs), and symmetric key encryption schemes that are commonly used in cryptographic systems. Additionally, the chapter examines the role of MAC addresses in providing security and explores how they can be used to generate session keys without the need for storing keys. By reviewing the existing literature on these topics, this chapter aims to provide a strong theoretical foundation for the proposed MAC address based key extraction and secrecy scheme. The following types of paper have been chosen to understand the fundamentals as shown in Figure 2:

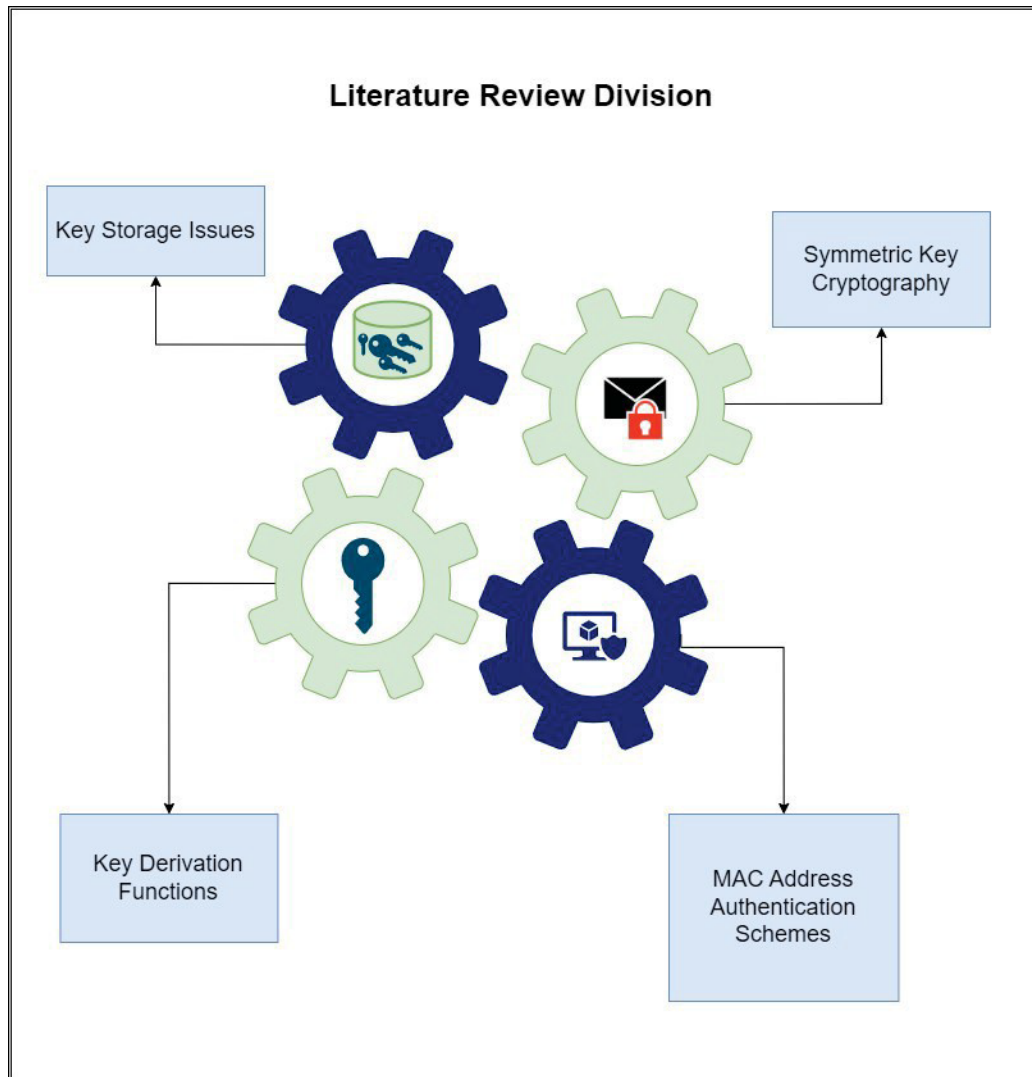


Figure 2 : Literature Review Division

## 2.1 Cryptographic Key Storage Issues

Key storage is a fundamental concern in cryptographic systems, as it is critical to ensure that cryptographic keys are securely stored and managed to prevent unauthorized access or use. However, the task of key storage is not trivial, as it presents several challenges. One of the primary issues with key storage is the potential for key compromise or loss, which can occur if keys are not adequately protected or if they are stored in a vulnerable location. Additionally, the increasing complexity and scale of distributed systems further exacerbate key storage and management challenges, particularly in cases where keys need to be shared across multiple nodes.



## *CHAPTER 2 : LITERATURE REVIEW*

Numerous papers in the literature have addressed key storage issues in different contexts. For instance, some works focused on the risks associated with key storage in mobile devices, proposing secure key storage frameworks for such devices. Other works tackled the challenge of key storage in the context of secure messaging protocols, presenting novel approaches to key management that address scalability and security issues [18][19]. The field of key storage continues to be an active area of research, and it is crucial to continue developing effective strategies for secure key storage and management in diverse contexts. Here we will look at some relevant studies.

### 2.1.1 Key Management Model in DIIS

In distributed networks, Supervisory Control And Data Acquisition (SCADA) devices and operators assume a critical role, operating under resource constraints and low bandwidth data transmission capabilities. DNP3 represents the primary SCADA protocol for reliable communication over physical channels, but its deployment over the internet or open networks may result in severe security vulnerabilities [8]. Deploying DNP3 over open networks necessitates the implementation of suitable security measures to mitigate the associated risks. Cryptography plays a fundamental and indispensable role in the secure management of information, and symmetric and asymmetric cryptography are the two primary classes of cryptography. The primary stages involved in implementing symmetric cryptography encompass key generation, distribution, and revocation. Key generation relies on the utilization of KDFs and random number generators to generate cryptographic keys. Longer keys offer enhanced security, but they also result in increased computational costs. Various techniques, such as Diffie-Hellman, PGP, and key wrapping, are used for key distribution. Research [8] studies a novel intelligent cryptographic key management model specifically developed to bolster the security of distributed industrial systems.. The proposed model uses a hybrid cryptographic algorithm to generate keys at specific intervals, thereby reducing the traffic per key compared to KPS implementation. The model proves to be cost-effective in scenarios requiring more than 34 keys, effectively eliminating the necessity for key storage. The proposed model also supports group-based authentication and identifies certain cryptographic attacks.

## *CHAPTER 2 : LITERATURE REVIEW*

The proposed scheme entails the involvement of 34 keys, which may result in processing delays or prove to be an inefficient approach when employed in smaller or resource-constrained devices. This can be attributed to the increased computational burden incurred by the generation and distribution of a large number of keys, which may exceed the computational capabilities of resource-limited devices. Furthermore, the allocation of memory space to store a large number of keys may not be feasible for devices with limited storage capacity, thereby rendering the proposed scheme unsuitable for deployment in such scenarios. Hence, it is imperative to evaluate the computational and storage capabilities of the targeted devices to ascertain the feasibility and effectiveness of the proposed scheme.

### **2.1.2 Data Integrity Auditing without Private Key Storage**

Numerous data integrity auditing schemes have been proposed to guarantee the trustworthiness of cloud-stored data. Typically, these schemes require users to employ their private keys for generating data authenticators, necessitating possession of a hardware token and memorization of a password to activate the private key [9]. However, the loss of the hardware token or forgetting the password can render existing data integrity auditing schemes inoperative. To address this issue, the authors [9] have proposed an innovative approach called "data integrity auditing without private key storage" and designed a corresponding scheme. This scheme leverages biometric data, like iris scans or fingerprints, as the user's "fuzzy" private key, eliminating the need for a hardware token. The proposed scheme uses a linear sketch with coding and error correction processes to verify the user's identity. Additionally, the authors have developed a new signature scheme that supports blockless verifiability and is compatible with the linear sketch. The scheme accomplishes several objectives for enabling data integrity auditing without private key storage in secure cloud storage. These objectives include ensuring auditing correctness, where the proof generated by the cloud for correctly stored user data can pass Third Party Auditor (TPA) verification, and achieving auditing soundness, guaranteeing that the cloud cannot pass TPA verification if it lacks complete user data. Lastly, the scheme enables auditing without private key storage by allowing users to utilize biometric data as a fuzzy private key for performing data integrity auditing without the need to store a private key. By fulfilling these goals, the proposed scheme ensures secure and dependable cloud storage with robust data integrity auditing.

Despite the high reliability of the proposed data integrity auditing scheme, the utilization of biometric data for key elimination introduces the possibility of capturing false positives and true negatives, which could potentially lead to security breaches. To mitigate the risk of false positives and true negatives in the proposed data integrity auditing scheme, additional measures such as multi-factor authentication or continuous biometric verification could be implemented. Additionally, regular updates and maintenance of the biometric database would also help ensure its accuracy and prevent security breaches.

### 2.1.3 Building Secure SRAM PUF Key Generators

The importance of securely maintaining keys to protect data in resource-limited devices, particularly those in the Internet of Things (IoT), is discussed in reference [10]. Silicon Unclonable Functions (PUFs) present a cost-effective alternative for secure non-volatile memory (NVM) by offering device-specific secrets. However, PUF reliability is susceptible to thermal noise and environmental changes, rendering them unsuitable as cryptographic keys without a fuzzy extractor. Implementing a fuzzy extractor on resource-constrained devices can be challenging due to its overhead. To address this, the authors propose a secure and lightweight methodology for constructing a PUF key generator tailored for resource-limited devices. They utilize pervasively embedded SRAM in modern microcontroller units and a batteryless computational radio frequency identification (CRFID) device as an illustrative case study. The authors concentrate on reducing token implementation overhead by mitigating response unreliability through two compatible methodologies: reliable response preselection and multiple reference response enrollment during the PUF provisioning phase. They experiment with a reverse fuzzy extractor-based PUF key derivation technique and employ intrinsic SRAM PUF for their investigation, which necessitates no hardware modification or additional area cost. A lightweight PUF key generator implementation is demonstrated on a batteryless CRFID device, followed by security analyses conducted on the implementation.

While PUFs offer a low-cost alternative to secure non-volatile memory for protecting data in resource-limited devices, they also have several limitations. Firstly, the response of a PUF can vary between different manufacturers, making it difficult to establish a universal standard for PUF-based security implementations. Secondly, PUFs can be susceptible to side-channel

attacks, where adversaries can exploit information leakage from the device's power consumption or electromagnetic emissions to deduce the device's secret key. Furthermore, implementing a fuzzy extractor on highly resource-constrained devices can be a challenging task due to its computational overhead. Thus, further research may be necessary to address these limitations and ensure the effectiveness and security of the proposed scheme.

## 2.2 Key Derivation Functions and Their Usage

Key derivation functions (KDFs) are an essential tool in cryptography and are used to derive secret keys from shared secrets or passphrases. They play a vital role in modern cryptographic systems by transforming initial keying material into derived keys with specific properties. One key consideration in KDF design is the ability to adapt to evolving security requirements. KDFs should be designed to allow for future expansion and accommodate changes in cryptographic algorithms or key sizes without compromising security. This flexibility ensures that cryptographic systems can be upgraded or adapted to meet emerging threats and advancements in technology, providing long-term resilience.

Additionally, KDFs are often designed to incorporate additional security features that protect against potential vulnerabilities. For example, some KDFs include countermeasures against side-channel attacks, which exploit information leaked through unintended physical channels such as power consumption or electromagnetic radiation. By integrating countermeasures like constant-time operations or secure masking techniques, KDFs can mitigate the risk of side-channel attacks and enhance the overall security of the derived keys.

Moreover, KDFs can offer a range of customization options to meet specific security requirements. For instance, certain KDFs allow for the inclusion of user-defined parameters or policies, enabling fine-grained control over the key derivation process. This customization can include factors like input data length, iteration counts, or specific cryptographic algorithms used within the KDF. By providing such flexibility, KDFs empower system designers to tailor the key derivation process to the unique security needs of their applications. Furthermore, some advanced KDFs incorporate mechanisms to support key evolution or key refreshment. These features allow for periodic updates of derived keys, which can be crucial in scenarios

## *CHAPTER 2 : LITERATURE REVIEW*

where long-term key secrecy is desired [47]. By periodically refreshing the derived keys, the overall security posture of the cryptographic system is enhanced, reducing the impact of potential key compromise over time.

KDFs not only provide the means to derive keys for cryptographic operations, but they also offer essential features such as adaptability, countermeasures against side-channel attacks, customization options, and support for key evolution. These unique aspects make KDFs a fundamental component in ensuring the security and longevity of cryptographic systems in the face of evolving threats and changing cryptographic requirements. They are designed to be computationally expensive and iterative, making it difficult for adversaries to brute force the derive keys. KDFs have numerous applications, such as password-based key derivation [22], key agreement protocols, and key strengthening. In addition, KDFs can be used to derive keys for secure storage or transmission of data, such as in disk encryption or SSL/TLS. Several KDFs have been proposed in the literature, each with their unique features and trade-offs. Their usage in the literature and how they can be applied in various cryptographic schemes has been seen as follows:

### 2.2.1 Managing Password using AES-256 & PBKDF2

As cyber-attacks become more sophisticated, organizations are implementing a range of security measures to safeguard their servers from intrusion. However, users are still vulnerable due to their tendency to use simple and easily guessed passwords, making them targets for hackers attempting to crack their passwords and access sensitive information. In response, authors have raised awareness about the importance of password security and developed effective solutions to protect users' confidential data. Password managers have become a popular solution because they combine usability and security features, storing and managing all of a user's passwords and generating unique, randomized passwords for users.

To ensure the utmost security of the user's sensitive information, password managers employ a range of encryption techniques. These include the utilization of the AES-256 encryption algorithm and the adoption of the PBKDF2. PBKDF2 incorporates a computationally intensive process known as key stretching, which transforms a user's master password into a cryptographic key. This process significantly complicates an adversary's

## *CHAPTER 2 : LITERATURE REVIEW*

attempts to determine the master password through repeated guessing, effectively thwarting brute-force attacks. Furthermore, PBKDF2's key stretching mechanism is specifically designed to hinder password-cracking software from fully leveraging GPUs, thereby reducing the guess rates from hundreds of thousands to tens of thousands per second. AES, on the other hand, employs symmetric key encryption, employing a single secret key for both encrypting and decrypting data. It is the sole publicly available cipher endorsed by the US National Security Agency (NSA) for safeguarding classified information [11]. The algorithm is designed to provide high levels of security and performance and is widely used in applications ranging from encryption of sensitive data in transit to securing stored data on hard drives.

PBKDF2 takes several inputs to execute its key derivation process. The primary input is the password, serving as the source for generating the cryptographic key. Additionally, PBKDF2 requires a salt, which is a randomly generated value that enhances the complexity and uniqueness of the derived key. The salt is typically stored alongside the derived key to ensure its security. Moreover, PBKDF2 incorporates an iteration count parameter, specifying the number of iterations the key derivation function undergoes, thereby increasing the computational cost and impeding potential attackers. Lastly, PBKDF2 demands a desired output key length, determining the size of the derived key. By employing these inputs effectively, PBKDF2 strengthens the security of password-based key derivation and safeguards sensitive information in diverse applications. Figure 3 shows the working of PBKDF2.

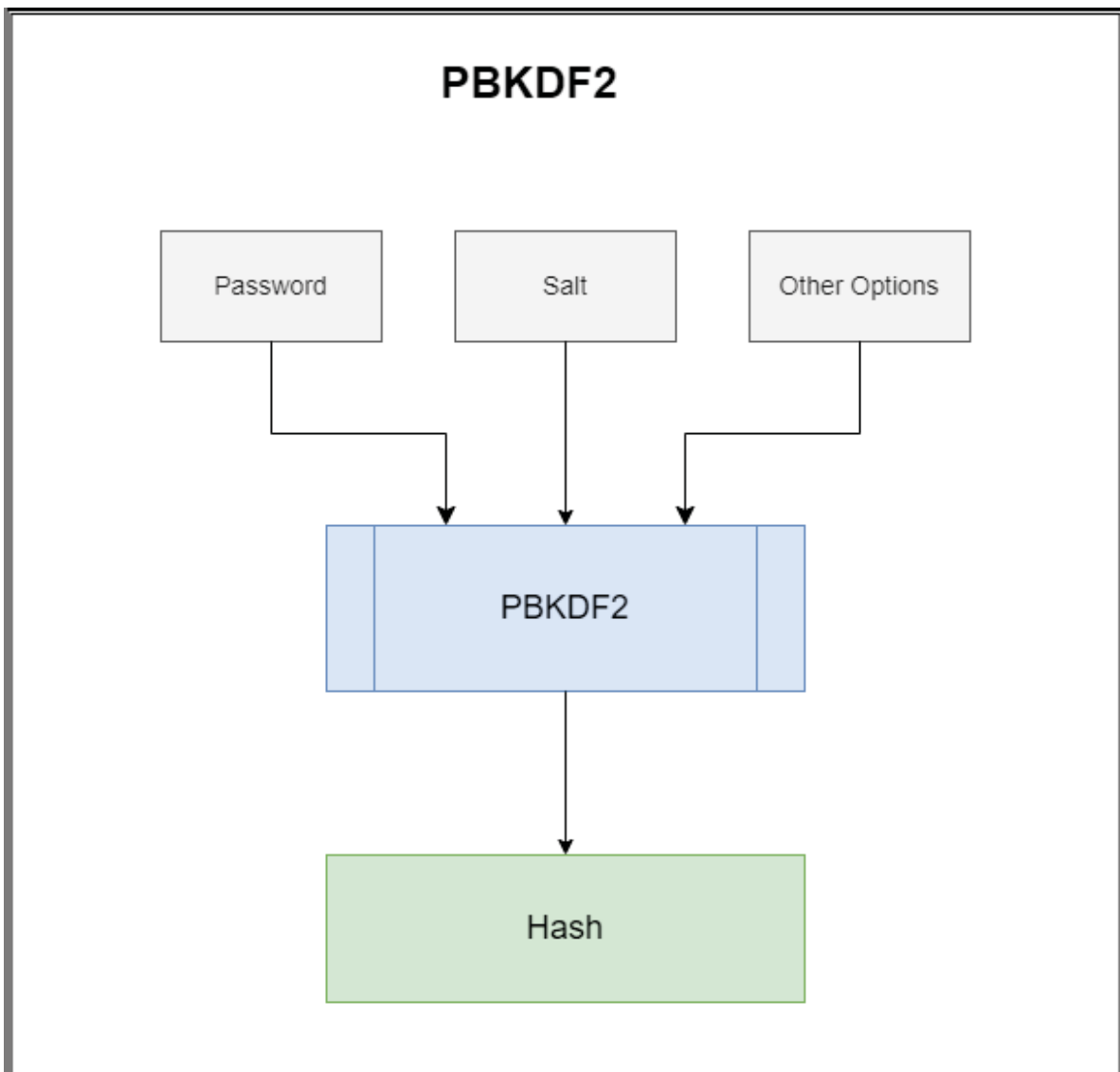


Figure 3 : PBKDF2 Overview

Research [11] proposes the data is subject to encryption using the AES with a key length of 256 bits, and PBKDF2, that is the prevailing industry standard at present. Furthermore, an additional layer of security is implemented by encrypting the username and password using a cryptographic key that is derived from the user's master password. This ensures the confidentiality and security of the user's sensitive data by preventing unauthorized access to the stored passwords. To provide a seamless user experience, password managers can be seamlessly integrated into web browsers as extensions. This integration ensures broad compatibility and user-friendly accessibility, allowing end-users to conveniently utilize password management functionalities while browsing the web. By implementing password

managers, users can protect their sensitive information from cyber-attacks and reduce the risk of becoming targets for hackers [11].

The paper introduces the utilization of PBKDF and AES to implement robust security measures. The use of PBKDF2 as a cryptographic key derivation function generates a cryptographic key from a user's master password and employs a computationally intensive process, thereby enhancing the resistance to brute-force attacks. AES is a symmetric key cryptography algorithm [42] [43]. It employs a single secret key for both encrypting and decrypting data. AES is widely recognized and has gained prominence as the only publicly available cipher authorized by the US National Security Agency (NSA) for securing top-secret information [41]. The combination of PBKDF2 and AES provides a highly secure and effective approach to protect sensitive information from cyber threats.

### 2.2.2 Optimization of PBKDF2 using HMAC-SHA2 and HMAC-LSH Families

The Password-Based Key Derivation Function 2 (PBKDF2) is a widely adopted cryptographic algorithm [16] [17] utilized for generating secure keys for passwords in various applications, including file encryption and authentication systems. However, the entropy of the derived key generated by PBKDF2 is lower compared to a typical cryptographic key, which limits its usability [7]. To address this limitation, it is recommended to increase the number of iteration counts in PBKDF2, enhancing the strength and security of the derived key. However, this leads to a higher computational overhead for generating the derived key. Therefore, [7] proposes various optimization techniques for PBKDF2 unnecessary block operations and improve the internal process of the underlying Pseudo Random Function (PRF). The main goal of this study is to introduce various techniques aimed at optimizing PBKDF2. These techniques focus on reducing redundant block operations and improving the internal process of the underlying Pseudo Random Function (PRF). By implementing these optimizations, the overall efficiency and performance of PBKDF2 can be significantly enhanced.

Through the implementation of the proposed optimization techniques, this research aims to enhance the efficiency of PBKDF2 while ensuring heightened levels of security by increasing



## *CHAPTER 2 : LITERATURE REVIEW*

the number of iteration operations. Moreover, these techniques can be specifically tailored to optimize the performance of PBKDF2 on Graphics Processing Units (GPU) and embedded devices. The approach put forth in this study combines multiple redundant operations and maximizes the utilization of constant values utilized in PBKDF2. To serve as the Pseudo Random Function (PRF) of PBKDF2, two HMAC algorithms are employed: one utilizing the SHA-2 family and the other utilizing the LSH family, which represents the latest hash function developed in South Korea. By employing the proposed implementation techniques, the security level of PBKDF2 can be significantly increased with the use of a larger number of iteration counts. Moreover, the optimization methods proposed in this study can be extended to enhance the performance of PBKDF2 on embedded devices and graphics processing units (GPUs).

The paper presents a novel approach to optimize PBKDF2 using parallel processing and multithreading features. By fully utilizing the constant values in PBKDF2, the proposed method aims to lessen recurring block operations and optimize the internal process of the underlying PRF. PBKDF2 can be used for the provision of robust security.

### 2.2.3 HMAC-based Extract-and-Expand Key Derivation Function (HKDF)

The HMAC Key Derivation Function (HKDF) is a cryptographic algorithm that is based on the HMAC construction. It is a powerful cryptographic algorithm that offers unique advantages in key derivation. One key aspect that sets HKDF apart is its ability to support key hierarchy and extraction of multiple keys from a single secret key material. This hierarchical structure enables the derived keys to be organized and managed in a logical and efficient manner, allowing for fine-grained control over key usage and minimizing the risk of key exposure [52]. The flexibility of HKDF in generating multiple keys from a single input is particularly useful in complex cryptographic systems where different keys are needed for distinct purposes and levels of access.

Another noteworthy feature of HKDF is its ability to incorporate optional contextual information during the key derivation process. This contextual information, such as application-specific parameters or metadata, can be included as additional input to HKDF. By

CHAPTER 2 : LITERATURE REVIEW

considering this information, HKDF can generate derived keys that are tailored to the specific requirements and constraints of the cryptographic application. This customization capability enhances the security and efficiency of the derived keys, ensuring they are optimized for the intended use case.

HKDF takes as input three primary components: a secret key material, optional salt, and contextual information. The secret key material is a random value of sufficient length, typically generated using a secure random number generator. The salt, if provided, adds additional randomness to the key derivation process. The contextual information can include details such as application-specific parameters or metadata. The inputs and outputs of HKDF are shown in the Figure 4.

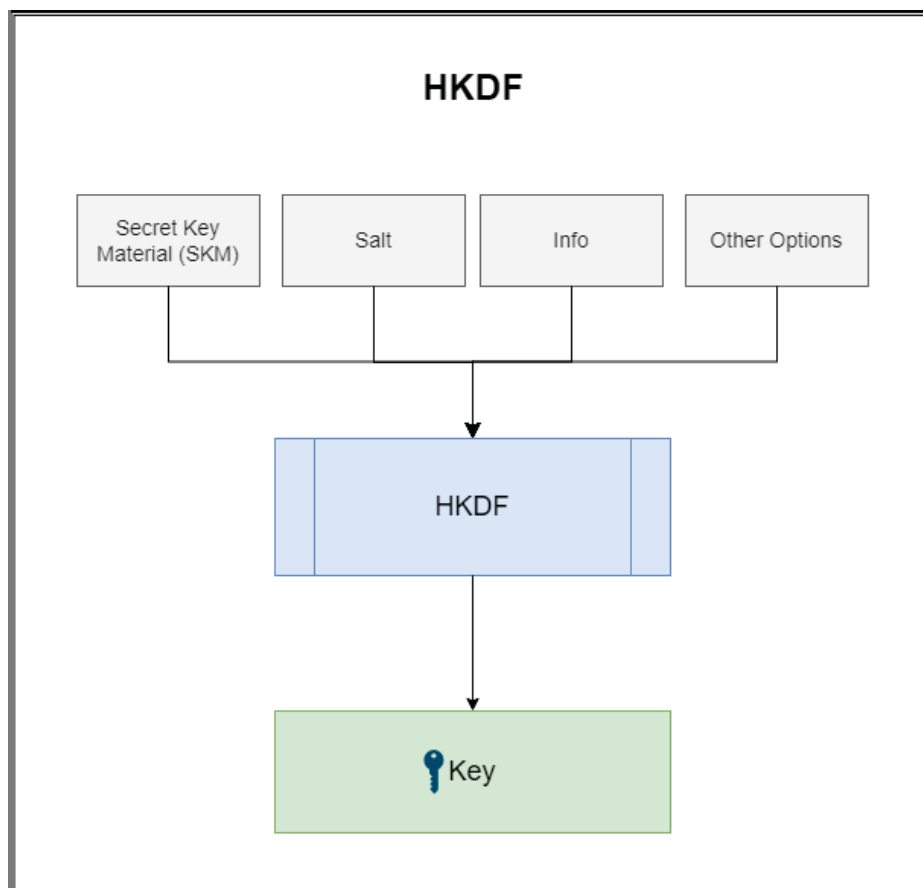


Figure 4 : HKDF

## CHAPTER 2 : LITERATURE REVIEW

The HKDF algorithm consists of two stages: extraction and expansion. In the extraction stage, an HMAC is computed using the secret key material and the salt (if provided) as the input to the HMAC function. The resulting output, known as the pseudorandom key (PRK), is a fixed-length cryptographic key [49].

In the expansion stage, the PRK is further processed to derive the desired output keying material (OKM) of the desired length. This process involves generating additional keys using the PRK and including the contextual information as input to the HMAC function [52]. By carefully managing the expansion process, HKDF guarantees that the derived keys maintain the necessary security properties, such as randomness, independence, and strength against cryptographic attacks. The HKDF algorithm ensures that the derived keys possess the desired security properties, such as pseudorandomness, key independence, and resistance to various cryptographic attacks. Furthermore, HKDF incorporates a secure extraction stage, known as the "extract" phase, which employs a pseudorandom function (PRF) to generate a high-quality pseudorandom key (PRK). This PRK serves as the foundation for the subsequent key expansion phase. The PRK exhibits strong cryptographic properties and is carefully designed to be resistant to various attacks, ensuring the derived keys possess the desired security attributes.

Moreover, HKDF is designed to be a versatile and widely applicable key derivation function. It has been extensively studied and standardized by organizations such as the Internet Engineering Task Force (IETF) due to its robustness and effectiveness. The availability of well-defined specifications and guidelines for HKDF makes it a reliable choice for cryptographic applications across various domains, including secure communication protocols, key establishment, and secure storage systems. By employing the HMAC construction, HKDF provides a secure and reliable approach to key derivation, making it suitable for various cryptographic applications, including key establishment, key refreshing, and cryptographic protocol design. HKDF's ability to support key hierarchy, incorporate contextual information, its secure extraction and expansion phases, and its versatility as a standardized algorithm make it a valuable tool in modern cryptography. These unique features enable HKDF to provide strong, customized, and efficiently derived keys that meet the

stringent security requirements of diverse cryptographic systems.

## 2.3 Symmetric Key Cryptography

Symmetric key cryptography is a cryptographic technique where the same key is utilized for both encrypting and decrypting data. It offers a straightforward and efficient approach to secure information. Several widely recognized algorithms are used for symmetric key cryptography, including the Advanced Encryption Standard (AES), Data Encryption Standard (DES), Triple Data Encryption Standard (3DES), and Blowfish. These algorithms provide strong encryption capabilities and are extensively employed in various applications that require secure data transmission and storage. AES is extensively used in various applications, such as secure communication protocols, electronic payment systems, and file encryption. DES is an older algorithm that is still used in legacy systems, while 3DES provides better security by applying DES three times [42]. Blowfish is a fast and secure algorithm that can be used in various applications, including virtual private networks (VPNs), password protection, and secure email. Below are explored some of these algorithms and their usage in more detail.

### 2.3.1 symKrypt

Researchers have proposed a new and highly secure symmetric-key cryptography algorithm called symKrypt to address the vulnerabilities of traditional symmetric-key cryptography [3]. The symKrypt algorithm introduces the concept of dynamically generated and secret private keys. Multiple private keys are generated and used to encrypt individual blocks of a message, with each block using a different private key. The original message is mixed with the private keys in a confidential manner, and the number of private keys remains unknown. These private keys are generated using a secure pseudo-random number generator that relies on unpredictable initial inputs.

Unlike other variants, symKrypt does not require the exchange of private keys between the sender and receiver during communication. The total number of private keys and rotation information are kept secret and calculated dynamically. For each iteration of a block in the

## *CHAPTER 2 : LITERATURE REVIEW*

message, symKrypt changes its private key for encryption or decryption. Additionally, the private keys are changed for each block of the message. The researchers conducted a theoretical analysis of symKrypt and demonstrated its effectiveness using millions of private keys. They also tested the proposed pseudo-random number generator using the NIST SP 800-22 statistical test suite, where it successfully passed all 15 tests. The symKrypt is a lightweight and robust encryption model that employs multiple private keys, providing a high level of security against various attacks. It represents a significant advancement in symmetric-key cryptography and offers enhanced protection for sensitive data.

The proposed scheme utilizes the use of multiple symmetric keys and PBKDF2 to enhance security. To achieve this, a highly unpredictable random number generator is created. The random number generator utilized in this context is specifically designed to produce a sequence of numbers that lack any identifiable pattern or predictability. This ensures that the generated keys are highly unpredictable and resistant to any attempts by an adversary to guess or deduce them. The goal is to enhance the security of the cryptographic system by incorporating a source of randomness that is robust and effectively mitigates the risk of key compromise. By leveraging this carefully designed random number generator, the cryptographic scheme can maintain a high level of confidentiality and protect sensitive information from unauthorized access or decryption. The use of multiple symmetric keys ensures that the encryption is strong and not easily compromised, while PBKDF2 allows for the generation of secure keys from passwords or passphrases. The combination of these techniques provides a robust and secure encryption scheme that can be used to protect sensitive information.

### 2.3.2 Performance Analysis of Symmetric Key Cryptography

Cryptography plays a critical role in securing data transmission over the internet by employing a range of methods and algorithms. These cryptographic techniques are designed to provide essential security properties such as authentication, confidentiality, integrity, and non-repudiation. There are several cryptographic algorithms available for secure data transmission, but the selection of an appropriate algorithm should consider factors such as robustness, efficiency, cost- effectiveness, high performance, and ease of deployment, which are specific

## *CHAPTER 2 : LITERATURE REVIEW*

to the customer's requirements. In the presented research, various symmetric key cryptographic algorithms, including DES, 3DES, AES, and Blowfish, were evaluated based on practical implementations using Java. The performance analysis focused on important factors such as encryption time, decryption time, entropy, memory usage, throughput, avalanche effect, and energy consumption. Unlike relying solely on theoretical concepts, the proposed work emphasized practical implementations to assess the tradeoff performance in terms of different parameters. Battery consumption and the avalanche effect of the algorithms were specifically discussed. The findings of the research indicated that AES demonstrated superior performance across the overall performance analysis when compared to the other considered algorithms. This suggests that AES is a favorable choice for practical implementations, taking into account its efficiency in terms of encryption time, decryption time, entropy, memory usage, throughput, avalanche effect, and energy consumption.

Encryption algorithms have unique features that make them more suitable for specific applications. In situations where memory and time are critical factors, the blowfish algorithm is a preferable choice due to its efficiency. AES, on the other hand, stands out for its ability to provide the maximum level of security with minimal energy consumption. AES is an ideal choice for applications that require robust encryption with a minimal energy footprint. Finally, if the need is for high-security performance without consuming excessive bandwidth, then the DES algorithm is an excellent choice. DES provides secure communication and message confidentiality without overburdening the network infrastructure. These algorithms offer distinct features that cater to different requirements and should be chosen accordingly to provide optimal performance in specific scenarios.

### **2.3.3 Classical and Physical Security of Symmetric Key Cryptographic Algorithms**

Symmetric key cryptography is vital in securing electronic communication in modern times. To ensure the security of a cipher, it needs to fulfill specific criteria, which can be categorized into classical and physical attacks. Recent research [13] has made significant contributions in advancing the state-of-the-art by exploring both classical and physical attack scenarios. Classical attacks have traditionally relied on rigorous mathematical analysis to identify vulnerabilities in cryptographic systems. The mentioned research has provided fresh insights by introducing two tool-assisted analysis methods: Mixed Integer Linear

## *CHAPTER 2 : LITERATURE REVIEW*

Programming (MILP) and Machine Learning (ML). These methods offer new avenues for analyzing ciphers and enhancing our understanding of their security properties. The utilization of MILP enables a comprehensive exploration of mathematical models, allowing for a systematic search for potential weaknesses in cryptographic algorithms. On the other hand, ML techniques leverage the power of data analysis and pattern recognition to identify vulnerabilities based on extensive training and analysis.. MILP modeling ensures the classical security of ciphers and proposes alternative modeling that is free from possible pitfalls. Physical attacks rely on observing/altering the physical characteristics of a device performing cryptographic operations. These attacks can be classified as fault attacks and side- channel attacks. The study [13] showed that fault attacks weaken ciphers considered secure against classical analysis. The research explored fault attacks from the perspective of classical analysis techniques and proposes countermeasures to common fault attack models. It covered a diverse set of topics ranging from classical attacks to device-dependent attacks. The work formulated fault attacks from the cipher designer's point of view, exploring new avenues to protect against such attacks.

The research presented analyzes the strengths and weaknesses of symmetric key cryptography, a cornerstone of modern electronic communication security. The study introduces new insights and contributions in both classical and physical attack contexts, with a focus on fault attacks and side channel attacks. The researchers propose several countermeasures to protect against these attacks.

### **2.4 MAC Address Authentication Schemes**

MAC address authentication schemes are a common form of access control used in network security. This authentication method involves verifying the unique hardware identifier, known as the MAC address, of a device before allowing access to the network. MAC address authentication can be implemented at various levels, such as at the router or switch level, or even at the application level. While this method is relatively simple to implement and provides an extra layer of security, it is not foolproof as MAC addresses can be spoofed. Therefore, it is often used in conjunction with other security measures, such as usernames and passwords or digital certificates, to strengthen the overall security posture of a network. There is a vast

## *CHAPTER 2 : LITERATURE REVIEW*

amount of literature available on MAC address authentication schemes, providing a wealth of information and insights for researchers presented in the following section.

These schemes have gained significant attention as a means of providing network access control and enhancing security in various environments. One unique advantage of MAC address authentication is its simplicity and ease of implementation. MAC addresses, assigned to NICs, serve as unique identifiers for devices on a network. Leveraging these addresses for authentication allows organizations to establish access policies based on the identity of devices, adding an additional layer of security to their networks. Another important aspect of MAC address authentication schemes is their compatibility with existing network infrastructure. Since MAC addresses are inherent to network communication and are utilized by network switches and routers for forwarding decisions, incorporating MAC address-based authentication can be seamlessly integrated into the existing network architecture. This compatibility minimizes the need for extensive modifications or additional hardware investments, making MAC address authentication a cost-effective solution for access control.

Furthermore, MAC address authentication schemes offer granular control over network access privileges. Network administrators can define access policies based on specific MAC addresses, allowing or denying network connectivity for individual devices. This fine-grained control enables organizations to enforce strict access restrictions, ensuring that only authorized devices can connect to the network. MAC address authentication is particularly valuable in scenarios where device identity verification is crucial, such as in critical infrastructure, government networks, or corporate environments with stringent security requirements.

However, it is essential to acknowledge that MAC address authentication has certain limitations. One significant limitation is that MAC addresses can be easily spoofed or manipulated. Malicious actors can modify their device's MAC address to impersonate authorized devices, bypassing MAC address-based authentication. To mitigate this risk, additional security measures such as cryptographic protocols or secure network protocols should be implemented in conjunction with MAC address authentication to strengthen the overall security posture. Moreover, MAC address authentication may not be suitable for large-



## *CHAPTER 2 : LITERATURE REVIEW*

scale networks or environments with a high volume of devices. Managing and maintaining a comprehensive MAC address-based authentication system can become cumbersome as the number of devices increases. Additionally, MAC addresses may change when devices are replaced or upgraded, leading to administrative overhead in updating access policies and maintaining accurate device records. These schemes offer simplicity, compatibility, and granular control in network access control scenarios. While they provide an additional layer of security and are easy to implement, organizations should be aware of the limitations associated with MAC address authentication and consider complementary security measures to mitigate potential vulnerabilities.

### 2.4.1 Practical Hash-Based Anonymity for MAC Addresses

Given the unique identification capabilities of MAC addresses for individuals or vehicles, concerns over privacy have arisen regarding continuous tracking over large geographic areas. Previous studies have highlighted the vulnerability of hash-based anonymization approaches, specifically concerning MAC addresses. Due to the limited search space of MAC addresses, which can be represented in 39 bits, these anonymization methods can be easily reversed. Frequency-based attacks can effectively enumerate MAC addresses in 31 bits, undermining the effectiveness of such approaches. However, recent research [14] has proposed a practical solution to MAC address anonymization by leveraging computationally expensive hash functions and truncating the resulting hashes to achieve k-anonymity. The paper introduces a mathematical expression that allows for the calculation of the expected collision percentage. It demonstrates that by using digests of 24 bits, it is possible to store up to 168,617 MAC addresses with collision rates below 1%. Additionally, the research presents experimental results that showcase the storage capacity and collision rates achieved using different numbers of bits. Storing datasets of 100 MAC addresses requires 13 bits, 1,000 MAC addresses necessitate 17 bits, and 10,000 MAC addresses can be accommodated within 20 bits while maintaining collision rates of 1% or less.

These findings provide valuable insights into the design and implementation of MAC address anonymization techniques. By employing computationally expensive hash functions and utilizing truncated hashes, it becomes feasible to achieve k-anonymity and mitigate the

## CHAPTER 2 : LITERATURE REVIEW

vulnerabilities associated with frequency-based attacks. The research contributes to the development of practical and secure MAC address anonymization approaches, enhancing privacy protection in data sets that include MAC addresses.

MAC addresses are commonly used for identification purposes in networking systems. Although MAC addresses are commonly represented as 48-bit values, research has shown that the entire range of allocated MAC addresses can be effectively represented using only 39 bits. This finding highlights the fact that a significant portion of the address space remains unused or unallocated. To enhance the security of MAC address tracking, a hash-based approach can be used. Furthermore, a practical approach to MAC address anonymization has been proposed, which involves truncating the resulting hashes to achieve k-anonymity. This technique ensures that multiple MAC addresses are indistinguishable from each other, enhancing privacy protection and reducing the risk of identification. By applying truncation to the hashed values, the uniqueness of individual MAC addresses is effectively concealed, making it harder to link them to specific devices or individuals. However, one major drawback of this proposed scheme is that it is computationally expensive.

### 2.4.2 Hash-MAC-DSDV

IoT-connected CPS systems are susceptible to security threats due to the decentralized deployment of IoT devices. D2D authentication is necessary to ensure information integrity, authenticity, and confidentiality. Most existing solutions rely on centralized techniques, which are computationally and communication-wise costly. To address this issue, a lightweight Hash-MAC- DSDV routing scheme is proposed for multi-WSN CPS devices. The recent study [6] proposed scheme registers MAC addresses in the first phase and advertises them in the second phase to ensure D2D authentication. It allows devices to modify their routing table and utilize one-way hash authentication to transfer data. Evaluation results show that the proposed scheme outperforms existing schemes in terms of attack detection, energy consumption, and communication metrics. The performance of the proposed scheme surpasses that of existing approaches in attack detection rate, computation cost, communication cost, energy consumption, packet loss rate (PLR), and latency. The unicast communication and one-way-hash authentication mechanisms incorporated in the proposed model make it highly suitable

for practical implementations with negligible resource consumption overhead [6].

The proposed Hash-MAC-DSDV routing scheme registers MAC addresses and utilizes one-way hash authentication to transfer data in IoT-connected CPS systems. Evaluation results indicate that the proposed scheme outperforms existing approaches in terms of attack detection, energy consumption, and communication metrics. Furthermore, the unicast communication and one-way-hash authentication mechanisms incorporated in the proposed scheme make it highly suitable for practical implementations with negligible resource consumption overhead. Overall, the proposed scheme offers an effective and efficient solution for ensuring D2D authentication in multi-WSN CPS devices.

### 2.4.3 Applying MAC Address based Access Control for Securing Admin's Login Page

Authentication is a crucial process to ensure the security of web applications. However, commonly used parameters like username and password are susceptible to breaches, allowing malicious actors to carry out harmful activities such as data theft or modification, or taking over administrator services. To mitigate this, additional authentication factors, such as MAC Address-based access control, are needed. The utilization of MAC addresses as an additional authentication parameter has gained attention in recent research [4]. By incorporating the MAC address, which is a unique identifier associated with network devices, into the authentication process, an extra layer of security can be added to web applications. ARP, known as the Address Resolution Protocol, plays a crucial role in the validation process by mapping the IP address of user to their corresponding MAC address. It enables the system to establish the association between the IP address and the MAC address, facilitating accurate identification and authentication of the user's device. By incorporating the MAC address as a mandatory requirement, the existing authentication mechanism is bolstered with enhanced security measures. Unauthorized parties are effectively prevented from accessing the admin's login page unless they possess the specific MAC address that corresponds to it. Experimental findings substantiate that the use of MAC addresses in the authentication process enhances resistance against various attacks, including dictionary attacks and shoulder surfing attacks.

## *CHAPTER 2 : LITERATURE REVIEW*

The proposed method of enhancing web application security by incorporating MAC addresses as an additional authentication parameter is clearly elucidated and deemed to be a relevant and effective approach. The incorporation of Address Resolution Protocol (ARP) in the validation process, which maps the user's IP address to their associated MAC address, offers a notable advantage by introducing an additional layer of security to the authentication mechanism. The inclusion of MAC addresses in the authentication process provides resistance against potential threats such as Dictionary Attack and Shoulder Surfing Attack. As a result, the likelihood of unauthorized access to the admin's login page is significantly reduced. In summary, the proposed method is a robust solution to bolster web application security and mitigate potential security breaches.

### **2.5 Summary**

This chapter presented the fundamental components that will serve as the foundation for the forthcoming research. Specifically, it examines the challenges associated with the cryptographic keys storage, the applications of key derivation functions and the use of symmetric keys. Additionally, the role of MAC Addresses in the context of authentication and access management schemes is explored. The upcoming chapter will elaborate on the proposed research methodology that will be employed to address these key issues.

## SYSTEM DESIGN

This chapter provides a comprehensive discussion of the proposed system, which utilizes a symmetric key generated using the Media Access Control (MAC) address of the device for symmetric encryption and decryption in both one-to-one and group communication scenarios. The chapter is divided into two main sections, as shown in the Figure 5, namely one-to-one communication and group communication. Despite the use of different encryption schemes, both communication setups rely on a common underlying factor of key generation.

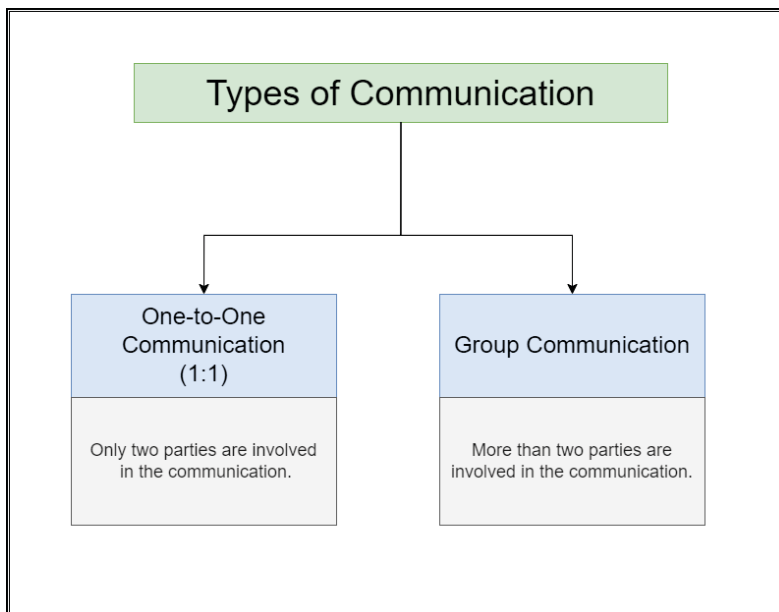


Figure 5 : Types of Communication in the Proposed System

Additionally, a seed value is generated using the MAC Address and employed as an input value for a random number generator and symmetric encryption/decryption in one-to-one communication. To enable secure group communication, a Key Distribution Center (KDC) has been developed, which generates a master key based on input keys from communicating parties. This master key can then be used for communication between the two parties employing any symmetric encryption scheme. Specifically, in the proposed scheme, the Advanced Encryption Standard (AES) 256 is utilized for this purpose.

### 3.1 One-to-One Communication

The proposed scheme involves the generation of a symmetric key using the MAC address of a device. The derivation of this key involves the utilization of the Password-Based Key Derivation Function 2 (PBKDF2), which incorporates multiple security measures including the number of iterations, key length, and a unique salt value. These parameters can be customized based on the specific requirements and system demands of the device. In addition to key generation, the scheme also extracts a seed value from the MAC address and employs a random number generator based on existing literature [3]. The bit size of the random number generator as well as the symmetric encryption and decryption processes can be tailored to match the available resources of the device. For resource-constrained devices, smaller bit sizes are recommended, while larger bit sizes are suitable for more powerful devices [38]. This customization allows for optimal utilization of the device's capabilities while maintaining the desired level of security. To perform symmetric encryption, the key, bit size, message, and seed are passed to a suitable encryption algorithm that incorporates a pseudorandom number generator for added security. The encrypted message generated by the algorithm has a length equivalent to the user-defined bit size. For decryption, the key, bit size, encrypted message, and seed are passed to a symmetric decryption algorithm that also includes the same pseudorandom number generator. This decryption algorithm produces the original message as an output. Figure 6 shows an in depth view of one-to-one communication.

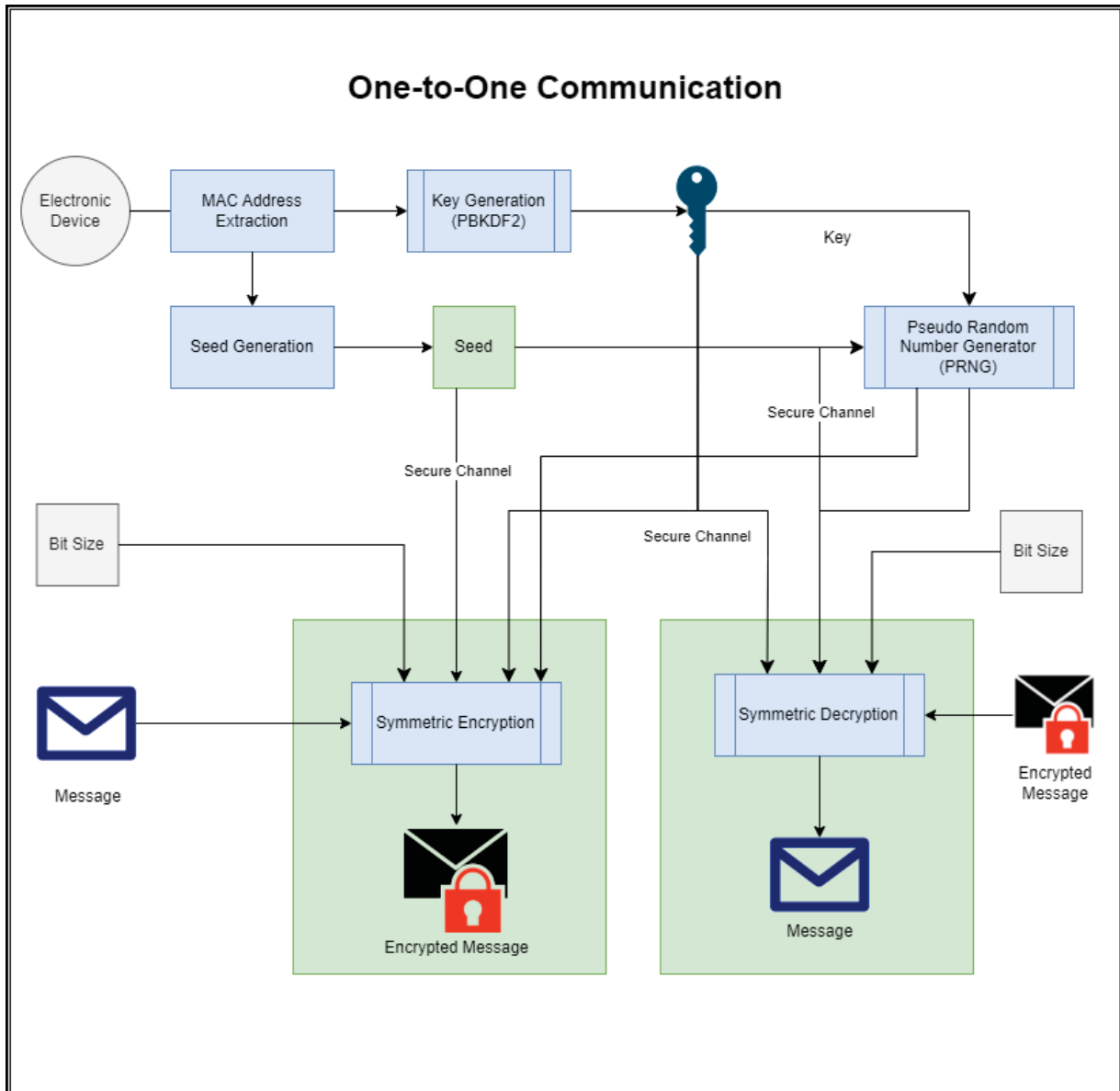


Figure 6 : One-to-One Communication Overview

### 3.1.1 Key Generation

The proposed scheme starts with the generation of a key using a MAC address. It utilizes the PBKDF2 for generating a symmetric key based on the MAC address of the device. In this approach, the MAC address of the device serves as a password, which is combined with a salt value and passed through a hash function. The output is then iterated over multiple rounds to derive the final key. PBKDF2 offers several benefits such as its resistance to brute-force attacks, the ability to customize the number of iterations and the key length, and the use of

a salt value to protect against precomputed attacks. These features make the generated key more secure and unique to the device. Additionally, by using the MAC address as a password, the generated key is specific to the device and cannot be used on other devices. By incorporating PBKDF2 in the key generation process, the proposed scheme provides a strong and secure method for generating symmetric keys that are resistant to various attacks. Figure 7 shows the process of key generation using MAC address.

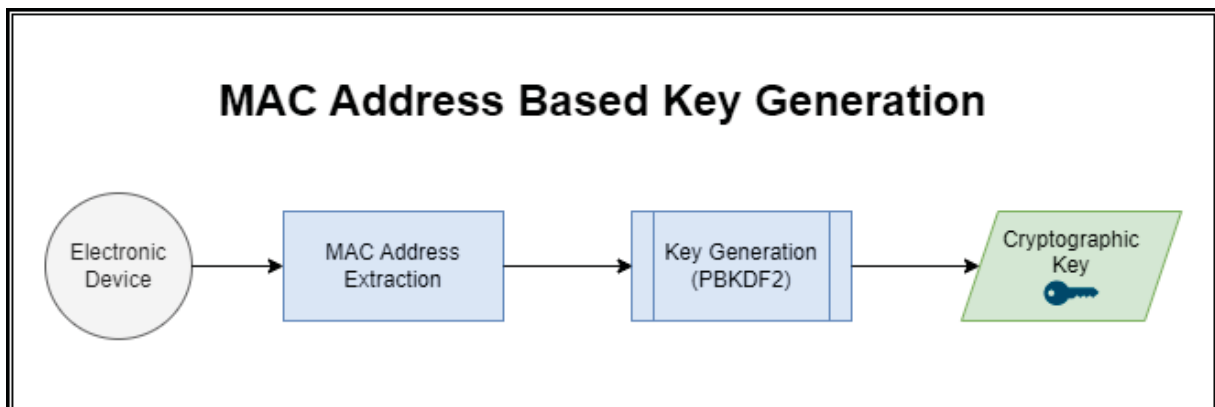


Figure 7 : Key Generation

### 3.1.2 Seed Generation

The second step of the scheme includes seed generation using the MAC address of the device. To generate the seed value, the MAC address is first converted into an integer. As the first 8 bits of the MAC address contain constant manufacturer information, they are not suitable for generating a unique seed value. Therefore, these bits are discarded to create a unique seed value that is specific to the device. The remaining bits of the MAC address are then utilized to generate a seed value that can be used in a random number generator. This method generates a unique seed value that is specific to the device, making it more secure and resistant to attacks. By using the MAC address to generate the seed value, the proposed scheme ensures that the seed value is unique to each device, and the generated random number is specific to that device. This step is explained in Figure 8.



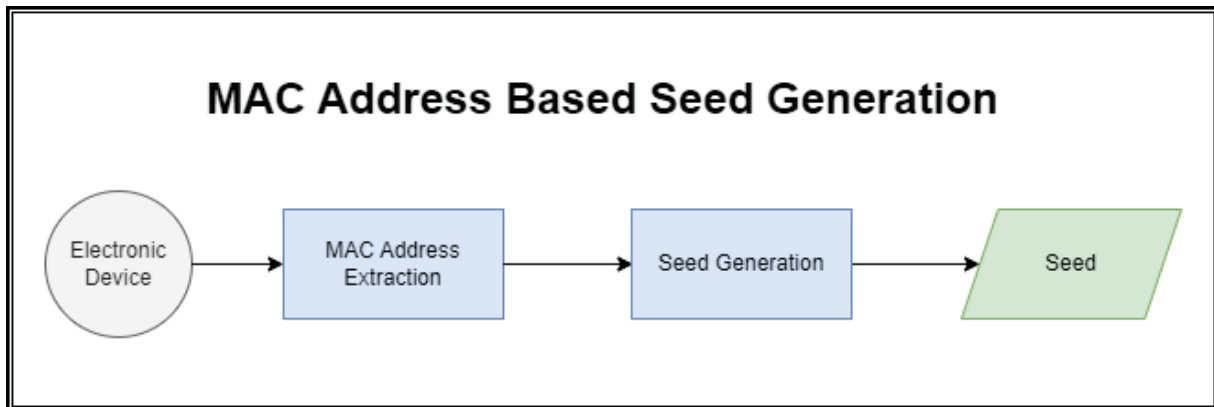


Figure 8 : Seed Generation

### 3.1.3 Pseudo Random Number Generator (PRNG)

To enhance the security of the encryption process, a PRNG is incorporated. The PRNG generates a sequence of numbers that appear to be random and unpredictable, which is essential for cryptographic operations. The PRNG takes input of the secret symmetric key generated using the MAC address, the seed value extracted from the MAC address, and the bit size specified by the user. The algorithm used in the PRNG includes features such as the SHA256 cryptographic hash function, XOR operator, AND operator which is used to generate pseudo-random sequences. These features provide high levels of security to the generated sequences. Additionally, the proposed scheme borrows some features of the PRNG from an earlier proposed scheme in the literature [3], but the secret key generation and seed value extraction from the MAC address are novel components that have been introduced in this scheme. The combination of these components enhances the overall security of the PRNG [50]. The final output of the PRNG is an integer value that is used as a security-enhancing factor in symmetric encryption and decryption. Figure 9 shows the process of generating the final output of the PRNG.

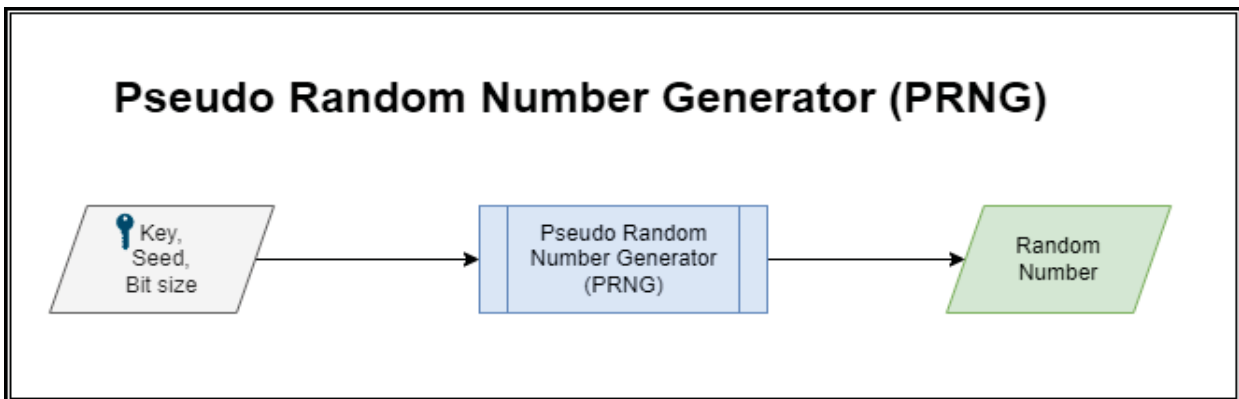


Figure 9 : PRNG

### 3.1.4 Symmetric Encryption

The proposed scheme employs symmetric encryption as the chosen method to ensure secure communication. The encryption process involves the generation of multiple private keys within the scheme, which are used to encrypt the message. The generation of private keys is achieved using the PRNG, which takes the secret symmetric key, seed, bit size, and other parameters as inputs. The generated private keys are circularly shifted to add more randomness to the encryption process. Circular shift operations are performed using bitwise operations, such as XOR and AND, to achieve this. The final output of the encryption process is an encrypted message, which is obtained by performing an XOR operation between the plaintext message and the generated private keys. The length of the encrypted message is the same as the bit size provided by the user. The encryption process is depicted in Figure 10, where the plaintext message is encrypted using the private keys generated by the PRNG.

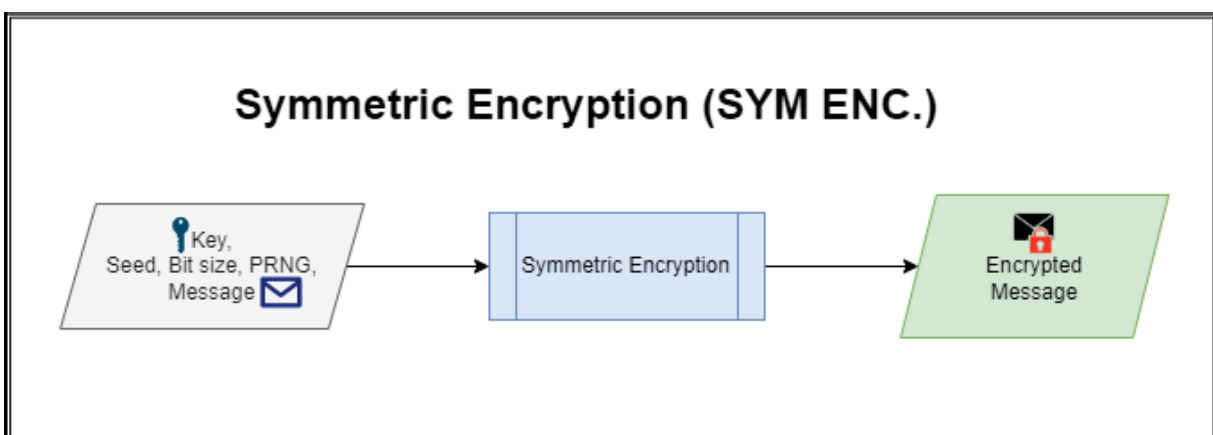


Figure 10 : Symmetric Encryption

### 3.1.5 Symmetric Decryption

Symmetric decryption is the reverse process of symmetric encryption. It involves taking an encrypted message, typically in the form of ciphertext, and applying a decryption algorithm using the same secret key that was used for encryption. The decryption algorithm reverses the encryption process, transforming the ciphertext back into the original plaintext form. In the proposed scheme, the decryption process uses the same secret key, seed, and bit size as the encryption process. The encrypted message is first passed through a decryption algorithm, which generates a random number using the PRNG with the same parameters as the encryption process. This random number is then used to create a series of private keys, which are applied to the encrypted message using circular right shift operations to obtain the original message. The decryption process is shown in Figure 11.

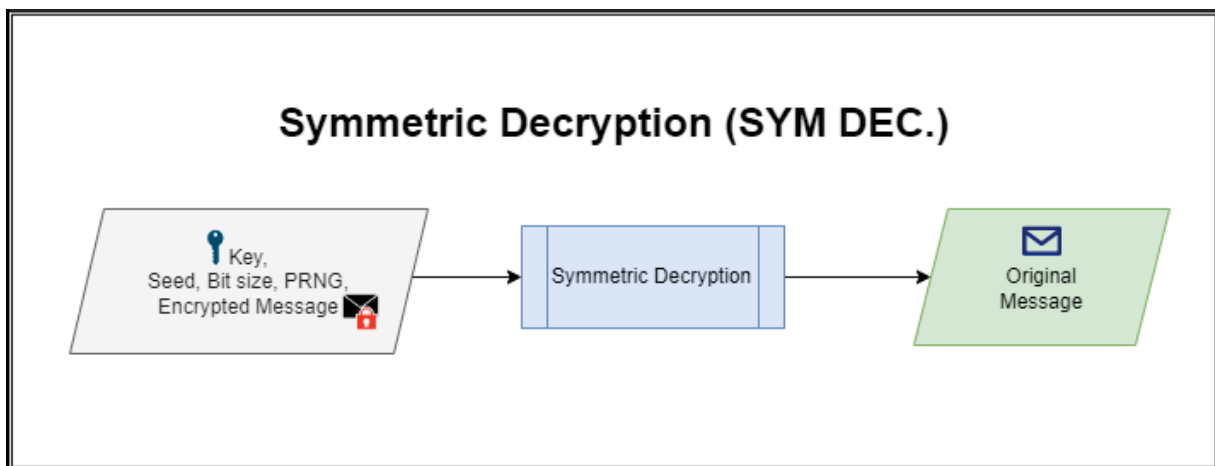


Figure 11 : Symmetric Decryption

It is worth noting that the circular right shift operation, also known as a right rotation, is indeed the inverse operation of the circular left shift operation used in the encryption process. This means that applying the same private keys in reverse order and using circular right shift instead of circular left shift will undo the encryption and recover the original plaintext message. The decryption process in the proposed scheme is designed to be fast and efficient, while still providing a high level of security.

### 3.1.6 Secure Communication between Two Parties using Secret Key in One-to-One Communication

In the proposed system design, secure communication between two parties is achieved through the utilization of the Secret Key (*SK*) generated via the Password-Based Key Derivation Function 2 (PBKDF2) and Symmetric Encryption. PBKDF2 is employed to derive a strong and unique *SK* from a given password or passphrase, applying a computationally intensive process that enhances the security of the derived key. This derived *SK* is then used in conjunction with Symmetric Encryption algorithms, which employ the same key for both encryption and decryption, ensuring confidentiality and integrity of the communication. The system design, along with the detailed process, can be observed in the schematic diagram depicted in the Figure 12 below.

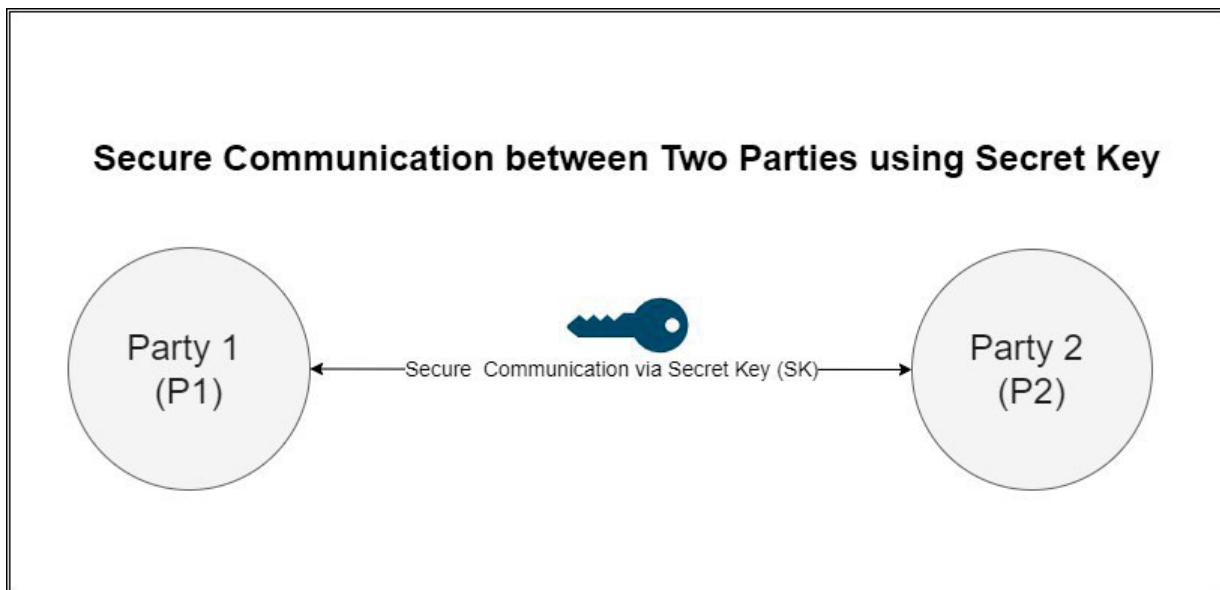


Figure 12 : Secure Communication between Two Parties using Secret Key

## 3.2 Group Communication

To enable secure group communication, the three communicating parties, Party1 ( $P1$ ), Party2 ( $P2$ ) and Party3 ( $P3$ ), first agree to share their MAC address-based keys with a trusted third party, known as the Key Distribution Center (KDC). The KDC then uses the HMAC-based Extract-and-Expand Key Derivation Function (HKDF) to combine these two keys,  $K1$  and  $K2$ , and generate a master key ( $Km$ ). This key is then transferred to all parties,  $P1$ ,  $P2$ , and  $P3$ , to enable secure communication. One of the main benefits of using HKDF is that it enables the creation of strong keys that are unique for each party, even when the input keys are weak or have low entropy. Additionally, HKDF is resistant to various types of attacks, including brute force attacks and side-channel attacks, due to its use of hash functions and salt values. Finally, HKDF allows for key expansion, enabling parties to generate additional keys from the master key as needed, which can improve the security of the overall system. Although any symmetric key algorithm can be used to encrypt and decrypt messages in this scheme, the Advanced Encryption Standard (AES) has been chosen to be used for its proven security and efficiency. Figure 13 depicts the process of group communication in the proposed scheme.

It is important to highlight that the underlying factor of key generation using MAC address and PBKDF2 in the group communication scheme is the same as in the one-to-one communication scheme. The distinguishing factor lies in how the keys are distributed among the communicating parties.

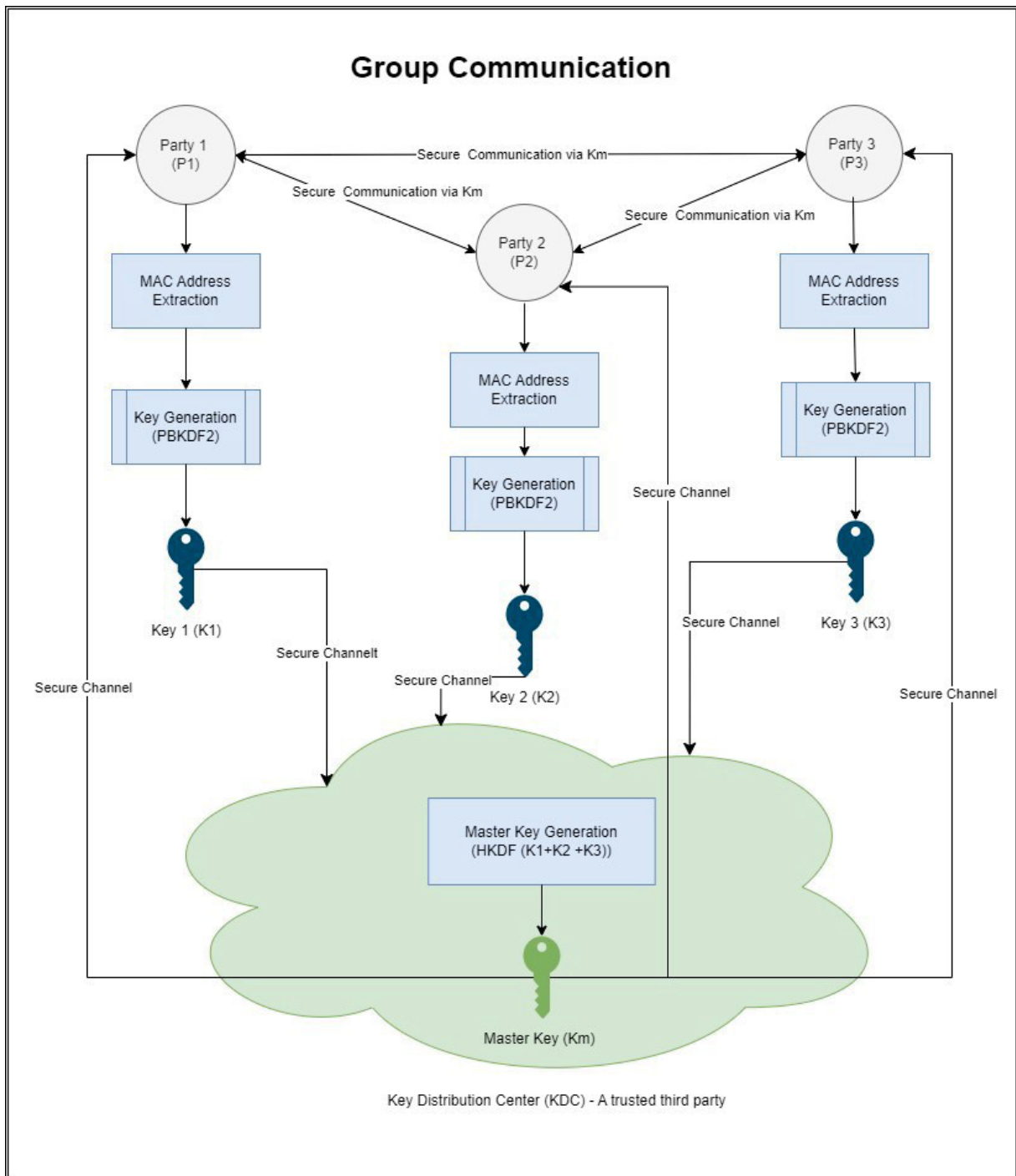


Figure 13 : Group Communication

### 3.2.1 Secure Communication between Group using Master Key in Group Communication

In the proposed system design, secure communication among three or more parties is facilitated through the utilization of the Master Key ( $K_m$ ), which is generated and shared by a KDC. The Master Key is derived and established using the HKDF and secured through the AES algorithm. HKDF ensures the generation of a strong and cryptographically secure Master Key by extracting pseudorandom keying material from the initial secret key material. AES is then employed to encrypt and decrypt the communication, leveraging the Master Key as the shared secret between the parties. The detailed process along with the interaction among the parties, can be observed in the schematic diagram depicted in the Figure 14 below.

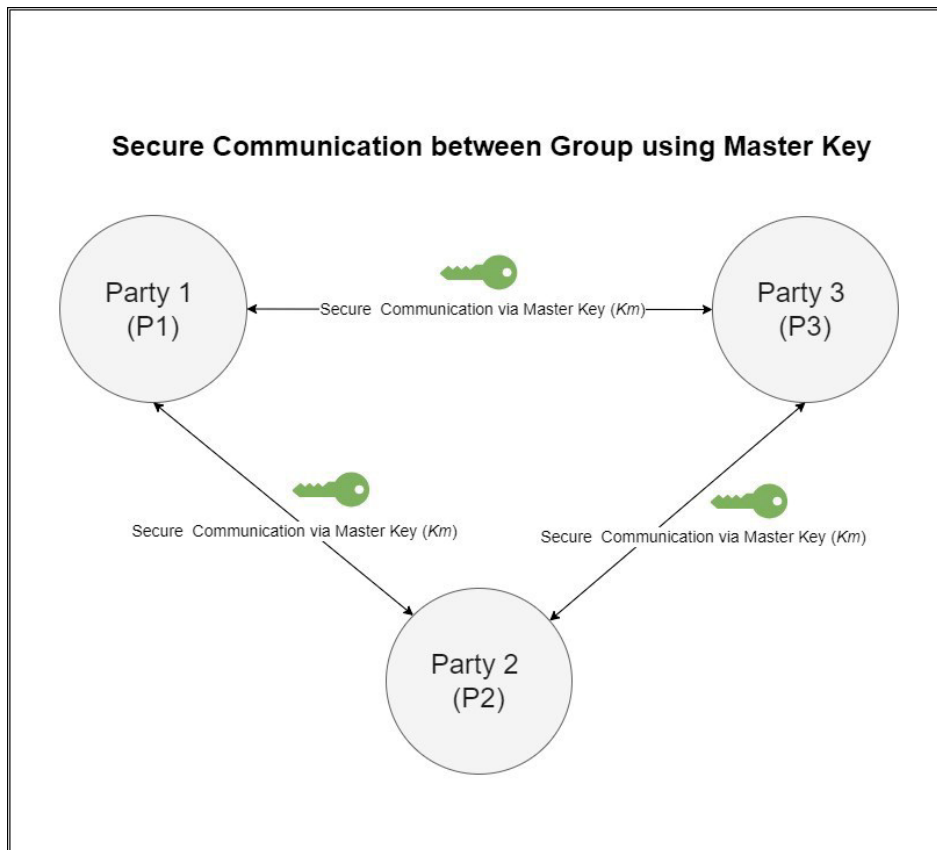


Figure 14 : Secure Communication between Group using Master Key

The utilization of the Master Key ( $K_m$ ) in the proposed system design allows for efficient and secure communication among multiple parties. By leveraging the HKDF and the encryption capabilities of AES, the system ensures that the communication remains confidential and protected against unauthorized access or tampering. The Master Key serves as a shared secret, enabling seamless and robust communication between the involved parties. This approach provides a strong foundation for secure multi-party communication in various scenarios, including collaborative environments, distributed systems, or group-based cryptographic protocols.

### 3.3 Design Assumptions

The proposed scheme makes several assumptions to ensure its security and functionality. The assumptions are as follows:

- The MAC addresses used for key generation are unique and cannot be spoofed or altered by an adversary.
- The key distribution center (KDC) is trusted and can securely distribute the master key to all parties.
- The symmetric encryption and decryption algorithms used are secure and cannot be easily broken by an adversary.
- The shared secret keys used in group communication are kept confidential and not shared with any unauthorized parties.
- The devices used for communication are not compromised by malware or other security vulnerabilities.
- The communication channels used for transmission of messages are secure and cannot be intercepted by an adversary.
- The parties involved in group communication are authentic and authorized to communicate with each other.
- The PRNG used for key generation is secure and not subject to any statistical



weaknesses or predictability.

- There is no man-in-the-middle (MITM) adversary intercepting the communication.

### 3.4 Summary

In this chapter, a novel scheme for secure communication using mac address-based key generation, seed generation, PRNG, symmetric encryption, and HKDF for group communication is proposed. Each step is explained in detail and highlighted the benefits of each component. The presented framework relies on well-established cryptographic techniques and protocols such as PBKDF2, SHA256, circular shift operations, and AES, while incorporating new elements and modifications to enhance security and efficiency.

The assumptions of the proposed scheme, including the assumption of no MITM have also been discussed. In the next chapter, the focus will be on the implementation details of the proposed scheme and provide examples of how to use it in different scenarios.

# IMPLEMENTATION, TESTING AND SECURITY ANALYSIS

This chapter focuses on the implementation of the proposed approach. Several cryptographic libraries in Python have been utilized and the implementation is carried out on the Google Colab environment. The chapter also provides an overview of the conducted test cases to assess the performance and efficiency of the approach. The focus has been on ensuring the efficiency and robustness of the scheme, and to this end. The scheme has been tested extensively for varying inputs sizes and types, varying bit sizes, and varying key sizes. Additionally, the chapter presents the testing results and a comprehensive security analysis of the approach, evaluating its resilience against various security attacks. Finally, the chapter concludes with a summary of the findings.

## 4.1 Implementation

The implementation of the methodology proposed in this study involves the use of several cryptographic libraries in Python, including the cryptography and hashlib libraries. The implementation process consisted of several steps, including the generation of keys based on MAC addresses, key distribution through a trusted third party, encryption and decryption of messages using symmetric encryption schemes and the use of HKDF to generate a master key. Each step of the implementation was carefully designed to ensure the efficiency, security, and

robustness of the scheme.

### 4.1.1 Environment

The implementation of the proposed methodology is conducted within the Google Colab environment. The environment ran on a Linux operating system with a 5.10.147+ kernel. The hardware configuration consisted of an x86\_64 CPU with 13GB of RAM and 108GB of disk space. The system specifications are as follows:

OS: Linux 5.10.147+

CPU: Intel(R) Xeon(R) CPU @ 2.20GHz (x86\_64)

RAM: 13 GB

Disk Space: 108 GB

The Python version installed on the system is Python 3.9.16. The implementation utilized several cryptographic libraries in Python, including the uuid, cryptography, os, hashlib, binascii, hmac, and pycryptodome. The version number of the library cryptography is 40.0.1. Additionally, the hashlib library was used for hashing purposes. The version number for os, hashlib, binascii, hmac, uuid libraries used in the implementation are not specified, as these are prior integrated Python libraries. The usage of all libraries has been explained in Table 1

Library Name	Usage/ Purpose
Uuid	Extraction of MAC Address
Os	To generate random salts that can be used in PBKDF2 and HKDF for secure key generation
Hashlib	Used in PRNG for string hashing
Binascii	For type conversions
cryptography	To use KDFs and generate key
pycryptodome	To use KDFs and generate key
hmac	For AES implementation

Table 1 : Libraries and Their Usage

### 4.1.2 Implementation of One-to-One Communication Scheme

To begin with, the first step in the implementation involves extracting the MAC Address of the device. This MAC Address serves as the input for generating a symmetric key using the PBKDF2. PBKDF2 offers various security features such as number of iterations, key length, and a salt value. A seed value is extracted from the MAC address and a pseudorandom number generator is employed [3] [45]. The encryption algorithm used for symmetric encryption incorporates a pseudorandom number generator for added security, and the encrypted message produced by the algorithm has a length equivalent to the user-defined bit size. For decryption, the key, bit size, encrypted message, and seed are passed to a symmetric decryption algorithm that also includes the same pseudorandom number generator. Finally, this decryption algorithm produces the original message as an output. The bit size of the encryption/decryption process and random number generator can be tailored to fit the device's resources and system requirements.

<b>One-to-One Communication Scheme</b>
<p><b>Input:</b></p> <ul style="list-style-type: none"><li>- user-defined bit size (bit_size)</li><li>- message to be encrypted (message)</li></ul>
<p><b>Output:</b></p> <ul style="list-style-type: none"><li>- encrypted message (ciphertext) from SYMENC</li><li>- decrypted message (message) from SYMDEC</li></ul>
<p><b>Process:</b></p> <ol style="list-style-type: none"><li>1. Extract the MAC Address (mac_addr) and generate a symmetric key using PBKDF2 with mac_addr as the password, a salt value, and a user-defined number of iterations and key length. secretkey = PBKDF2(mac_addr, salt, iterations, key_length)</li><li>2. Extract a seed value from the MAC address. seed = extract_seed(mac_addr)</li></ol>

```
3. Creation of a pseudorandom number generator using the seed
value and secret key to generate a pseudo random number.
    random_number = genPRNG(seed, secretkey, bit_size)

4. Encrypt the message using a symmetric encryption algorithm,
with the generated secretkey, bit_size, message, and random
number as inputs.
    ciphertext = SYMENC(secretkey, bit_size, message,
random_number)

5. Return the encrypted message (ciphertext).

6. Decrypt the message using a symmetric decryption algorithm,
with the generated secretkey, bit_size, ciphertext, and random
number as inputs.
    ciphertext = SYMDEC(secretkey, bit_size, ciphertext,
random_number)

7. Return the decrypted message (message).
```

### 4.1.3 Implementation of Group Communication Scheme

Party 1 ( $P1$ ), Party 2 ( $P2$ ), and Party 3 ( $P3$ ), agree to share their MAC address-based keys with the KDC. The KDC then uses the HKDF to combine these two keys and generate a master key,  $Km$ . The  $Km$  is then shared with all parties, Party 1 ( $P1$ ), Party 2 ( $P2$ ), and Party 3 ( $P3$ ), to enable secure communication. This approach allows for the creation of strong keys that are unique for each party, which enhances the overall security of the system. Moreover, HKDF offers additional security benefits by being resistant to various types of attacks, including brute force attacks and side-channel attacks. Finally, the AES algorithm is used for encryption and decryption in this scheme due to its proven security and efficiency. Overall, this group communication scheme provides a secure and efficient way for multiple parties to communicate with each other in a confidential and reliable manner.

### Group Communication Scheme

#### Key distribution phase:

1. Extract the MAC Address (`mac_addr1`, `mac_addr2`, `mac_addr3`) of all parties and generate a symmetric key using PBKDF2 with `mac_addr1`, `mac_addr2` and `mac_addr3` for party 1, party 2 and party 3 respectively as the password, a salt value, and a user-defined number of iterations and key length.

```
K1 = PBKDF2(mac_addr1, salt1, iterations, key_length)
```

```
K2 = PBKDF2(mac_addr2, salt2, iterations, key_length)
```

```
K3 = PBKDF2(mac_addr3, salt3, iterations, key_length)
```

2. Share Party 1, Party 2, and Party 3 keys with the Key Distribution Center (KDC)

```
send(K1, KDC)
```

```
send(K2, KDC)
```

```
send(K3, KDC)
```

3. The KDC combines the two keys using HKDF and generates a master key

```
Km = generate_master_key(K1, K2, K3)
```

4. The KDC sends the master key to Party 1 and Party 2

```
send(Km, Party1)
```

```
send(Km, Party2)
```

```
send(Km, Party3)
```

#### Secure communication phase:

1. Party 1 encrypts a message using the master key and sends it to Party 2 and Party 3

```
message = "Hello, Party 2 and 3!"
```

```
ciphertext = aes_encrypt(message, Km)
```

```
send(ciphertext, Party2)
```

2. Party 2 and Party 3 decrypt the message using the master key

```
plaintext = aes_decrypt(ciphertext, Km)
print(received_plaintext)
```

## 4.2 Testing

Testing holds paramount importance in the domain of software development, as it helps to ensure that the system is functioning correctly and meeting the required specifications. In the context of the secure communication scheme, testing is particularly important in assessing the system's performance and security aspects. A comprehensive set of tests has been carried out to assess the scheme's speed, accuracy, and performance, as well as its ability to withstand various attacks. Specifically, the scheme is tested against varying key sizes and input sizes of the plain text to determine the impact on the system's overall performance and security. The results of these tests are presented below, along with a discussion of their implications for the scheme's security and functionality.

It is worth emphasizing that the performance of the system the scheme can be impacted by other factors, such as the network, hardware and software used in the testing environment. Therefore, further testing and optimization may be required for optimal performance on different platforms. Additionally, network latency has also been included in the testing.

### 4.2.1 Varying Key Sizes

The performance of the proposed system has been evaluated for varying key sizes in both one- to-one and group communication. The key generation time, encryption and decryption time have been tested. The scheme has been tested for different key sizes, including 32, 64, 128, 256, 512, 1024, 2048, 4096, and 8192 bits. The key generation time and encryption/decryption time increased with an increase in key length. The plaintext used was "This is a plain text," with a constant bit size of 32.

In one-to-one communication, the scheme worked well for key lengths up to 1024 bits. However, there were size issues in key lengths above 2048 bits, but the keys were generated successfully. For the key length 2048 and above, the message could not be

CHAPTER 4 : IMPLEMENTATION, TESTING AND SECURITY ANALYSIS

encrypted/decrypted since it had string conversion issues. Table 2 provides the testing results of one-to-one communication for different key sizes.

Key Size (bits)	Key Generation Time (s)	Encryption Time (s)	Decryption Time (s)
32	0.016122	0.010652	0.011142
64	0.023805	0.013807	0.012951
128	0.050168	0.017721	0.023378
256	0.066142	0.042692	0.031848
512	0.124463	0.063555	0.062464
1024	0.514248	0.07052	0.067797
2048	0.494393	Integer String Conversion Issues - Exceed the limit	Integer String Conversion Issues - Exceed the limit
4096	0.369251	Integer String Conversion Issues - Exceed the limit	Integer String Conversion Issues - Exceed the limit
8192	2.642268	Integer String Conversion Issues - Exceed the limit	Integer String Conversion Issues - Exceed the limit

Table 2 : Key Generation Testing For One-to-One Communication

For group communication, the testing results showed that the scheme worked well with no significant issues in key generation or encryption/decryption time for key sizes up to 4096 bits. But for key size 8192 bits, an error message appeared: "Cannot derive keys larger than 8160 octets." This error indicated that the scheme was not capable of handling key sizes larger than 8160 octets. This suggests that the HKDF used in the scheme has a limitation on the key size it can derive. The plaintext and the bit size used was the same as in the one-to-one communication testing. Table 3 presents a detailed comparison of master key generation time



CHAPTER 4 : IMPLEMENTATION, TESTING AND SECURITY ANALYSIS

and encryption/ decryption.

Key Size (bit)	Master Key Generation Time (s)	Encryption Time (s)	Decryption Time (s)
32	0.001318	0.000368	0.000169
64	0.001421	0.000945	0.000302
128	0.001609	0.000732	0.000231
256	0.001654	0.001224	0.000185
512	0.002012	0.001453	0.000203
1024	0.003843	0.001743	0.000154
2048	0.006253	0.001935	0.000193
4096	0.009325	0.004109	0.000614
8192	Cannot derive keys larger than 8160 octets.	—	—

Table 3 : Key Generation Testing For Group Communication

The testing results revealed that the scheme worked well for key sizes up to 1024 bits in one- to-one communication and up to 4096 bits in group communication. The testing results also indicated that network latency had a significant impact on the scheme's performance. Therefore, it is recommended to consider network latency when using the scheme for large group. It is also essential to keep in mind that the scheme's performance may vary with different plaintext sizes and input lengths. Overall, the testing results validated the effectiveness and efficiency of the scheme against varying key sizes for both one-to-one and group communication.

### 4.2.2 Varying Input Size

To test the performance of the encryption and decryption process for varying input sizes, a plaintext of different sizes ranging from 100 bytes to 1000 KB has been used. The input consisted of all possible symbols, small case, upper case, spaces, and other characters. The key size was kept constant at 32 bits for both one-to-one communication and group communication schemes. For each input size, the encryption and decryption process 10 times to get an average time in which network latency is also included.

For both one-to-one and group communication, the results showed that the encryption and decryption times increased as the input size increased. It is also observed that the key generation time remains constant across all input sizes, as it is only dependent on the key size and not the input size. Table 4 and Table 5 along with graphs present the encryption and decryption time for both one-to-one and group communication respectively.

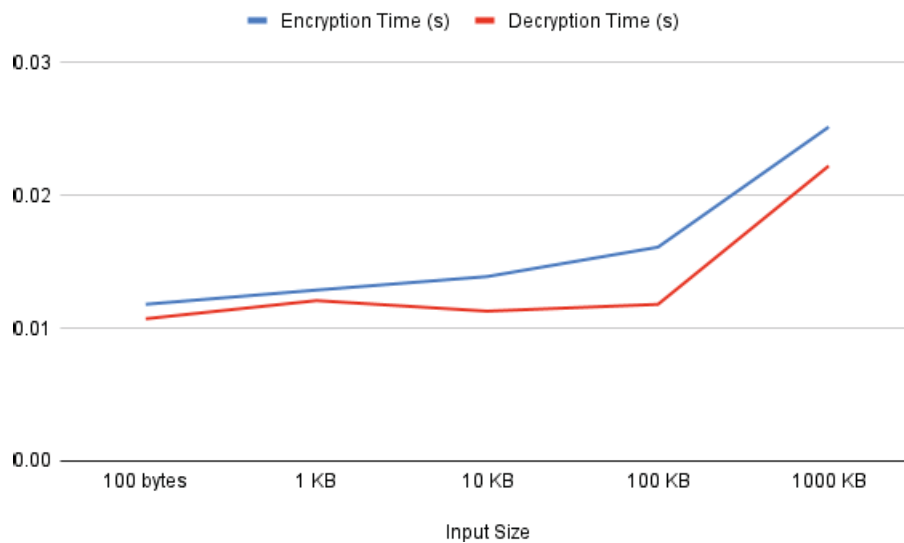


Figure 15 : Varying Input Sizes for One-to-One Communication - Encryption Time(s) and Decryption Time(s)

CHAPTER 4 : IMPLEMENTATION, TESTING AND SECURITY ANALYSIS

Input Size	Encryption Time (s)	Decryption Time (s)
100 bytes	0.011803	0.010712
1 KB	0.012875	0.012081
10 KB	0.013892	0.011285
100 KB	0.016107	0.011799
1000 KB	0.025161	0.022228

Table 4 : Input Size Testing For One-to-One Communication

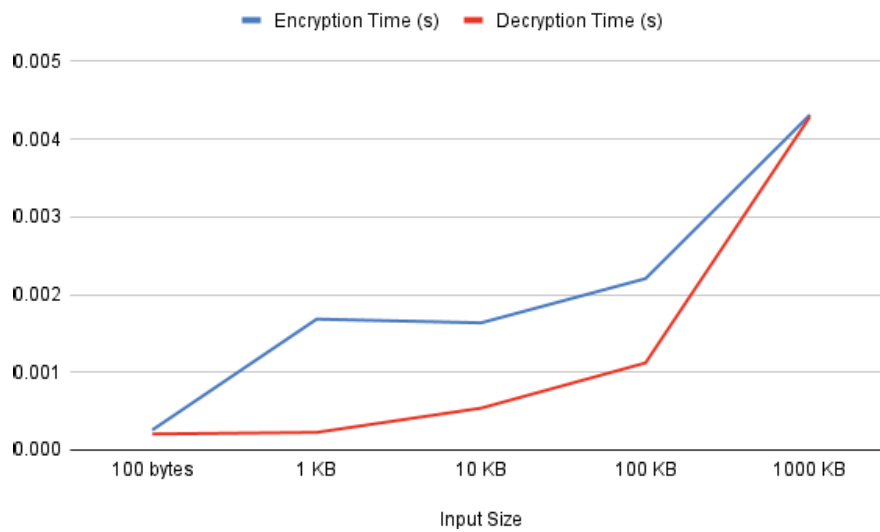


Figure 16 : Varying Input Sizes for Group Communication - Encryption Time(s) and Decryption Time(s)

Input Size	Encryption Time (s)	Decryption Time (s)
100 bytes	0.000257	0.000206
1 KB	0.001684	0.000226
10 KB	0.001636	0.000538
100 KB	0.002205	0.001120

1000 KB	0.004313	0.004288
---------	----------	----------

Table 5 : Input Size Testing For Group Communication

It has been noted that the encryption scheme performed well for plaintext of varying input sizes and key sizes. However, the time taken for encryption and decryption increased linearly with an increase in plaintext size and key size, and network latency also affected the performance of the scheme. Despite these limitations, the scheme was able to encode and decode the plaintext accurately, including all symbols, small case, upper case, spaces, and other characters, with no data loss or corruption.

To summarize, the testing results have confirmed the effectiveness and efficiency of the scheme across a range of key sizes and input sizes. However, the results have also underscored the importance considering factors such as plaintext size, and input length, and network latency, when evaluating the scheme's performance.

It's worth noting that the performance of the scheme can be influenced by several other factors, including the characteristics of the network, hardware configurations, and software components utilized in the testing environment. Therefore, additional testing and optimization may be necessary to achieve optimal performance across different platforms and environments.

### 4.3 Security Analysis

Security analysis is a critical component of any system design process, particularly for those systems that handle sensitive data or are expected to operate in hostile environments. In the field of information security, security analysis aims to identify and assess potential vulnerabilities, threats, and attacks that have the potential to compromise the confidentiality, integrity, and availability of a system or information. By conducting a comprehensive security analysis, system designers can better understand the potential risks associated with their design choices and develop appropriate countermeasures to mitigate those risks.

### 4.3.1 Forward Secrecy

#### **One-to-One Communication Scheme**

##### **Security Proof**

The one-to-one communication scheme uses a pseudorandom number generator to generate a random number, which is then used in the encryption and decryption process. The use of a pseudorandom number generator ensures that the same random number is not used for multiple messages, thus preventing an adversary from obtaining the secret key if they were to intercept multiple ciphertexts. This provides forward secrecy as the confidentiality of future messages is protected even if the secret key is compromised in the future. The purpose of this security proof is to ensure that it is not possible for an adversary to decode ciphertext even if the secret key is revealed. PBKDF2 is a publicly known key derivation function that is known to both parties.

##### **Setup Phase**

During the setup phase, the one-to-one communication scheme generates a secret key using PBKDF2, which is derived from the MAC address, salt value, and user-defined number of iterations and key length. This ensures that the key is unique and not easily guessable.

##### **Challenge Phase**

The Scheme uses a pseudorandom number generator to generate a random number that is unique for each message. This ensures that the same random number is not used for multiple messages, providing forward secrecy.

##### **Outcome Phase**

It encrypts the message using the generated secret key, bit size, message, and random number as inputs. The encrypted message is then returned. The decryption process uses the same inputs, including the random number generated during the challenge phase, ensuring that forward secrecy is maintained.

## **Group Communication Scheme**

### **Security Proof**

The Group Communication Scheme uses a master key that is generated by the Key Distribution Center (KDC) using the keys of both parties. The use of a master key ensures that even if one of the party's keys is compromised, the confidentiality of future messages is still protected. This provides forward secrecy.

### **Setup Phase**

The scheme generates a symmetric key using PBKDF2, which is derived from the MAC address, salt value, and user-defined number of iterations and key length. The keys of both parties are shared with the KDC, which combines them using HKDF to generate a master key. This ensures that the master key is unique and not predictable by the adversary.

### **Challenge Phase**

In the challenge phase, the group communication scheme uses the master key to encrypt the message, ensuring that the confidentiality of future messages is protected even if one of the party's keys is compromised.

### **Outcome Phase**

In the outcome phase, Party 2 decrypts the message using the master key. The use of the same master key ensures that forward secrecy is maintained even if one of the party's keys is compromised.

Overall, the one-to-one communication scheme and group communication scheme both provide forward secrecy to ensure the confidentiality of future messages, even if the secret key or one of the party keys is compromised. The use of unique and not easily guessable keys during the setup phase and the use of unique random numbers during the challenge phase ensure that forward secrecy is maintained.

### 4.3.2 Backward Secrecy

#### One-to-One Communication Scheme

##### Security Proof

The security proof of backward secrecy for the one-to-one communication scheme relies on the fact that if an adversary gains access to the secret key at any point in time after a message has been transmitted and the random number generator has been used to generate a random number, the adversary would still not be able to decrypt the message as the random number generator uses the MAC address as a seed, which is unique to the sender's device. Therefore, the adversary would not have access to the same seed value and would not be able to generate the same random number, resulting in an inability to decrypt the message.

##### Setup Phase

When a message is to be encrypted, the sender's MAC address (*mac\_addr*) is extracted and used to generate a symmetric key using PBKDF2 with *mac\_addr* as the password, a salt value, and a user-defined number of iterations and key length (*key\_length*).

$$secretkey = PBKDF2(mac\_addr, salt, iterations, key\_length)$$

The seed value is extracted from the MAC address.

$$seed = extract\_seed(mac\_addr)$$

A pseudorandom number generator is created using the seed value and secret key to generate a pseudo-random number.

$$random\_number = genPRNG(seed, secretkey, bit\_size)$$

The message is encrypted using a symmetric encryption algorithm, with the generated secretkey, bit\_size, message, and random number as inputs.

$$ciphertext = SYMENC(secretkey, bit\_size, message, random\_number)$$

##### Challenge Phase

Suppose an adversary gains access to the secret key at a later point in time, but they do not have access to the same seed value as the sender. Under such circumstances, the adversary

will encounter difficulty in generating the identical pseudo random number that was used for message encryption. Consequently, will be unable to decrypt the message successfully.

### **Outcome Phase**

Backward secrecy is achieved as an adversary who gains access to the secret key at any time after the message has been transmitted and the random number generator has been used to generate a random number will not be able to decrypt the message.

## **Group Communication Scheme**

### **Security Proof**

The security proof of backward secrecy for the group communication scheme relies on the fact that if an adversary gains access to the master key at any point in time after a message has been transmitted, they still would not be able to decrypt the message as the master key is not used to encrypt the message directly. Instead, the master key is used to derive a session key used for encryption/decryption, and if an adversary gains access to the master key, they would not have access to the session key as it is not transmitted over the network.

### **Setup Phase**

The MAC address ( $mac\_addr1$ ,  $mac\_addr2$ ) of both parties is extracted and used to generate a symmetric key using PBKDF2 with  $mac\_addr1$  and  $mac\_addr2$  for party 1 and party 2, respectively, as the password, a salt value, and a user-defined number of iterations and key length.

$$K1 = PBKDF2(mac\_addr1, salt1, iterations, key\_length)$$
$$K2 = PBKDF2(mac\_addr2, salt2, iterations, key\_length)$$

Party 1 and Party 2 keys are shared with the Key Distribution Center (KDC)

$$send(K1, KDC)$$
$$send(K2, KDC)$$



## CHAPTER 4 : IMPLEMENTATION, TESTING AND SECURITY ANALYSIS

The KDC combines the two keys using HKDF and generates a master key.

$Km = generate\_master\_key(K1, K2)$

The KDC sends the master key to Party 1 and Party 2

$send(Km, Party1)$

$send(Km, Party2)$

### Challenge Phase

Attempt to decrypt a previous session's ciphertext using the current session's master key. Since the master key is unique for each session, decryption of the previous ciphertext will fail.

### Outcome Phase

The previous session's ciphertext cannot be decrypted using the current session's master key, ensuring backward secrecy.

### 4.3.3 Spoofed MAC Address

#### Security Proof

Assuming an adversary can spoof the MAC address, the secrecy of the generated symmetric key and the confidentiality of the message may be compromised. To prove the security of the scheme, it needs to be shown that even if the adversary can spoof the MAC address, the security of the scheme remains intact.

#### Setup Phase

In the setup phase, the MAC address is extracted and used to generate the symmetric key using PBKDF2. However, if the MAC address is spoofed, the symmetric key generated will be different from the intended key. The adversary can then potentially obtain the symmetric key and use it to decrypt any encrypted messages.

### **Challenge Phase**

To test the security of the scheme against a spoofed MAC address, assume that the adversary has successfully spoofed the MAC address and obtained the symmetric key. The adversary will then try to decrypt a message encrypted with the same symmetric key.

### **Outcome Phase**

If the adversary can successfully decrypt the message, it means that the confidentiality of the message has been compromised. Therefore, the scheme does not provide secrecy against a spoofed MAC address.

The scheme is vulnerable to attacks where the MAC address is spoofed. To mitigate this vulnerability, additional measures such as message authentication codes (MACs) can be added to ensure message integrity and authenticity. Alternatively, a public key infrastructure (PKI) can be used to ensure that the symmetric keys are securely exchanged between the communicating parties.

## **4.4 Summary**

In this chapter, a discussion on the implementation and testing of two communication schemes: a one-to-one communication scheme and a group communication scheme has been presented. The one-to-one communication scheme generates a symmetric key using PBKDF2 and extracts a seed value from the MAC address to create a pseudorandom number generator. The group communication scheme uses a KDC to distribute a master key to the communicating parties.

During the testing phase, both schemes have been tested for their functionality and performance, including encryption and decryption times, message size limits, and scalability. The tests have shown that both schemes work as intended and can handle the desired message size limits and number of communicating parties.

In the security analysis, some potential security threats such as a spoofed MAC address and

#### *CHAPTER 4 : IMPLEMENTATION, TESTING AND SECURITY ANALYSIS*

a compromised KDC have been identified. The security proofs, setup phase, challenge phase, and outcome phase for forward and backward secrecy to ensure the security of both schemes have also been provided.

In the upcoming chapter, a conclusion will be provided and future works will be discussed, including possible improvements to the schemes and further security analysis to ensure their continued security in the face of evolving threats.

# CONCLUSION AND FUTURE WORKS

This chapter marks the conclusion of this research. Thus a summary of the research findings, conclusions, and possible future work has been presented. In this chapter, an overview of the proposed work, including the design and implementation of the proposed encryption schemes, the testing of the proposed schemes for accuracy and efficiency, and the security analysis of the proposed schemes against potential attacks will be provided. It will also discuss the limitations of the proposed work and suggest directions for future research. Finally, the chapter will conclude by highlighting the contributions of the proposed work and its potential impact in the field of encryption and network security.

## 5.1 Conclusion

This research has proposed a novel solution to the significant concern of key storage in cryptographic systems. The centralized storage of keys presents a risk of a one-to-one point of failure, and storing keys on devices poses a risk of key theft through physical or remote attacks. While modern solutions like PUF and TPM have inherent problems like cost and stability, the complexity of key management also increases as the number of keys grows, making it difficult to maintain the security and integrity of cryptographic systems. Additionally, the costs associated with key storage hardware and management can be significant. The proposed MAC address-based key extraction and secrecy scheme eliminates the need for stored keys and improves the security of cryptographic systems by utilizing a

## *CHAPTER 5 : CONCLUSION AND FUTURE WORKS*

device's unique MAC address to derive a symmetric key. This scheme can be used for both one-to-one and group communication and can also be integrated with modern symmetric key algorithms.

Both schemes were implemented and subjected to thorough testing, yielding positive outcomes that demonstrated their efficiency, security, and practicality in real-world scenarios. The results have shown that the proposed scheme is effective in providing backward secrecy and forward secrecy against common attacks. However, it is important to note that these schemes are not immune to all types of attacks, and further research is needed to improve their security. The security analysis has identified potential threats and provided measures to mitigate them, ensuring the overall security and integrity of the system.

The proposed MAC address based key extraction and secrecy scheme presents a significant improvement in the security of cryptographic systems. By eliminating the need for stored keys and leveraging a device's unique MAC address, this scheme provides a cost-effective and efficient solution to the complex issue of key storage and management.

### 5.2 Future Works

The presented work can make a significant contribution towards the implementation of better security services. However, the schemes are not comprehensive and there is room for further research as follows:

#### 5.2.1 MAC Address-Based Key Extraction and Secrecy Scheme

1. The current scheme extracts the key from the MAC address of the device, but other unique identifiers could also be used, such as the device's serial number or firmware version. Future research could explore the use of these other identifiers and compare their effectiveness to the MAC address.
2. The scheme could be tested with a larger number of devices to validate its scalability and performance under various network conditions.

3. The proposed scheme assumes that the MAC address cannot be spoofed, but future work could explore potential countermeasures to mitigate MAC address spoofing attacks.

### 5.2.2 Group Key Management Scheme

1. The proposed work uses a KDC to distribute keys to group members, but future work could explore alternative distribution methods, such as a decentralized approach where each member has a copy of the key and updates are broadcasted to the group.
2. The proposed scheme only supports the addition of new group members; future work could investigate the removal of members and how to handle revoked or compromised keys.
3. The scheme could be extended to support other group communication scenarios, such as multicast and broadcast, to improve its versatility.

The conclusions and future works highlight the main objectives of the research, which were to address the issues related to key management in cryptographic systems, and to propose a novel MAC address-based key extraction and secrecy scheme. The research findings indicate that the proposed scheme offers a high level of security and efficient way of generating symmetric keys without the need for stored keys, which can improve the security and reduce the cost associated with key management. It is suggested that further research can be conducted to explore the performance and security of the proposed scheme in different network environments, and to investigate the use of advanced cryptographic algorithms with the scheme.

The future work section outlines the possible research areas that can be explored in the future to enhance the security and efficiency of the proposed scheme. This includes the use of blockchain technology to improve the security and integrity of the key extraction process, and the integration of the proposed scheme with other security mechanisms to provide a

## *CHAPTER 5 : CONCLUSION AND FUTURE WORKS*

comprehensive security solution for cryptographic systems.

Overall, the conclusion and future works provide a roadmap for future research and development in the area of key management in cryptographic systems, and demonstrate the potential of the proposed MAC address-based key extraction and secrecy scheme in improving the security and efficiency of cryptographic systems.

## REFERENCES

- [1] Arora, S., & Hussain, M. (2018, September). Secure session key sharing using symmetric key cryptography. In 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 850-855). IEEE.
- [2] C. Lipps, A. Weinand, D. Krummacker, C. Fischer and H. D. Schotten, "Proof of Concept for IoT Device Authentication Based on SRAM PUFs Using ATMEGA 2560-MCU," 2018 1st International Conference on Data Intelligence and Security (ICDIS), South Padre Island, TX, USA, 2018, pp. 36-42, doi: 10.1109/ICDIS.2018.00013.
- [3] Patgiri, R. (2021). Symkrypt: A general-purpose and lightweight symmetric-key cryptography. Cryptology ePrint Archive.
- [4] B. M. Prasetya Pagar Alam, R. Septiasari and A. Amiruddin, "Applying MAC Address- Based Access Control for Securing Admin's Login Page," 2019 6th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Bandung, Indonesia, 2019, pp. 292-296, doi: 10.23919/EECSI48112.2019.8977016.
- [5] W. Xie, J. Yu and G. Deng, "A Secure DHCPv6 System Based on MAC Address Whitelist Authentication and DHCP Fingerprint Recognition," 2021 7th Annual International Conference on Network and Information Systems for Computers (ICNISC), Guiyang, China, 2021, pp. 604-608, doi: 10.1109/ICNISC54316.2021.00114.
- [6] M. Adil et al., "Hash-MAC-DSDV: Mutual Authentication for Intelligent IoT-Based Cyber-Physical Systems," in IEEE Internet of Things Journal, vol. 9, no. 22, pp. 22173- 22183, 15 Nov.15, 2022, doi: 10.1109/JIOT.2021.3083731.
- [7] Choi, H., & Seo, S. C. (2021). Optimization of PBKDF2 using HMAC-SHA2 and HMAC- LSH families in CPU environment. IEEE Access, 9, 40165-40177.
- [8] Chaeikar, S. S., Alizadeh, M., Tadayon, M. H., & Jolfaei, A. (2022). An intelligent cryptographic key management model for secure communications in distributed industrial intelligent systems. International Journal of Intelligent Systems, 37(12), 10158-10171.



## REFERENCES

- [9] Shen, W., Qin, J., Yu, J., Hao, R., Hu, J., & Ma, J. (2019). Data integrity auditing without private key storage for secure cloud storage. *IEEE Transactions on Cloud Computing*, 9(4), 1408-1421.
- [10] Y. Gao, Y. Su, W. Yang, S. Chen, S. Nepal and D. C. Ranasinghe, "Building Secure SRAM PUF Key Generators on Resource Constrained Devices," 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kyoto, Japan, 2019, pp. 912-917, doi: 10.1109/PERCOMW.2019.8730781.
- [11] Khande, R., Ramaswami, S., Naidu, C., & Patel, N. (2021). AN EFFECTIVE MECHANISM FOR SECURING AND MANAGING PASSWORD USING AES-256 ENCRYPTION & PBKDF2. *Technology (IJEET)*, 12(5), 1-7.
- [12] Vyakaranal, S., & Kengond, S. (2018, April). Performance analysis of symmetric key cryptographic algorithms. In 2018 International Conference on Communication and Signal Processing (ICCSP) (pp. 0411-0415). IEEE.
- [13] A. Bakshi, "Classical and Physical Security of Symmetric Key Cryptographic Algorithms," 2021 IFIP/IEEE 29th International Conference on Very Large Scale Integration (VLSI-SoC), Singapore, Singapore, 2021, pp. 1-2, doi: 10.1109/VLSI-SoC53125.2021.9606988.
- [14] Ali, J., & Dyo, V. (2020). Practical hash-based anonymity for mac addresses. arXiv preprint arXiv:2005.06580.
- [15] Colombo AW, Karnouskos S, Kaynak O, Shi Y, Yin S. Industrial cyberphysical systems: a backbone of the fourth industrial revolution. *IEEE Ind Electron Mag.* 2017;11(1):6-16.
- [16] Gope P, Das AK, Kumar N, Cheng Y. Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks. *IEEE Trans Ind Inf.* 2019;15(9):4957-4968.
- [17] Mandal Susmita, Mohanty Sujata, Majhi Banshidhar. CL-AGKA: certificateless authenticated group key agreement protocol for mobile networks. *Wireless Networks.* 2020;26(4):3011-3031. <http://dx.doi.org/10.1007/s11276-020-02252-z>
- [18] Yousefpoor MS, Barati H. Dynamic key management algorithms in wireless sensor

## REFERENCES

- networks: a survey. *Comput Commun.* 2019;134:52-69.
- [19] Ma M, Shi G, Li F. Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the IoT scenario. *IEEE Access.* 2019;7:34045-34059.
- [20] Li L, Xu G, Jiao L, et al. A secure random key distribution scheme against node replication attacks in industrial wireless sensor systems. *IEEE Trans Ind Inf.* 2019;16(3):2091-2101.
- [21] Yazdanpanah S, Saman SC, Mazdak Z, Reza K. Security features comparison of master key and IKM cryptographic key management for researchers and developers. In: *International Conference on Software Technology and Engineering*; 2011:365-369.
- [22] M. Ytldrtrm & I. Mackie "Encouraging users to improve password security and memorability", *International Journal of Information Security* volume 18, pages 741–759(2019)
- [23] Rossouwvon Solms Johanvan Niekerk, "From information security to cyber security", *Computers & Security*", Volume 38, October 2013, Pages 97-102
- [24] Katha Chanda, "Password Security: An Analysis of Password Strengths and Vulnerabilities", *I. J. Computer Network and Information Security*, 2016, 7, 23-30
- [25] M. Hiller, "Key derivation with physical unclonable functions," Ph.D. dissertation, University of Munich, 2016.
- [26] Viktor Taneski, Marjan Hericcko, BoStjan Brumen, "Systematic Overview of Password Security Problems", *Acta Polytechnica Hungarica*, Vol. 16, No. 3, 2019
- [27] Y. Gao, H. Ma, S. F. Al-Sarawi, D. Abbott, and D. C. Ranasinghe, "PUFFSM: A controlled strong PUF," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 64, pp. 2532–2543, 2017.
- [28] G. T. Becker, "The gap between promise and reality: On the insecurity of XOR Arbiter PUFs," in *Cryptographic Hardware and Embedded Systems (CHES)*. Springer, 2015, pp. 535–555.

## REFERENCES

- [29] J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede, "Helper data algorithms for PUF-based key generation: Overview and analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 6, pp. 889–902, 2015.
- [30] J. Delvaux, "Security analysis of PUF-based key generation and entity authentication," Ph.D. dissertation, University of KU Leuven and ShangHai Jiao Tong University, 2017.
- [31] Y. Gao, Y. Su, L. Xu, and D. C. Ranasinghe, "Lightweight (reverse) fuzzy extractor with multiple reference puf responses," *IEEE Transactions on Information Forensics and Security*, 2018.
- [32] J. Delvaux and I. Verbauwhede, "Attacking PUF-based pattern matching key generators via helper data manipulation," in *Topics in Cryptology–CT-RSA*. Springer, 2014, pp. 106–131.
- [33] W. Shen, J. Yu, H. Xia, H. Zhang, X. Lu, and R. Hao, "Light-weight and privacy-preserving secure cloud auditing scheme for group users via the third party medium," *J. Netw. Comput. Appl.*, vol. 82, pp. 56–64, 2017.
- [34] C. Guan, K. Ren, F. Zhang, F. Kerschbaum, and J. Yu, "Symmetrickey based proofs of retrievability supporting public verification," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2015, pp. 203–223.
- [35] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 598–609.
- [36] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: Security and privacy concerns," *IEEE Secur. Privacy*, vol. 1, no. 2, pp. 33–42, Mar. 2003.
- [37] A. Fu, S. Yu, Y. Zhang, H. Wang, and C. Huang, "NPP: A new privacy-aware public auditing scheme for cloud data sharing with group users," *IEEE Trans. Big Data*, 2017. [Online]. Available: doi:10.1109/TBDDATA.2017.2701347.
- [38] J. Li, L. Zhang, J. K. Liu, H. Qian, and Z. Dong, "Privacy-preserving public auditing protocol for low-performance end devices in cloud," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 11, pp. 2572–2583, Nov. 2016.
- [39] K. M. Moriarty, B. Kaliski, and A. Rusch, *PKCS #5: Password-Based*

## REFERENCES

- Cryptography Specification Version 2.1, document RFC 8018, 2017, pp. 1–40.
- [40] Mandal Susmita, Mohanty Sujata, Majhi Banshidhar. CL-AGKA: certificateless authenticated group key agreement protocol for mobile networks. *Wireless Networks*. 2020;26(4):3011-3031. <http://dx.doi.org/10.1007/s11276-020-02252-z>
- [41] "Specification for the advanced encryption standard (aes)," Federal Information Processing Standards Publication 197, 2001. [Online]. Available: <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
- [42] A. Aly, T. Ashur, E. Ben-Sasson, S. Dhooghe, and A. Szepieniec, "Design of symmetric-key primitives for advanced cryptographic protocols," *IACR Transactions on Symmetric Cryptology*, vol. 2020, no. 3, pp. 1–45, Sep. 2020.
- [43] A. Boldyreva, J. P. Degabriele, K. G. Paterson, and M. Stam, "Security of symmetric encryption in the presence of ciphertext fragmentation," in *Proceedings of the 31st Annual International Conference on Theory and Applications of Cryptographic Techniques*, ser. EUROCRYPT'12. Berlin, Heidelberg: Springer-Verlag, 2012, p. 682– 699. [Online]. Available: [https://doi.org/10.1007/978-3-642-29011-4\\_40](https://doi.org/10.1007/978-3-642-29011-4_40)
- [44] K. McCusker and N. E. O'Connor, "Low-energy symmetric key distribution in wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 3, pp. 363–376, 2011.
- [45] L. E. Bassham III, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks et al., SP 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications. National Institute of Standards & Technology, 2010. [Online]. Available: <https://csrc.nist.gov/publications/detail/sp/800-22/rev-1a/final>
- [46] A. Baksi, S. Bhasin, J. Breier, D. Jap, and D. Saha, "Fault attacks in symmetric key cryptosystems," *Cryptology ePrint Archive*, Report 2020/1267, 2020, <https://eprint.iacr.org/2020/1267>.
- [47] D. J. Bernstein, *The Salsa20 Family of Stream Ciphers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 84–97. [Online]. Available: [https://doi.org/10.1007/978-3-540-68351-3\\_8](https://doi.org/10.1007/978-3-540-68351-3_8)

## REFERENCES

- [48] A. Visconti and F. Gorla, "Exploiting an HMAC-SHA-1 optimization to speed up PBKDF2," *IEEE Trans. Depend. Sec. Comput.*, vol. 17, no. 4, pp. 775–781, Jul. 2020.
- [49] Krawczyk, H., & Eronen, P. (2010). HMAC-based extract-and-expand key derivation function (HKDF) (No. rfc5869).
- [50] National Institute of Standards and Technology, "Recommendation for Key Derivation Using Pseudorandom Functions", NIST Special Publication 800-108, November 2008.
- [51] S. Raza, L. Seitz, D. Sitenkov, and G. Selander, "S3k: Scalable security with symmetric keys—dtls key establishment for the internet of things," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 3, pp. 1270–1280, 2016.
- [52] Krawczyk, H., "Cryptographic Extraction and Key Derivation: The HKDF Scheme", *Proceedings of CRYPTO 2010* (to appear), 2010, <http://eprint.iacr.org/2010/264>.
- [53] Forsberg, D., Ohba, Y., Ed., Patil, B., Tschofenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, May 2008
- [54] Ali, J. (2017). Mechanism for the prevention of password reuse through anonymized hashes. *PeerJ PrePrints*, 5:e3322v1.
- [55] Martin, J., Rye, E., and Beverly, R. (2016). Decomposition of mac address structure for granular device inference. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 78–88.
- [56] Red Hat, Inc. (2020). Generating a new unique mac address.
- [57] Blogg, M., Semler, C., Hingorani, M., and Troutbeck, R. (2010). Travel time and origin-destination data collection using bluetooth mac address readers. In *Australasian transport research forum*, volume 36.
- [58] Alimpia, J. B., Sison, A. M., and Medina, R. P. (2018). An enhanced hash-based message authentication code using bcrypt. *Proceedings of the International Journal for Research in Applied Science and Engineering Technology*, 6(4).
- [59] Ertaul, L., Kaur, M., and Gudise, V. A. K. R. (2016) Implementation and

## *REFERENCES*

- performance analysis of pbkdf2, bcrypt, scrypt algorithms. In Proceedings of the International Conference on Wireless Networks (ICWN), page 66. The Steering Committee of The World Congress in Computer Science, Computer.
- [60] Clearview Intelligence Ltd (2020). Product specification m830 traffic monitoring.

### **Certificate for Plagiarism**

It is certified that PhD/M.Phil/MS Thesis Titled "Design of a MAC Address Based Secure Communication Scheme" by ZUBARIA QURASHAI has been examined by us. We undertake the follows:

- a. Thesis has significant new work/knowledge as compared already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.
- b. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results or words of others have been presented as Author own work.
- c. There is no fabrication of data or results which have been compiled/analyzed.
- d. There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.
- e. The thesis has been checked using TURNITIN (copy of originality report attached) and found within limits as per HEC plagiarism Policy and instructions issued from time to time.

#### **Name & Signature of Supervisor**

Dr. Hasan Tahir

Signature :

