Secure PUF Based Key Generation for Large Communication Groups

By

Muhammad Taha
Fall 2016 - MS(IS) - 00000170632

Supervisor
Dr. Hasan Tahir

Department of Computing

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

In

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

(July, 2020)

DEDICATION

I dedicate this thesis to my **Parents** for their endless prayers, love, support and encouragement. I would also like to dedicate this thesis to my Late Grandmother **Razia Bano**.

Approval

It is certified that the contents and form of the thesis entitled "Secure PUF based Key Generation for Large Communicating Groups" submitted by MUHAMMAD TAHA have been found satisfactory for the requirement of the degree

Advisor : Dr. Hasan Tahir

Signature: _____

Date: 06-Jul-2020

Committee Member 1:Dr. Qaiser Riaz

Signature: _____

Date: 06-Jul-2020

Committee Member 2:Mehdi Hussain

Signature: _____

Date: 06-Jul-2020

Committee Member 3:Dr. Sana Qadir

Signature: _____

Date: 06-Jul-2020

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled "Secure PUF based Key Generation for Large Communicating Groups" written by MUHAMMAD TAHA, (Registration No 00000170632), 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: 06-Jul-2020

Signature (HOD): _____

Date: _____

Signature (Dean/Principal): _____

Date: _____

Certificate of Originality

I hereby declare that this submission titled "Secure PUF based Key Generation for Large Communicating Groups" 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: MUHAMMAD TAHA

Student Signature: 

ACKNOWLEDGEMENT

First and foremost, I would like to thank Allah, the Almighty for giving me the ability and strength to carry out this research. I would like to pay my special regards to the 14 Infallibles (A.S), due to their sadaqah I am able to carry out this research.

My deepest gratitude to my supervisor Dr. Hasan Tahir for his continuous support and guidance during my thesis. I could not have imagined having a better supervisor and mentor for my master's degree. I am also thankful to my teachers for providing me with an academic base, which enables me to complete this thesis.

I am thankful to all my fellows and friends especially Mubarak Mehdi and Zahoor Ahmed Alizai for their support and motivation.

Last but not the least, I would like to thank my parents for their endless prayers and support throughout.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ASSOCIATED PUBLICATIONS

M. Mehdi, M.T. Ajani, H. Tahir, Z. Alizai, S. Tahir, F. Khan, Q. Riaz, M. Hussain, "PUF-based Key Generation Scheme For Secure Group Communication Using MEMS" *Submitted to Neural Computing and Applications. Springer*.

ABSTRACT

Conventional cryptographic algorithms have been depended on stored keys for providing the security services. Since the keys are stored on a device, it makes them vulnerable to key theft attacks. Increasing the key size makes the brute force attack difficult but does not eliminates the threat of key theft. This thesis proposes secure key generation schemes for group communication. The research makes three major contributions to improve the security of devices in multiparty environment. The thesis also demonstrates that the novel root of trust can be used for the provision of security services. The first contribution of the thesis is the creation of a symmetric group key for group environment using Physical Unclonable Function (PUF). By embedding the novel root of trust (PUF) the threat of key theft attack is mitigated. The second contribution made in this research is the symmetric key created in contribution fashion means all the members of the group has participated in creating a key. The third contribution made in this research is the provision of a scheme that can be used for key authentication generated for the group. The proposed schemes have been tested for creating the group keys.

CHAPTER 1: INTRODUCTION

1.1 Introduction

Use of computer and other information processing devices has increased due to the recent advancements in technology. Earlier, computers were used as standalone devices for personal use or in a controlled network like in labs or offices. The evolution of internet and emergence of smart devices has caused the rapid adoption of computing devices. The usage of computers is no longer restricted to controlled environment thus, devices have penetrated more complex environment and large networks like online data sharing, e-commerce, etc. The advancement in these technologies has largely been facilitated through ease of availability of information system on mobile devices. Financial, educational, health, energy sectors, etc. have experienced significant improvement in remote access [1][2]. With the rapid increase in usage of smart devices, group communication, cyber physical system and internet of things, it is now increasingly difficult to ensure the privacy and security of data because the risk of malicious factors compromising the information system has increased. Attackers can compromise the security of the systems using multiple techniques like side channel attack [3] which can completely expose the security of the system. It is particularly painful since many of these attacks do not target the algorithm; instead they rely on channels that are often not considered vulnerable. Conventional attacks that are based on the weakness in design of algorithm are difficult to correct as they require a complete redesign of the algorithm. To achieve confidentiality of data, encryption is used that often relies on stored keys. These keys are crucial to the design of the security implementation as highlighted by Kerchoff. According to Kerchoff's security principle "The security of a system should lie in keeping the key secret and not the

algorithm"[4]. Increasing the size of key makes it difficult for attacker to brute force but it does not eliminate the risk of key theft through other means/ channels.

1.2 Motivation

In recent years, the number of devices has grown exponentially. This means that a large number of devices are connected to the Internet. With the increase in Internet communications, the attack surface has also merged. Report many IT security incidents every year, leading to financial losses and data theft [5]. Thanks to easy-to-use computing power and improved connectivity, opponents are now more powerful than ever. For example, as new network capacity increases, many new applications are realized. B. Real-time information service, teleconference and collaboration environment, in which information can be exchanged in groups. In group communication the message send by one authorized member of the group is received by all the other authorized members of that group. Cyber physical systems work collaboratively such that information and data is shared between devices. Environments in which multiple devices are connected and sharing data, are a particular favorite of the adversary as they can be easy to compromise. Adversary can exploit the system and gain access to the device by exploiting the weaknesses and vulnerabilities in the design of the system or by stealing the key using different key theft attacks and techniques. For an attacker, the cause of the attack may be a violation of national security or just a mild persecution. If the key theft attack is possible on a system, then this creates the need for a new and better approach for key generation, key communication, key retrieval and storage.

1.3 Problem Statement

Most commonly used cryptographic schemes have made there algorithm publicly available while the cryptographic key is the only thing that is kept secret. Kerchoff's security principle [4] states that the cryptographic system security is about keeping key secrets, not the protocols.. If the key of the system is compromised, the complete system can get compromised. For password-based keys, it is better to use strong and unique passwords and memorize them, so that there are less chances that someone will be able to guess the password. But there are different attacks like dictionary attack or rainbow table attack which can help the attacker in guessing the password. Keys by design are hexadecimal strings and are lengthy in size which

make it impossible for a human to memorize, which is why the keys are commonly stored on the device. This makes them vulnerable to many internal and external attacks. Cryptologists have extended keys, making it difficult for intruders to guess or force the use of keys, but this does not eliminate many key theft attacks.

In group settings, theft of encryption keys creates a unique environment for attackers because they can attack a variety of attractive targets. If only one participant in the group is attacked, the attack can be stepped up to destroy the entire group and control the group. Because there are many attack points, group communication is more vulnerable to attack.

This study represents a complete study that can ensure the safety of communication devices in a group configuration. The purpose of this study is to apply the theory and concept of Physical Non-Clone Function (PUF) to create keys in a multi-part environment. Research shows that PUF-ID devices can be used to create communication keys to ensure security in group settings.

The purpose of this research is to prove that PUF technology can generate encryption keys that guarantee confidentiality, integrity and identity verification. This guarantee is due to the fact that the key is based on a new trusted source. In this study, PUF technology has two search methods, one is the basis for generating encryption keys, and the other is a method to prevent theft.

1.4 Contributions

In conventional cryptography the security of the system relies on keys that are often stored. Keys stored in a device are vulnerable as an adversary can capture these keys using different methods [6][7]. Hence incorporating the PUF technology as a method of key theft deterrence can provide enhanced security as it can mitigate a big concern that is often faced by even the strongest cryptographic algorithms.

The purpose of this study is to show that the PUF file shows that unique device functions can be used to provide device identification, and that the logo can be used to provide security services. The first contribution to this article is to use PUF to create a symmetric key for the group. Symmetric key generation algorithms are based on group PUF, complex security primitives, and the use of symmetric key creation algorithms for group technology. These

algorithms can be used to generate encryption keys that ensure confidentiality, integrity, and authentication. The warranty is based on the fact that the key is based on a new and reliable source. In this study, PUF has two search methods, one is the basis for generating the encryption key, and the other is the anti-theft method.

The second contribution of this research is that the symmetric key is generated by the input. As a result, many members generate group keys after receiving donations, thereby eliminating the need for potentially sensitive third parties.

The third contribution of this article is to provide an architecture that you can use to verify the keys generated for the group.

Perhaps the biggest contribution of this article is that it can provide a high level of security without having to make major changes to the existing security system. Therefore, PUF technology can be integrated with any IT system with minimal impact on the existing infrastructure.

1.5 Aim of Research

When designing a cryptosystem, the most important point is to achieve the goals on which the development is based. Each safety goal must be defined according to other safety goals to ensure complete safety of the resulting system. The safety goal is the focus of this study because the system design options are based on the selected safety goal. The security scheme proposed in this study aims to achieve three basic security objectives: confidentiality, integrity, and identity verification. The project's safety objectives and their interpretation of the research are as follows.

- Confidentiality is defined as hiding information. Confidentiality means that only people who have been authenticated and authorized can view the information. Encryption is the most common method to ensure confidentiality.
- Integrity is defined as preventing unauthorized modification. Integrity ensures that no malware or unauthorized people have changed the data, and that the stored data on the device is correct. Honestly ensure that there are no contributions, or no changes are made

to the communication and prevent adversaries from making these fraudulent contributions.

- Authentication is defined as the process of identity verification and identification based on unique information, and the unique information is only known from the identity verification of the entity. Identity verification ensures that the person is the identity they applied for.

1.6 Physical Root of Trust

Attackers can now use advanced technologies with sufficient resources to carry out powerful attacks. Therefore, alternative support methods are being explored that can be used to improve the security of conventional cryptographic implementations. Cryptographic systems are traditionally based on mathematical principles. Most importantly, the algorithm is based on problems that are difficult or impossible to solve by brute force. through algorithmic intractability.

Mathematical difficulties are not enough to protect the system, because the attacker's behavior is not in accordance with the algorithmic flow. Attackers often attack systems without exploiting the inherent mathematical weaknesses or the algorithm. Attacks like side channel cold attack and cold boot attack are also used to penetrate the system, and these methods of attack are particularly deadly because they do not target the underlying design of the algorithm, which focuses on all activities in the design phase. In addition, security engineers often ignore many possibilities for attacking through side of the tunnel, which means that new technologies and methods of physical reflection are needed [8]. Since the primitives of the system are rooted in the physical world, you can use physical thinking to provide a higher level of security. In this study, the physical non-cloning function (PUF) was regarded as a reliable physical element [9]. PUF is essentially a function based on challenge and response. When a PUF is requested, the function returns a secret answer based on the physical and unique attributes of the device. This is a one way function whose function is unguessable, reproducible and unique to the device. Due to its powerful functionality, PUF can be used to generate extended random numbers (RNGs), perform authentication and provide password-related hardware services. Perhaps the best quality of the PUF is that it can be used as an alternative way to store keys. This quality makes it an

appropriate technology that can be used for improving both modern and traditional cryptographic schemes.

The thesis flows logically and builds on first the literature review. Later in the thesis novel key generation algorithms have been discussed. The precise flow of the thesis is.

- Chapter 2 studies the literature related to cyber physical systems, cryptography, security concerns and Physical Unclonable Functions. The beginning of the chapter describes the communication suite for devices in cyber physical system. The chapter also describes the concepts of cryptography along with its types and keys. This chapter also introduces the security issues that must be resolved to ensure safe communication between devices. This chapter also introduces PUF, which is a security problem that must be solved when using PUF to generate encryption keys.
- Chapter 3 focuses on key exchange. The beginning of the chapter describes the key exchange between two parties. It also describes the group key exchange and the limitations of the available schemes. This chapter details the PUF-based group key creation scheme.
- Chapter 4 focuses on Key distribution and key authentication. This chapter details the scheme for generating group keys using PUF-based key authentication.
- Chapter 5 closes the thesis by concluding the discussions and how the research can be explored in future.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Technological advancements in technology have led to computers becoming part of everyday life. Initially, computers were used for personal use or in placed in controlled environments like offices or labs. As computing technology evolved the communications through the internet increased, the use of a computer moved from controlled environments to more complex environment like online data centers, e-commerce systems, etc. As the computing and communication power increased the hardware also gets compact.

Due to advancements and availability of compact microprocessors, embedded systems are present in our physical environment. The capability of computing and communication embedded in all object and structures in the physical environment has coined a new term known as Cyber-Physical System (CPS).

2.2 Cyber-Physical System

Cyber physical system (CPS) is a new generation system that can use computers and communication functions to interact with the physical world. [10]. CPS supports the connection between the physical world and the network world. CPS was developed to integrate powerful computer logic and monitor and control the continuous dynamics of physical and technical systems. The system is designed to ensure that the integration of physical processes and computational processes goes smoothly [11]. CSP research is still in its early stages. but some breakthroughs have been found in domains of transportation, smart grid and renewable energy, biomedical and healthcare systems but there are still many challenges for CPS in these domains[12][13].

2.2.1 Smart Grid

Energy is vital for any company, city or country and protecting the critical infrastructure is very important for the economy. Initially, the power grids were operated manually due to which the communication between different grids were also a hassle. But now these electric supply grids have become smart and are a very good example of CPS. Smart grids have been used to manage and control the energy distribution and this is the reason that it is at forefront of public interest. Smart grids are a network of interconnected electric transmission units[13].

2.2.2 Transportation

Smart cars and driverless cars are another application of CPS. Research is underway on producing cars which will be driven autopilot whereby there is no need for a physical driver to drive those cars. Another application of CPS can be seen in the aviation industry as they are moving towards next-generation air transport and drones are a very good example[13].

2.2.3 Health Care

CPS is showing numerous opportunities in the domain of health care. Hospitals are becoming smart and the use of technology in the field of medicine has increased rapidly. Intelligent operation rooms, smart fluid flow control and monitoring, image-guided surgery are some examples of CSP[14].

2.3 Cryptography

Cryptography is the study of mathematical techniques related to information security, such as confidentiality, data integrity, entity identity verification and data source identity verification. About 4000 years ago, the Egyptians used encryption technology. The Greek army used encryption technology to encrypt. Later it was mainly used for diplomatic, military and government services.

A cipher is an algorithm that is used to generate a ciphertext of any plaintext or to extract plaintext from the ciphertext. Some of the early and popular ciphers used were Caesar Cipher and Vigenere Cipher. The German Enigma machine was also one of the most popular cipher machines which were used by the German Navy[15].

After the increase of use of computer system for communication, commercial use of cryptographic applications increased. Today cryptography can be considered an enabling technology for systems that are communicating via the internet. Digital communications and transactions would not be possible today without the presence of cryptographic techniques. Modern cryptography can basically be classified into two basic categories. The first is a symmetric password, and the second is an asymmetric password, also known as a public key password. [16]. Symmetric and asymmetric key cryptography dictate how keys are held by the individual parties involved in the secure communications. The Kerckhoff's [4] Security principle is observed regardless of the method used.

2.3.1 Symmetric Cryptography

For symmetric cryptography or symmetric key cryptography, only one key is used for encryption and decryption. All the communicating parties should have the same key so that they can decrypt the message which they have received from the other party or can encrypt their message and send to another party. Due to the usage of only one key for both the operations it is very important to use a strong key and to ensure its secrecy so that no one can read the conversation. To successfully implement symmetric key schemes, it is important to first establish a scheme for secure key distribution; as in its absence the keys could be compromised while being shared/ communicated.

The Data Encryption Standard (DES) is one of the most widely used symmetric encryption systems. It has been used as a standard by the National Institute of Standards and Technology (NIST), but due to its small size and low-key generation, it has not yet been considered for use. Safe to use, inconvenient and unpopular [17]. Advance Encryption Standards (AES) originally known as Rijndael [18] is the new NIST standard for symmetric encryption algorithm[19]. It has overcome the key size issue of DES and has three variations in key size means the key can be 128, 192 or 256 bits.

2.3.2 Asymmetric Cryptography

In asymmetric cryptography or public key cryptography, two keys are usually used as a public key and a private key. When someone wants to send you a message using a public key password, they use their public key to encrypt the message and send the encrypted message.

After receiving the message, use the private key to decrypt it. If others know the public key, they should keep the private key secret. Compared with the symmetric cryptographic algorithm, the asymmetric cryptographic algorithm is slower, but there is no key distribution problem found by the symmetric key algorithm.

One of the most commonly and basic used public key cryptographic system is RSA. It was proposed in 1977 and is still used as NIST Digital Signature Standard[20]. RSA's security is based on the factorization problem.

2.4 Security Concerns

Adversaries often attempt to infiltrate a system's security by exploiting the vulnerabilities of the system to gain illegitimate access. Security concerns have increased because the systems are moving out from the environmentally secure homes and offices to more complex and universal environments. The security of both, software and hardware, is highly important. In the section below, we have discussed the possible attacks on a system and their relevance in the everyday processing of a system.

2.4.1 Physical Attacks

Physical tampering of any hardware device is an important and fast-growing security concern. Since the hardware device is responsible for processing and storing data, it is very important to protect the device from attackers, because the attacker may cause the attacker to change or intercept the data. Generally, the data which is in the processing mode is only accessible by the system embedded for processing of the device and any access which is external from the system is not allowed to defeat the device tampering. Research [21] proves any physical device can be tempered by removing or detaching the components, temperature imprinting, probing, etc. All these attacks use the physical and chemical properties of the device to gain unauthorized access to the system.

Physical attacks on the system may lead to data theft [22]. An attacker can capture the data and then clone a device thus convincing the verifier that the device is legitimate. In [6] the authors used a cold reboot technique to acquire the data from the DRAM. They sprayed the canister of multi-purpose duster upside down directly onto memory chips to decrease its temperature. At low temperature, the data persist for a longer period. After decreasing the

temperature, they cold boot the system by removing the power supply from the system and they removed the DRAM from the system. Now the attacker can plug this DRAM to any other computer or can use different digital forensic tools to capture an image of DRAM and then extract the information from that image. Cryptographic solutions and strong access control can be used to defeat such attacks and the cloning of the device [23].

2.4.2 Communication Attacks

Communication-based attacks are another way through which the adversary tries to compromise a system's security. By exploiting communications, the attacker tries to penetrate the network to gain privileges of the user. Once privileged, the attacker can try to capture the encryption key from any system.

The most common attack in the network is an IP spoofing attack [24] [25]. In this attack, the attacker forges the IP address of the system to impersonate the actual user and generates falsified IP packets. If this attack is executed successfully, an attacker can gain complete control of the network, for example, rerouting traffic to its desired location, modifying or deleting the packets in the network, capturing or eavesdropping network packets, etc. IP spoofing is dangerous because the attacker can impersonate any authenticated and authorized user, so it provides the attacker an online cover or disguise and it's hard to find. IP spoofing attacks are most commonly used in distributed denial of service (DDoS) attacks.[24].

In October 2016 a DDoS attack was launched against a service provider named "Dyn". Web servers of many high profile social media and e-commerce sites [26]. Another similar yet more powerful attack named "Memcached" was launched recently in March 2018 which effected GitHub [27]. The attackers carried out this attack by exploiting the software "Memcached". The software is designed to load websites faster by buffering large amounts of data required for access. The attacker sent a small amount of unnecessary data to the memory cache server, thus generating a large amount of data for the attacker.

Another attack which is very common in a communication system is eavesdropping. Many wearable devices send data wirelessly over an insecure channel. If this data is sent unencrypted, then it can easily be eavesdropped by the attacker. Due to nonencrypted data, the

attacker can easily read or modify it. The way to defeat the threat of eavesdropping is to ensure the confidentiality of the data by encrypting it prior to communication.

2.4.3 Key Theft Attacks

Generally, encryption schemes are based on publicly available protocols and algorithms, and encryption keys are kept secret. Therefore, the security of the encryption algorithm depends on its key. If the key is compromised at any point of time, the security of the whole system can be compromised. In the traditional cryptographic system, the cryptographic keys used are precomputed and are either hardcoded or stored locally on the device which makes the system vulnerable to key theft attacks.

Due to the small key size or weak key and high computation power, it was easy for adversaries to brute force a cryptographic key. Cryptographers try to overcome this issue by increasing the size of cryptographic keys which makes the process of brute force hard for the adversaries to implement [28] but it does not in any way decrease the threat of key theft. There are many other methods and techniques by which key theft attacks are possible [29] [7] [6]. Some of the possible key theft attacks are listed below.

- Attackers can use brute force, other dictionary attacks, rainbow tables or man-in-the-middle attacks to destroy the encryption system.
- An adversary can use a keylogger to log or steal the password or key which is being entered by the user.
- It is possible for an attacker to physically extract the key from the system using different techniques. In [6] the authors used the cold reboot technique to extract the cryptographic keys from DRAM.

The attacker can also capture cryptographic keys by using side channel attacks. In [7] the authors used side channel attacks to extract 4096 bits keys of the RSA algorithm. They used a parabolic microphone to record the noise created because of high computation and power consumption. By analyzing the noise, they extracted the RSA algorithm key.

2.5 Physical Unclonable Functions

Research [30] [31] shows that no two identical silicon chips are produced. Even if they come from the same batch, the same plate or different batches. Even if the design, material, and manufacturing are the same there will be variations in the chips. These variations can be due to different reasons for instance pressure variance and process temperature at the time of manufacturing. Because of the variation between the chips the output generated is also different. These variants are used to create one-way functions called physical unclonable functions (PUFs).

PUF is a function based on physical attributes and quality and will not be cloned [31]. It is a challenge and response-based function. If a challenge x is queried to a PUF the function will provide a secret response y based on the unique characteristics of the device as shown in Figure 2.1. Due to the unique characteristics and physical properties of a chip, the output generated is unpredictable [30] [31].

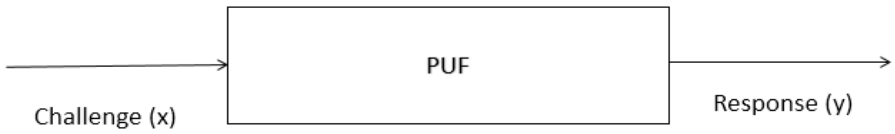

Figure 2.1: PUF as a challenge response function

PUF is robust against environmental variables [32] means when a challenge is given to a PUF it should produce the similar output with a high probability despite being affected by environmental variables like temperature, pressure, etc. Due to the variations between the two ICs, it is not possible for an attacker to generate the same output by two different PUFs which makes them unclonable [30] [31]. The response produced by the PUF is unpredictable because it uses physical characteristics of IC [31]. It is impossible for an adversary to tamper the PUF without affecting its challenge-response behavior.

Because of these properties, PUF can be used in various security related applications[32] [33]. In classic cryptographic schemes, the keys are important and the whole security of the system is based upon them, there are different techniques through which these keys can be attacked [22] [6] [7]. Research [32][31][34] shows that it can be used for random number

generation, authentication, and hardware entangled cryptography. PUF can also be used for generating keys because it is non-volatile and can eliminate the danger of the key theft.

Research is being carried out currently looking for unique features to create powerful PUF. In [35] the studies show that laser speckle fluctuations and coherent multiple scattering techniques can be used as optical PUF. Research [36] shows that RFID chips can also be used as PUF. The authors have designed the RFID ICs based on MUXes. They provide the RFID chip with a challenge of 64 bits to produce a unique output. In [32] the authors have established PUF silicon delay circuit based on arbiter and MUX. PUF based on hidden time or IC delay. Authors observed that the logical gates can be affected by factors like operation temperature and voltage supplied to the logical gates which is why this type of PUF is considered as weak PUF.

CHAPTER 3: GROUP KEY DH SCHEME

To achieve confidentiality of data during transmission encryption is used. Encryption of plaintext is regularly performed by using a key that converts from plaintext to cipher text. It is the key that needs to be kept secret because revealing it will defeat the purpose of encryption. If involved parties are using symmetric key cryptographic scheme, so everyone involved must have the same secret key and if they are using asymmetric key cryptographic scheme then all the communicating parties should have other's public keys.

Exchanging keys is a critical part of cryptography because many attacks at this point of a scheme will cause the attacker to obtain the cryptographic key. Key exchange also is known as key establishment is a method to exchange cryptographic keys between two or more parties willing to communicate using cryptographic algorithms. This chapter discusses the key exchange between two parties. In addition, the group key exchange limitations and available solutions are finally discussed, we have presented and discussed our group key generation scheme based on PUF and have also discussed the analysis of time for generating the keys under different configurations.

3.1 Two-Party Key Exchange

Key exchange is a difficult task, especially for symmetric key encryption. In a symmetric key, both the sender and receiver must have the same key to encrypt and decrypt messages. Therefore, leaking the key may damage the system. In order for both parties to exchange confidentiality, they must first exchange keys in a confidential manner, which no one knows[37]. For this purpose, either secure networks are used or some external channels like trusted couriers or diplomatic bags have been used.

Asymmetric or public key cryptography was one solution for this key exchange problem. The keys were exchange using public key cryptography and once the key is exchanged both the parties can use symmetric key cryptography for the transmission of messages[38]. In this case, the problem with public key encryption is that the public key is correctly distributed to its owner. Therefore, the user of the public key must ensure that it belongs to its owner and has not been misled or forged. Whitfield Diffie and Martin Hellman proposed another solution in 1976. They proposed a key exchange system called Diffie-Hellman key exchange.[39].

3.1.1 Diffie-Hellman

Diffie-Hellman Key Agreement Protocol (DHKE) is the most popular key exchange protocol and is still in use, but with some modifications. The security of the key MOU is based on the discrete logarithm problem[39]. The protocol allows two users to exchange keys on an insecure network without having to maintain/replace the old key.

In the original scheme [39] if two parties A and B want to agree upon a secret key they must follow the following process: First, both parties must agree upon a prime p and then select a generator g of the multiplicative group Z_p^*. Both parties A and B select the random secret values x and y respectively. After selecting the random secret both parties must calculate their respective public values $g^x \, mod \, p$ and $g^y \, mod \, p$ and exchange them. Finally, party A computes$(g^y)^x \, mod \, p$ and party B computes $(g^x)^y \, mod \, p$. Since $(g^y)^x \, mod \, p$ is equal to $(g^{xy}) \, mod \, p$ and $(g^x)^y \, mod \, p$ is also equals to $(g^{xy}) \, mod \, p$ so both the parties have now the same shared secret[39]. The original DHKE was vulnerable to man in the middle (MITM) attacks.

3.2 Group Key Exchange

Due to the growing popularity of team-oriented applications, secure group communication is considered an important aspect of privacy. The most important element in any cryptographic system is the key. If the key generation and key distribution process of any cryptographic system are flawed, then the cryptographic system is considered a vulnerable system. Generating and distributing the keys in group setting is a difficult task.

A Group Key Agreement (GKA) protocol is a protocol where a group of members can agree upon a key in such a manner that the output of the algorithm is based on the contributions from all the members. The main objective of the GKA is to establish a confidential channel for the members of the group to communicate. Here we limit our discussion to symmetric keys only.

Since the key agreement is essential for secure group communication, different schemes were proposed but most are considered weak and vulnerable to known attacks or are very expensive in terms of computations. Another issue with GKA schemes was that either a precomputed secret or a certificate was required for key agreements. The scheme proposed in [40] has a single point of failure, thus if the central connecting device which in this case is a bridge is compromised or is not available this scheme cannot work. The message size required was also very large which make the scheme computationally expensive. In [41] the scheme proposed is vulnerable to key theft attack. An attacker can derive the key if it manages to eavesdrop the message at three consecutive links in the conference network. The scheme proposed in [42] requires a precomputed certificate for the initiation of the protocol. Due to the precomputed certificate, this scheme is prone to key theft attack and the overhead to store the certificate is also there. In [43] the researchers have proposed the extension of two-party Diffie-Hellman to n parties Diffie-Hellman.

3.3 Group Key DH Scheme

3.3.1 Initial Key Agreement (IKA)

The proposed scheme is based on the protocol discussed in [43]. It is similar to two parties DHKE. Similar in its working with two parties; the security of DHKE is also based on Diffie Hellman discreet logarithm problem. Our proposed scheme consists of three stages.

Before discussing the first stage a basic setup is needed just like two party Diffie-Hellman. All the members must agree upon a large prime p and generator g. In the first stage, every member must create its unique secret which will be used for calculating contributions. Each member has it’s unique PUF ID. They will select a large random number. After selecting a random number, they will concatenate the random number with PUF ID and take the hash of the concatenated string as shown in equation (1) which they will use for in the next stage.

$$HASH\ (PUF\ ID \mid Random\ Number) \quad (1)$$

The second stage is to collect contributions from all members of the group. In this stage, each member must compute its share based on the values received from the previous member and send the computed intermediate values to the next member as shown in equation (2).

$$M_i \qquad\qquad M_{i+1}$$
$$\left\{G^{\prod\{R_k \mid k \in [1,i] \wedge k \neq j\}} \mid j \in [1,i]\right\}, G^{R_1 * \ldots\ldots * R_i} \quad (2)$$
$$Upflow: round\ i; i\ \in [1, n-1]$$

For example, if P_4 receives a set of values $\{g^{R1R2R3}, g^{R1R2}, g^{R1R3}, g^{R2R3}\}$ from P_3. P_4 must compute $\{g^{R1R2R3R4}, g^{R1R2R3}, g^{R1R2R4}, g^{R1R3R4}, g^{R2R3R4}\}$ and send this to P_5.

The third stage is to calculate the final key. In this stage, the final member of the group will broadcast all the intermediate values so that all the other group members can calculate the final key using their respective intermediate values as shown in equation (3).

$$M_i \qquad\qquad M_n$$
$$\left\{G^{\prod\{R_k \mid k \in [1,n] \wedge k \neq i\}} \mid i \in [1,n]\right\} \quad (3)$$
$$Broadcast: round\ n$$

For example, if P_5 is the final member of the group then at this stage P_5 will calculate $\{g^{R1R2R3R4R5}, g^{R1R2R3R4}, g^{R1R2R3R5}, g^{R1R2R4R5}, g^{R1R3R4R5}, g^{R2R3R4R5}\}$ and will broadcast the intermediate values $\{g^{R1R2R3R5}, g^{R1R2R4R5}, g^{R1R3R4R5}, g^{R2R3R4R5}\}$ so that all the other members can calculate their keys. User P_4 will take its relevant intermediate value $g^{R1R2R3R5}$ and calculate $(g^{R1R2R3R5})^{R4}$ to get the final key $g^{R1R2R3R4R5}$. Similarly, all the remaining members will calculate the final key in the same manner. Figure 3.1 shows the complete scheme design.

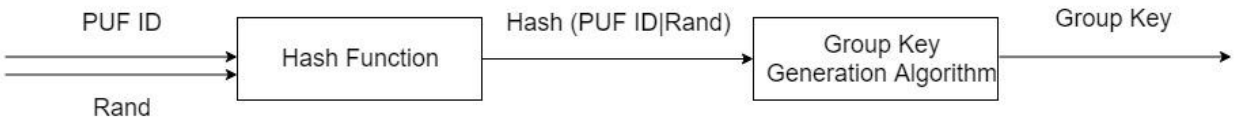

Figure 3.1: Complete Scheme Design

3.3.2 Alteration of Group Memberships

Alteration of members is an important part in dynamic groups. Due to the changes in the group the freshness of the key is very important. In this section, we will discuss other auxiliary group key operations like adding a new member and removing the old member.

3.3.2.1 Member Addition

When adding a new member in the group it must be ensured that the new member cannot decrypt the old messages, hence a new key must be computed. By computing the new key, backward secrecy can be achieved.

For example, if P_5 was the final member of the group and we want to add a new member P_6 in the group. Now P_5 will have to calculate its new unique secret $R5'$. Now P_5 will calculate $\{g^{R1R2R3R4R5'}, g^{R1R2R3R4}, g^{R1R2R3R5'}, g^{R1R2R4R5'}, g^{R1R3R4R5'}, g^{R2R3R4R5'}\}$ and will send this to the new member P_6. Now P_6 will calculate its unique secret $R6$ and compute $\{g^{R1R2R3R4R5'R6}, g^{R1R2R3R4R6}, g^{R1R2R3R5'R6}, g^{R1R2R4R5'R6}, g^{R1R3R4R5'R6}, g^{R2R3R4R5'R6}\}$.
Member P_6 will broadcast the intermediate values $\{g^{R1R2R3R4R6}, g^{R1R2R3R5'R6}, g^{R1R2R4R5'R6}, g^{R1R3R4R5'R6}, g^{R2R3R4R5'R6}\}$ so that all the other member can come to the final key.

3.3.2.2 Member Deletion

When deleting a member from the group it must be made sure that the deleted member cannot decrypt the new message using its old key thus achieving forward secrecy. This can be accomplished by computing a new key.

For example, if P_5 is the final member of the group and a group member P_3 wants to leave the group. To achieve forward secrecy a new key need to be calculated. For this P_5 will have to calculate its new unique secret $R5'$. Now P_5 will calculate $\{g^{R1R2R3R4R5'}, g^{R1R2R3R4}, g^{R1R2R3R5'}, g^{R1R2R4R5'}, g^{R1R3R4R5'}, g^{R2R3R4R5'}\}$ and will send intermediate values $\{g^{R1R2R3R5'}, g^{R1R3R4R5'}, g^{R2R3R4R5'}\}$ to all the member except P_3 so that all the remaining members can compute the final key.

3.3.3 Pseudo Code

3.3.3.1 Take Contribution

```
Procedure: TakeContribution
Input: BigInteger "G, N, R, A list of numbers Previous"
Output: A list of intermediate values "Values"
Values[Previous.Length + 1]
Cardinal, PreviousCV, Intermediate, Temp  ←  0
Temp  ←  Previous[0]
Cardinal  ←  (Temp ^ R) mod N
Values[0]  ←  Cardinal
PreviousCV  ←  Temp
Values[1]  ←  PreviousCV
IF Previous.Length EQUALS 2
THEN,
     Intermediate  ←  (G ^ R) mod N
     Values[2]  ←  Intermediate
ELSE,
     FOR i  ←  2 TO Previous.Length
     DO,
          Temp  ←  Previous[i-1]
          Intermediate  ←  (Temp ^ R) mod N
          Values[i]  ←  Intermediate
     FOR END
RETURN Values
```

The above pseudo-code is of a procedure used for collecting the contributions from the members of the group. The values required as input by this procedure are G, N, R, and an array named $Previous$. G is a large prime number used as an exponential base, N is a large prime number used for order of the algebraic group (mod), R is the hash of PUF ID with a random number and $Previous$ is an array of intermediated values received from the previous participant. In the case of the first participant, this array will be empty.

3.3.3.2 Calculate FinalKey

```
Procedure: CalculateFinalKey

Input: BigInteger "IntermediateValueRelevant, R, N"

Output: BigInteger "FinalKey"

FinalKey ← (IntermediateValueRelevant ^ R) mod N
```

The above pseudo-code is of a procedure for calculating the final group key. All the intermediate values are broadcasted by the last member of the group so that all the members can calculate the final group key. In this procedure all the members take their relevant intermediate value and get the final key by taking the calculating $(IntermediateValueRelevant \text{ ^ } R)\ mod\ N$ where $IntermediateValueRelevant$ is the relevant intermediate value for that member, R is the secret of that member and N is a large prime number used for order of the algebraic group (mod).

3.3.4 Implementation and Outcomes

3.3.4.1 Implementation

The proposed symmetric key scheme has been simulated and tested on a 2.60 GHz second-generation Intel Core i5 3320M computer with 8 GB RAM. The language used for programming is JAVA[44] and the version is Java 1.8.0_121. The platform used for development is Net Beans IDE[45] version 7.3.1.

3.3.4.2 Outcomes

The Diffie-Hellman group graph is affected by two parameters, which are the key size and the number of participants in the group. In order to test the performance of the algorithm, five keys were created, the number of participants in the group was different, and the key size remained the same. The generated key sizes are 160, 256, and 512 bits, and the number of participants is 100, 200, 300, 400, and 500. Table 3.1 shows the total time required by Diffie-Hellman when there is difference in the size of the group key and the number of participants. The bigger the key the longer it took.

Table 3.1: Total Time Taken by The Group Key DH Scheme

Key Size	Number of Participants	Total Time (Milliseconds)
160 Bits	100	2811
	200	5100.6
	300	9992.4
	400	15753.8
	500	24848.2
256 Bits	100	3559
	200	9953
	300	19276.2
	400	33401.6
	500	49265.4
512 Bits	100	8546.6
	200	29635.2
	300	80917.6
	400	147624.8
	500	236857.2

Analysis of the group key graph shows that a 512-bit key with 500 participants will take longer to operate. Other analysis shows that 100 or 200 participants have created keys with moderate needs. The increase in the number of participants will increase the time to calculate keys of the same size. The analysis shows that increasing the size of the key will also increase

the time required to calculate the key and keep the number of participants unchanged. Figure 3.2 shows the effect of key size on time taken to produce keys.

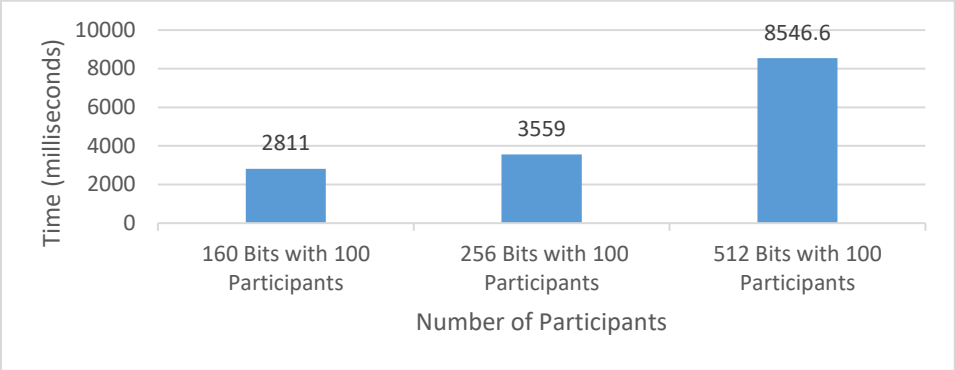

Figure 3.2: Graph Showing Time Taken by Various Key Size with The Constant Number of Participants

Study of the proposed scheme shows that the time requires to calculate the key is not only affected by the key size but also by the number of participants. As shown in the graph below an increase in time required to calculate a key if we keep the key size constant and change the number of participants. Graph depicting the effect of participants on time requirement are shown in Figure 3.3.

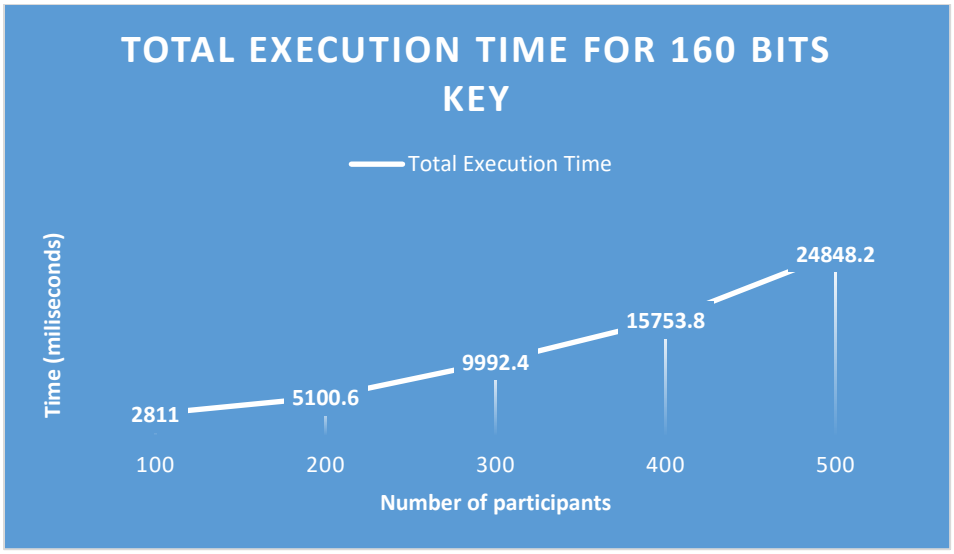

Figure 3.3: Graph Showing Time Taken by 160 Bits Key with The Variable Number of Participants

The proposed scheme has two basic functions. The first function is for taking contributions from all the group members and the second is to calculate the final key. Analysis

shows that calculating the unique secret and taking contributions from the group members (upFlow function) requires more time as compared to calculating the final key (finalKey function) which is shown in Figure 3.4.

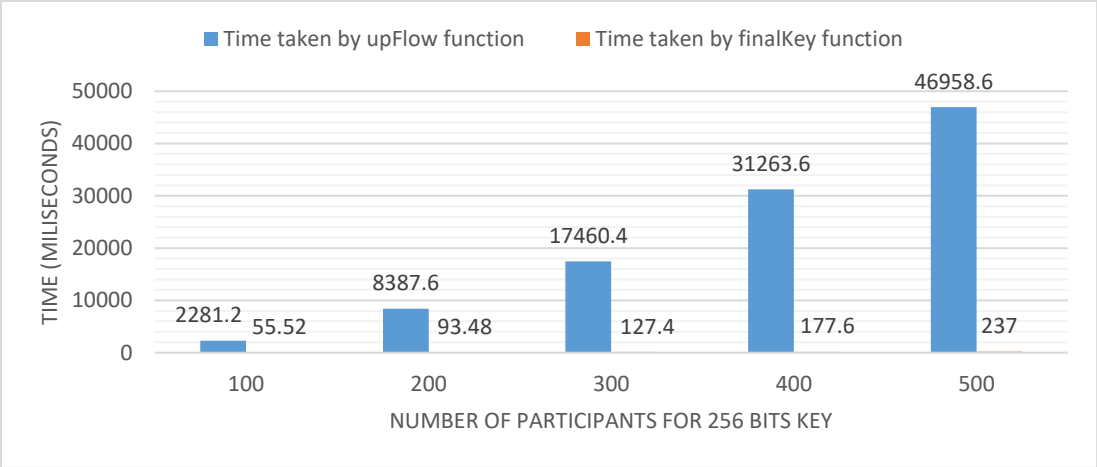

Figure 3.4: Graph Showing a Comparison Between Time Taken by UpFlow Function and FinalKey Function

Figure 3.5 and Figure 3.6 respectively shows the graphs depicting the effect on the time taken by upFlow and finalKey function keeping the key size constant and varying the number of participants in the group.

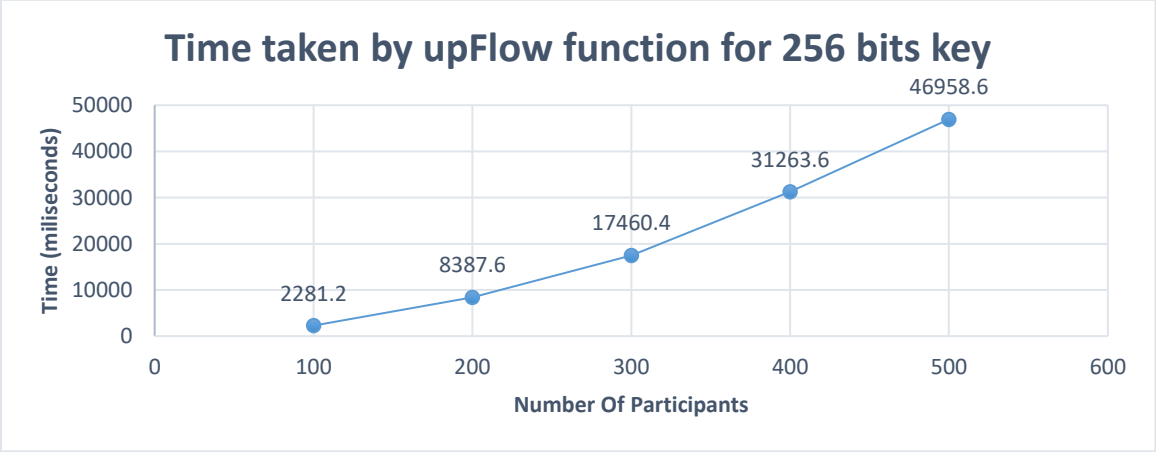

Figure 3.5: Graph Showing Time Taken by UpFlow Function

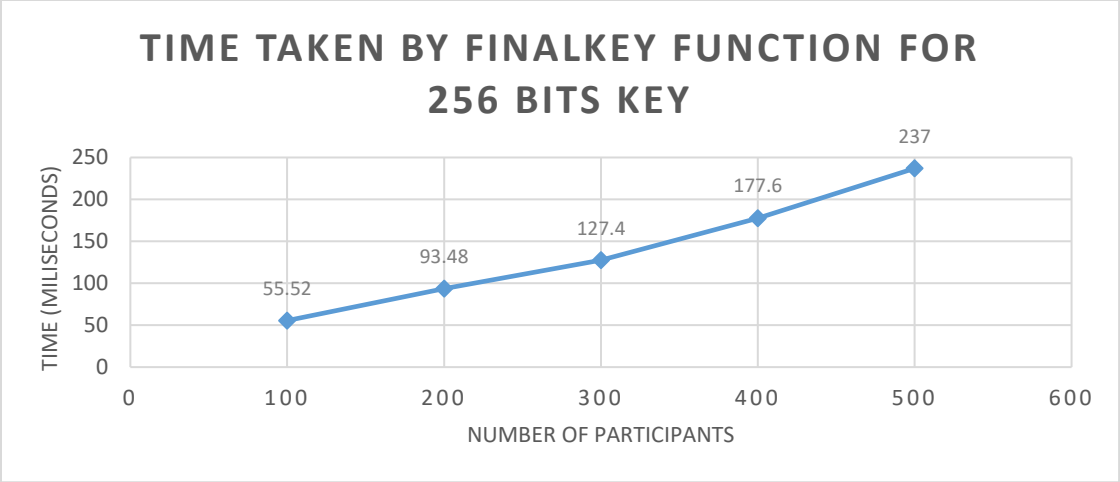

Figure 3.6: Graph Showing Time Taken by FinalKey Function

We have discussed the IKA functions and shown their graphs now we will discuss the effects due to alteration of members in a group. Earlier in the chapter the importance of alteration of the member in a dynamic group was discussed. . The effect of group membership alteration is highlighted next.

The analysis shows that the time required to add a new member and recompute the new key requires more time than deleting a member from the group and recomputing the new key. Figure 3.7 shows the comparison between the time required to add a new member and the time required to remove a member from the group.

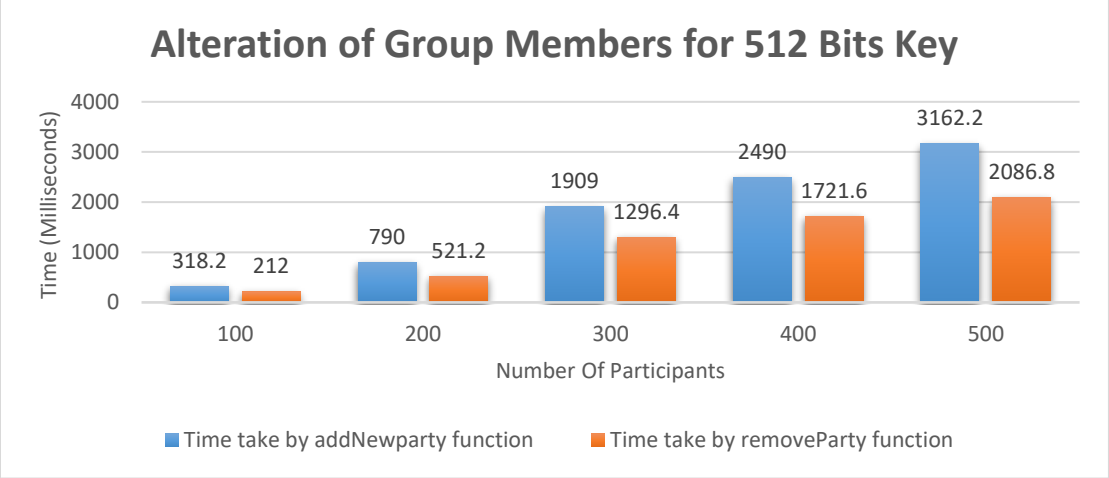

Figure 3.7: Graph Showing Comparison Between addNewparty Function and removeParty Function

The time required to add new member can be affected if the number of initial participants in the group is kept constant and the key size is changed. Simulation analysis is shown in Figure 3.8.

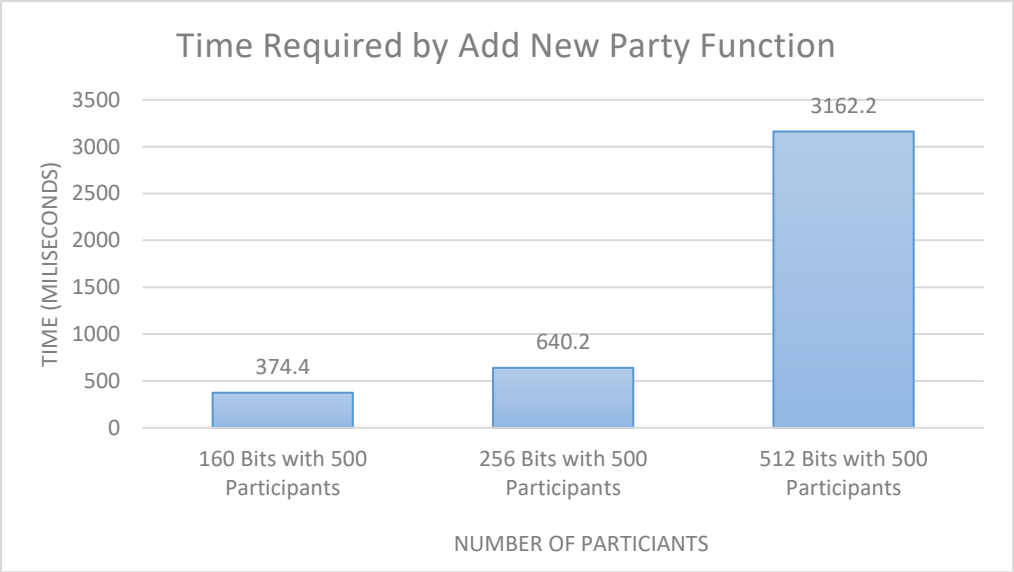

Figure 3.8: Graph Showing Time Taken by addNewparty Function with Constant Participants and Varied Key Size

Figure 3.9 and Figure 3.10 respectively shows the graphs depicting the effect on the time taken by addNewparty and removeParty function keeping the key size constant and varying the number of participants in the group.

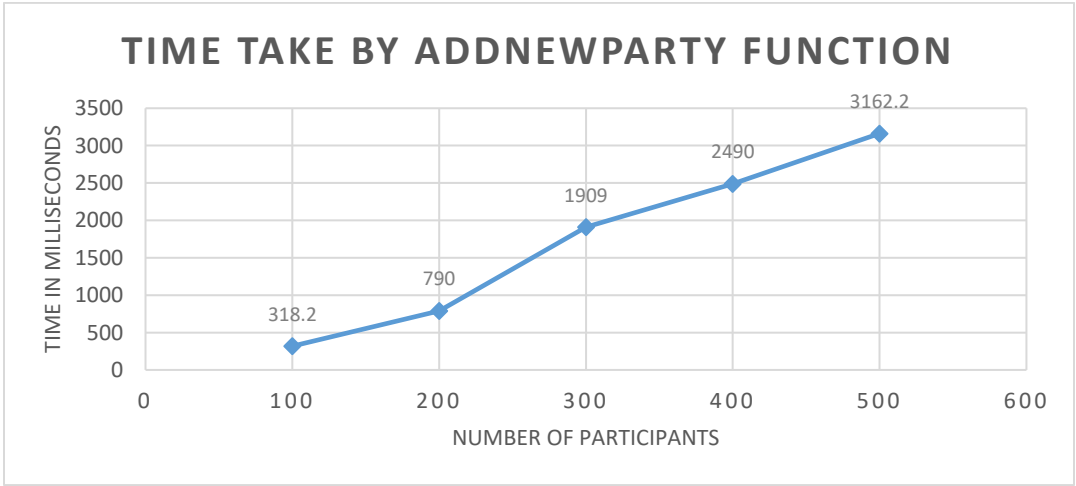

Figure 3.9: Graph Showing Time Taken by addNewparty Function

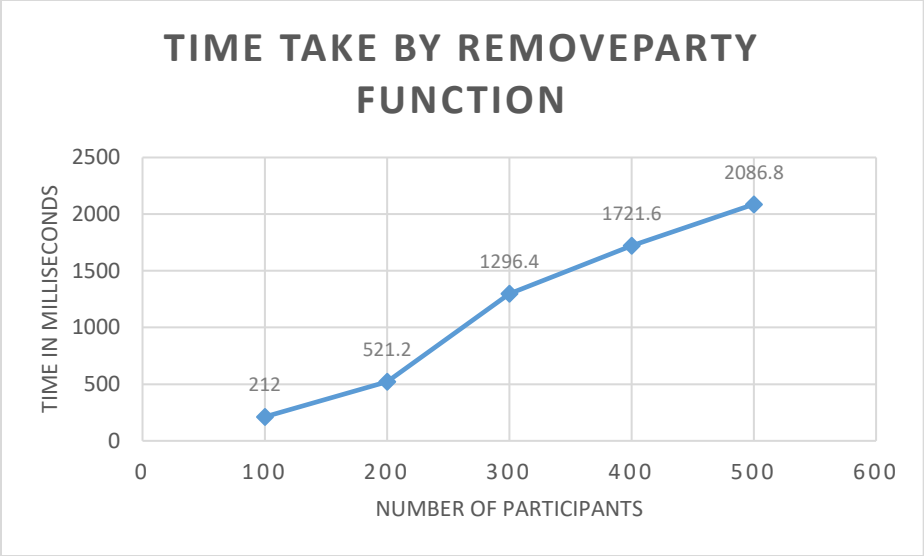

Figure 3.10: Graph Showing Time Taken by removeParty Function

3.4 Summary

The function that cannot be physically cloned is considered to be a secure basis for creating encryption schemes. This chapter explains that group keys can be created based on symmetric PUF. The proposed scheme shows that the combination of PUF and Diffie-Hellman group can generate a symmetric security group key. Based on the above results, we can conclude that the time required to create a group key is affected by two factors: the first is the size of the key, and the second is the number of group members. If the key size increases and the group configuration members remain the same, it will take longer to generate the key. Similarly, if the key size remains the same and the group membership increases, the time required to generate the group key also increases. The analysis also shows that adding new members takes more time than deleting existing members.

CHAPTER 4 GROUP KEY DH SCHEME WITH AUTHENTICATION

Confidentiality of data can be achieved through the use of encryption schemes. There are many different cryptographic algorithms available which can be used for encryption to make sure the confidentiality of the data is retained. It is mentioned in [22] that the security of the system depends on the confidentiality of the key, not the algorithm. If symmetric key cryptography is used, then all the members involved need to have the same key. If asymmetric key cryptography is used, then all the communicating members should have the public keys of every other member and the keys should be exchanged or distributed between the members involved regardless of the type.

Key distribution is a critical part of any cryptographic algorithm. Multiple key distribution schemes are available. Some algorithms are either prone to know attacks or have exploitable vulnerabilities or have authentication issue. This chapter discuss the key distribution and key authentication between multiple parties. Furthermore, the limitations of group key distribution and authentication in the existing schemes are also discussed, and in the end, we have presented and discussed our group key generation and authentication scheme based on PUF and have also discussed the analysis of time for generating the keys under different configurations.

4.1 Key Distribution

Exchanging or distribution of cryptographic key is a complex task if we consider the existence of adversaries. Asymmetric encryption uses two keys. The first is the private key, which the owner keeps secret, and the second is the public key, whose name is recommended to be distributed to the public. The keys function such that a single key will carry out the reverse

operation of what was carried out by a corresponding key. There are many ways to distribute the public key either by posting them on the website of the owner or by publishing it on public forums, but the most common and widely accepted technique is the public key certificate[46][47].

In symmetric cryptography, only one key is used for encryption and decryption. This means that the sender and recipient must have the same password and it should be kept secret because an exposure of the key can compromise the confidentiality of the data. If two party's A and B wish to exchange the symmetric keys a way of doing this is to deliver the key physically. Physical key exchange is not always a practical option as the communicating parties could be geographically distributed. Another option is the establishment of a secure channel to exchange or distribute the symmetric key[39]. Using a secure channel will not only increase the overheads but also the parties need to trust the secrecy of the established channel.

To overcome these issue, two techniques were proposed. First is to use a third party for distribution of the key and second is to use tickets [48]. For third party distribution the most commonly used scheme is based on a Key Distribution Center (KDC) and for ticket-based technique the most commonly used scheme is Kerberos [49].

4.1.1 Key Distribution Center (KDC)

Needham Schroeder Protocol [50] is one of the most common schemes used for symmetric key distribution. If one party called party A wants to securely communicate with the other party called party B, both the parties should be registered with the same KDC server. When the parties register with the KDC server, a master key is shared with that party which will be used for communications with the KDC server. When party A wants to communicate with party B, it will first go to KDC server and will request for the session key for party B.

There are multiple issues in this scheme. Single point of failure is one of the issues because the KDC server is the center point for key distribution and all the parties who require keys will approach the KDC. Another known issue is that if party A wants a session key to communicate with party B from KDC server, how can party A be sure that the key is send by the KDC server. Thus, trusting the third party is an inherent issue in key distribution schemes. Even

if the members have trust in the KDC, authentication of the key is still an issue in this scheme [51].

4.1.2 Kerberos

Kerberos is a key distribution and authentication scheme that is widely used for network based authentication [52][53]. The algorithm is based-on client server model and is designed for authentication of clients in the network domain using cryptographic keys. If a client or user wants to access some services, then it must first prove its authenticity. The client or the user must first go to the Kerberos server and proves its identity. Kerberos server consists of two main parts. First is the Key Distribution Server (KDS) which itself consists of two parts and the second is the ticket granting server (TGS). KDS also includes two parts: the first part is the authentication server, the second part is the database.

Kerberos works on the ticket-based principle. If a client wants to access some resource, it must get the ticket from the TGS which is a part of the Kerberos server and then must present that ticket to the service. The concept of the ticket is to enable centralized authentication. Kerberos is used widely for authentication in networks and is used by Microsoft for authentication [54]. Despite its wide use there are some issue in Kerberos scheme. It is a centralized system, so it is vulnerable to single point of failure. Another issue is that it is not a good option for group authentication. The most critical vulnerability is the golden ticket or non-expiring ticket. If an attacker gets a golden or non-expiring ticket it can bypass the authentication mechanism [55].

4.1.3 Key Authentication

After verifying the key identity, the claimed user key identity will be verified. Authenticity of the key is checked after the key is distributed. If the key distributed is corrupted or changed by the attacker or by dishonest participant, then the confidentiality and integrity will be compromised. Dishonest participants are a great concern for key generation and distribution in group communication.

4.1.4 Dishonest Participants

A multi-part environment consists of many devices that communicate with each other. In a multi-party environment, the presence of dishonest participants is one of the most important challenges for group key distribution security. When distributing the key in presence of dishonest participants the security of the group key distribution can be compromised. Many group key generation schemes that are available or widely used are weak and dishonest participants can take advantage and it threatens the safety of key group communication. The scheme proposed[56] is vulnerable to key theft attack as a dishonest participant can connect to three different participants at the same time, thus deriving the key. The scheme proposed in [57] requires precomputed certificates hence if a dishonest participant can craft the packet with known plaintext or known cyphertext and forge the certificate then the dishonest participant can create a key of its choice.

4.2 Group Key DH Scheme with Authentication

4.2.1 Initial Key Agreement (IKA)

This scheme is based on the protocol discussed by [43]. It is similar to two parties DHKE. Similar to its working with two parties, the security of the algorithm is also based on DH discreet logarithm problem. The key authentication has been incorporated in the proposed scheme so that all members of the group can authenticate that they are in possession of the correct key. This scheme is distributed into three stages namely:

1. Party Secret Generation
2. Up flow Stage
3. Broadcast Stage

Before going to the first stage a basic setup is needed just like two party Diffie-Hellman. All the members must agree upon a large prime p and generator g. In the first stage, all the members must create its unique secret which will be used for calculating contributions. Each member has it's unique $PUF\ ID$. They will select a large random number. After selecting a random number, they will concatenate the random number with $PUF\ ID$ and take the hash of the concatenated string as shown in equation (4)s which they will be used in the next stage.

$$HASH\ (PUF\ ID \mid Random\ Number) \tag{4}$$

The second stage is to collect contributions from all members of the group. In this stage, each member must compute its share based on the values received from the previous member. and send the computed intermediate values to the next member as shown in equation (5).

$$M_i \qquad\qquad M_{i+1}$$
$$\left\{G^{\prod\{R_k \mid k \in [1,i] \wedge k \neq j\}} \;\middle|\; j \in [1,i]\right\}, G^{R_1 * \ldots\ldots * R_i} \longrightarrow \tag{5}$$
$$Upflow: round\ i; i \in [1, n-1]$$

For example, if P_4 receives a set of values $\{g^{R1R2R3}, g^{R1R2}, g^{R1R3}, g^{R2R3}\}$ from P_3. P_4 must compute $\{g^{R1R2R3R4}, g^{R1R2R3}, g^{R1R2R4}, g^{R1R3R4}, g^{R2R3R4}\}$ and send this to P_5.

In the third step, the last key is calculated. At this time, the final member of the group sends all intermediate values, and the last member uses the HMAC value calculated from the final keys generated from the messages of all members so that all other members of the group use the appropriate final key. End key authenticity as shown in equation (6).

$$Mi \qquad\qquad Mn$$
$$\left\{(G^{\pi\{R_k \mid k \in [1,n] \wedge k \neq i\}} \;\middle|\; i \in [1,n]), (HMAC)\right\} \longleftarrow \tag{6}$$
$$Broadcast: round\ n$$

As an example consider P_5 is the final member of the group then at this stage P_5 will calculate $\{g^{R1R2R3R4R5}, g^{R1R2R3R4}, g^{R1R2R3R5}, g^{R1R2R4R5}, g^{R1R3R4R5}, g^{R2R3R4R5}\}$ and will broadcast the intermediate values $\{g^{R1R2R3R5}, g^{R1R2R4R5}, g^{R1R3R4R5}, g^{R2R3R4R5}\}$ and $HMAC$ so that all the other members can calculate their keys. P_4 will take its relevant intermediate value $g^{R1R2R3R5}$ and calculate $(g^{R1R2R3R5})^{R4}$ to get the final key $g^{R1R2R3R4R5}$. After calculating the final key P_4 will calculate the $HMAC$ using the final key and compare the calculated $HMAC$ with the one received from the broadcast. Similarly, all the remaining members will calculate the final key in the same manner. Figure 4.1 shows the complete scheme design.

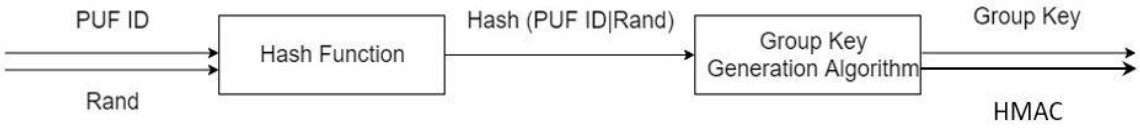

Figure 4.1: Complete Scheme Design

4.2.2 Alteration of Group Members

Alteration of members is an important part in dynamic groups. Due to the changes in the group the freshness of the key is very important. This section discusses other auxiliary group key operations like adding a new member and removing the old member.

4.2.2.1 Addition of New Member

When adding a new member in the group we must make sure that the new member cannot decrypt the old messages, so we must calculate a new key. By calculating the new key, we can achieve backward secrecy.

For example, if P_5 was the final member of the group and we want to add a new member P_6 in the group. Now P_5 will have to calculate its new unique secret $R5'$. Now P_5 will calculate $\{g^{R1R2R3R4R5'}, g^{R1R2R3R4}, g^{R1R2R3R5'}, g^{R1R2R4R5'}, g^{R1R3R4R5'}, g^{R2R3R4R5'}\}$ and will send this to the new member P_6. Now P_6 will calculate its unique secret $R6$ and calculate $\{g^{R1R2R3R4R5'R6}, g^{R1R2R3R4R6}, g^{R1R2R3R5'R6}, g^{R1R2R4R5'R6}, g^{R1R3R4R5'R6}, g^{R2R3R4R5'R6}\}$, the new $HMAC$ and will broadcast the intermediate values $\{g^{R1R2R3R4R6}, g^{R1R2R3R5'R6}, g^{R1R2R4R5'R6}, g^{R1R3R4R5'R6}, g^{R2R3R4R5'R6}\}$ so that all the other member can calculate the final key and verify the authenticity of the final key generated using the HMAC.

4.2.2.2 Member Deletion

When deleting the member from the group we must make sure that the deleted member cannot decrypt the new message using an old key that means we have to achieve forward secrecy. To achieve forward secrecy, a new key need to be computed.

For example, if P_5 is the final member of the group and a group member P_3 wants to leave the group. Now to achieve forward secrecy a new key need to be calculated. For this P_5 will have to calculate its new unique secret $R5'$. Now P_5 will calculate $\{g^{R1R2R3R4R5'}, g^{R1R2R3R4}, g^{R1R2R3R5'}, g^{R1R2R4R5'}, g^{R1R3R4R5'}, g^{R2R3R4R5'}\}$ and the new $HMAC$ and will send intermediate values $\{g^{R1R2R3R5'}, g^{R1R3R4R5'}, g^{R2R3R4R5'}\}$ to all the member except P_3 so that all the remaining member can calculate final key and verify the authenticity of the final key generated using the HMAC.

4.2.3 Pseudo Code

4.2.3.1 Take Contribution

The take contribution stage is use for collecting contributions from all the members of the group for generating the group key. The pseudo-code below is of a procedure used for collecting the contributions from the members of the group. The values required as input by this procedure are G, N, R, and an array named $Previous$ that holds intermediate values. G is a large prime number used as an exponential base, N is a large prime number used for order of the algebraic group (mod), R is the hash of PUF ID with a random number. In the case of the first participant, the $Previous$ array will be empty.

```
Procedure: TakeContribution

Input: BigInteger "G, N, R, A list of numbers Previous"

Output: A list of intermediate values "Values"

Values[Previous.Length + 1]

Cardinal, PreviousCV, Intermediate, Temp  ←  0

IF Previous.Length GREATER THEN 0

Then,

     Temp  ←  Previous[0]

     Cardinal  ←  (Temp ^ R) mod N
```

```
        Values[0]  ←  Cardinal

        PreviousCV  ←  Temp

        Values[1]  ←  PreviousCV

    ELSE,

        Values[0]  ←  (G ^ R) mod N

        Values[1]  ←  Values[0]

    IF Previous.Length EQUALS 2

    THEN,

        Intermediate  ←  (G ^ R) mod N

        Values[2]  ←  Intermediate

    ELSE,

        FOR i  ←  2 TO Previous.Length

        DO,

            Temp  ←  Previous[i-1]

            Intermediate  ←  (Temp ^ R) mod N

            Values[i]  ←  Intermediate

        FOR END

    RETURN Values
```

4.2.3.2 Calculate FinalKey

Calculate FinalKey is used for broadcasting the final set of intermediate values so that all the members of the group can calculate the final key. The pseudo-code below is of a procedure

for calculating the final group key. All the intermediate values and the HMAC are broadcasted by the last member of the group so that all the members can calculate the final group key. In this procedure all the members take their relevant intermediate value and get the final key by taking the calculating $(IntermediateValueRelevant \text{ ^ } R)\ mod\ N$ where $IntermediateValueRelevant$ is the relevant intermediate value for that member, R is the secret of that member and N is a large prime number used for order of the algebraic group (mod).

```
Procedure: CalculateFinalKey

Input: BigInteger "IntermediateValueRelevant, R, N"

Output: BigInteger "FinalKey"

FinalKey ← (IntermediateValueRelevant ^ R) mod N
```

4.2.4 Implementation and Outcomes

4.2.4.1 Implementation

The proposed symmetric key scheme has been simulated and tested on a 2.60 GHz second-generation Intel Core i5 3320M computer with 8 GB RAM. The language used for programming is JAVA[44] and the version is Java 1.8.0_121. The platform used for development is Net Beans IDE[45] version 7.3.1.

4.2.4.2 Outcomes

The Diffie-Hellman group scheme is affected by two parameters, namely the size of the key and the number of participants or members in the group. In order to test the performance of the algorithm, five public keys will be generated, the number of participants in the group is different, and the key size remains constant. The generated key sizes were 160, 256 and 512 bits while the number of participants is 100, 200, 300, 400 and 500. Table 4.1 shows the total time taken by the proposed Group Key Diffie-Hellman scheme when there is difference in the size of the group key and the number of participants. The bigger the key the longer it took

Table 4.1: Time Taken by The Group Key DH Scheme with Authentication

Key Size	Number of Participants	Total Time (Milliseconds)
160 Bits	100	5174.2
	200	8094
	300	12306.4
	400	16273.8
	500	24236.4
256 Bits	100	6592.6
	200	12716.2
	300	25759.6
	400	42889.4
	500	66686.4
512 Bits	100	15840.6
	200	44092.4
	300	94026.6
	400	171014.4
	500	257074.2

It was found that increasing the number of participants would extend the time required for the system to calculate keys of the same size. Analysis shows that increasing the size of the key will also increase the time required to calculate the key and keep the number of participants unchanged. Figure 4.2 shows the effect of key size on time taken to produce keys.

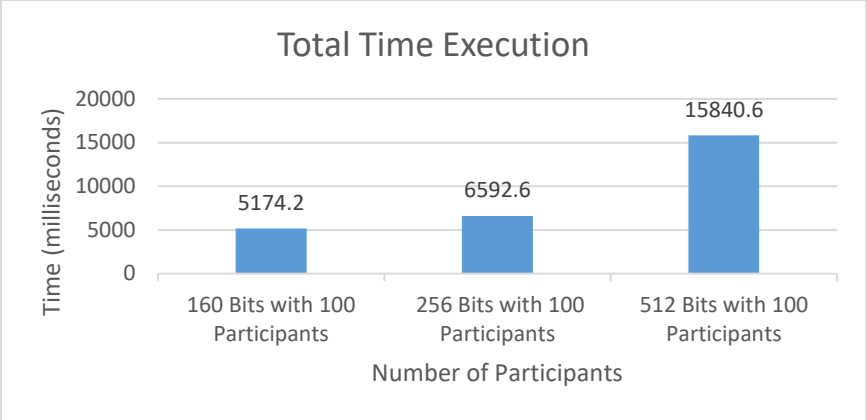

Figure 4.2: Graph Showing Time Taken by Various Key Size with The Constant Number of Participants

The analysis shows that the time requires to calculate the key is not only affected by the key size but also by the number of participants. It is clear that there is an increase in time required to calculate a key if the key size is kept constant and there is a change in the number of participants. Graph depicting the effect of participants on time requirement is shown in Figure 4.3.

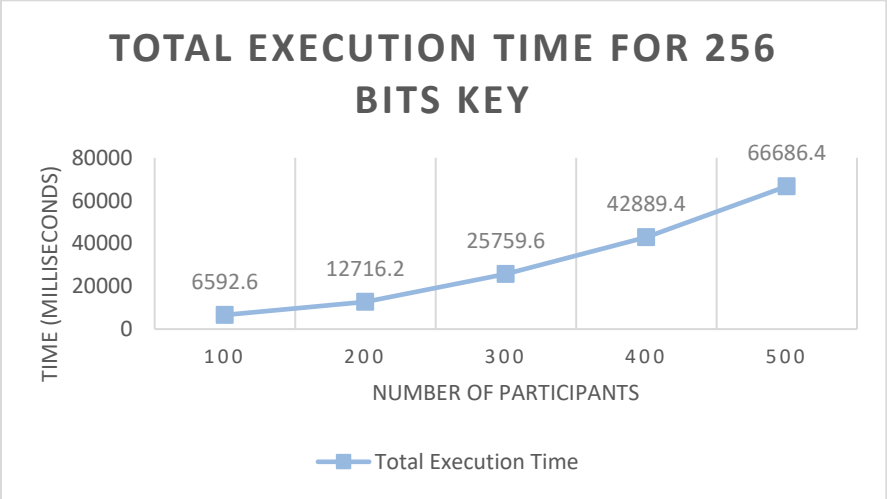

Figure 4.3: Graph Showing Time Taken by 256 Bits Key with The Constant Number of Participants

In the proposed scheme, HMAC has been used for authentication of key so that all the members can check if the final key they have calculated is correct or not. In the previous scheme no key authentication technique was incorporated. Analysis shows that the time required to

generate a key with key authentication technique requires more time compared to the standard key generation scheme as shown in Figure 4.4

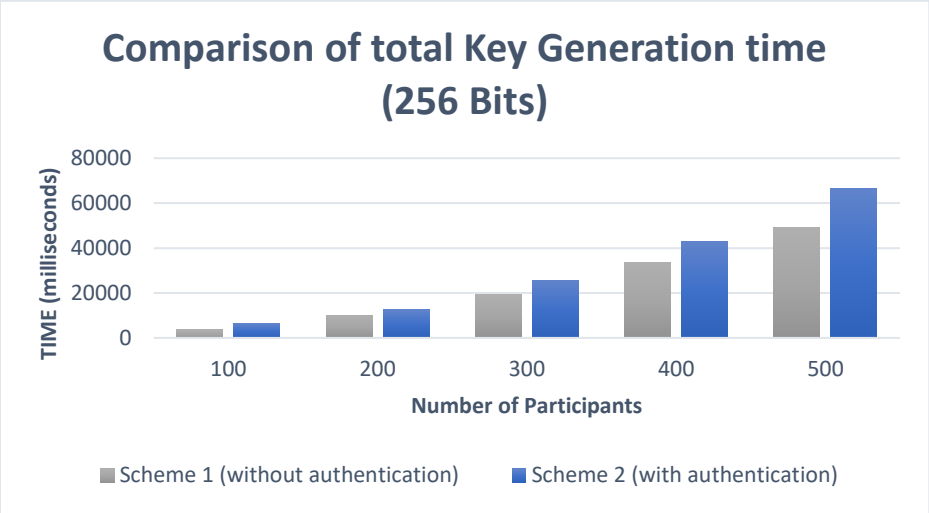

Figure 4.4: Graph Showing Comparison of Total Time Required for Key Generation with and Without Key Authentication Technique

This scheme has two basic functions. The first function is for taking contributions from all the group members and the second is to calculate the final key. The analysis show that time required by upFlow function and finalKey function with key authentication technique is more than that required by the module without key authentication technique as shown in Figure 4.5 and Figure 4.6.

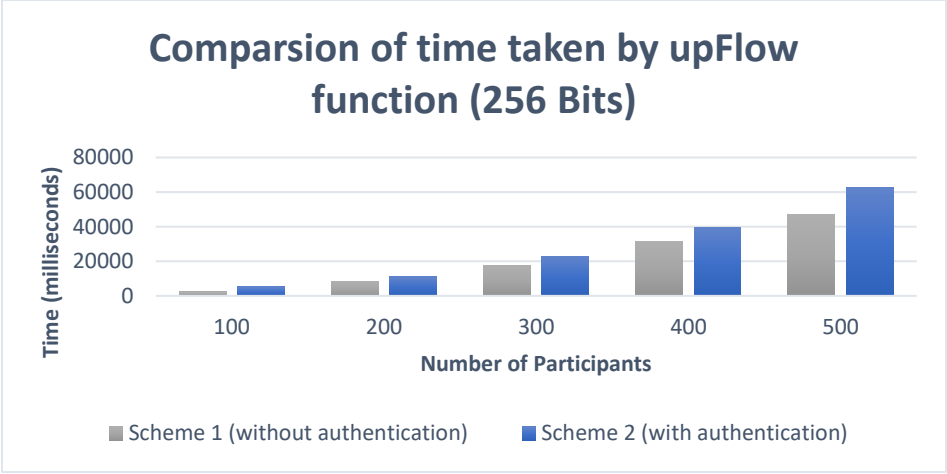

Figure 4.5: Graph Showing Comparison of Time Required for upFlow Function with and Without Key Authentication Technique

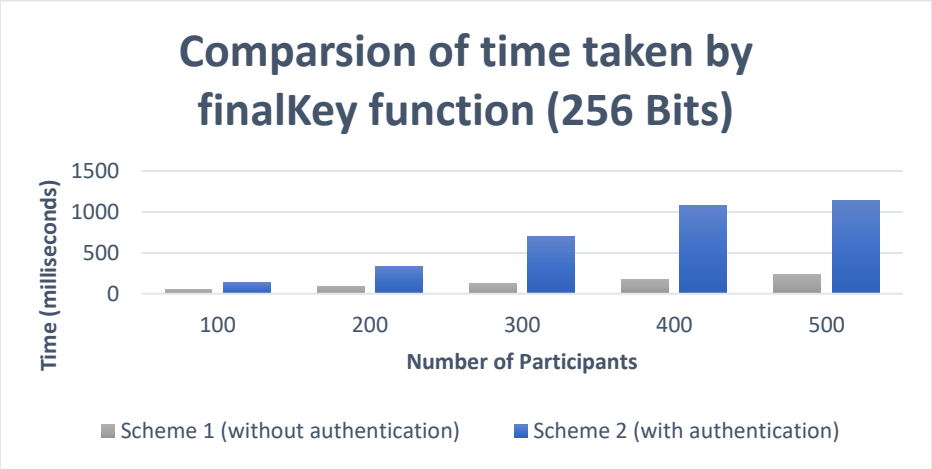

Figure 4.6: Graph Showing Comparison of Time Required for finalKey Function with and Without Key Authentication Technique

4.3 Summary

In this chapter the importance of key authentication and the issues with the existing schemes have been discussed. We have also discussed our Key distribution scheme with Key authentication. Key authentication is a critical element for secure group key distribution. To eliminate dishonest participants and to protect key theft, key authentication is very important in group communication. The chapter explains that a party can generate a symmetric group key based on PUF and use HMAC to authenticate the group key. The scheme proposed shows that by combining PUF and group Diffie-Hellman we can generate secure symmetric group keys and by using the HMAC we can achieve the group key authentication. For different key sizes and participant groups, the proposed scheme was simulated and tested. Based on previous results, we can see that the time required to generate a group key is affected by two factors: first, the size of the key, and second, the number of members in the group. If the size of the key increases and the members of the group configuration remain the same, the generation of the key will take longer. Similarly, if the size of the key remains the same and the group membership increases, the time required to generate the group key also increases. The analysis also showed that adding new members takes longer than deleting existing members. The analysis also shows that the time required for the "scheme 2" that passed the key authentication takes more time to calculate the key than the "scheme 1" that did not pass the key authentication.

CHAPTER 5: CONCLUSION

5.1 Conclusion

Currently, most encryption schemes rely on the key, and as long as the algorithm can be released, the key must be kept secret. Therefore, in most modern cryptographic schemes based on symmetric or asymmetric keys, whereas the success of the system lies in maintaining the confidentiality of the keys. This is in accordance with the Kerckhoff principle, which says that "only the secret of the key can guarantee security".

The cryptographic key is often a hexadecimal data block with a size of variable length. Because of its data type and size, people cannot memorize keys just like they would memorize passwords. In order to reuse the key when necessary, the keys are stored in the device [58] . The problem with storing keys on devices is that attackers can use different key attack/theft methods. Violent attacks can be reduced by lengthening the key [59] , but of course this does not prevent the key from being stolen. This is especially important in equipment systems with limited resources or poor performance, so it does not pose an obstacle for adversaries to find encryption keys. Adversaries can probe the physical device to obtain the key. Other methods can include booting into the system and obtaining the memory dump in an effort to locate the key. In no way is this a comprehensive list of possible attacks. To create a more effective cryptosystem resilient to key theft, cryptographers are currently considering other reliable sources, such as the physically unclonable function (PUF).

This research discusses the use of PUF for generating and distributing group keys. The aim of this research has been to provide heightened security to the group setting through the use of a physical root of trust.

Chapter 2 discusses the literature related to Cyber Physical Systems. In this chapter cryptography and its types are also discussed along with the cryptographic keys. Security concerns related to key theft and other key based attacks are also discussed in this chapter. The deterrent quality of PUF technology and the strengths of the technology have been brought to light in the chapter. Using PUF technology, devices can create identities based on device attributes. The device ID created using PUF is used to generate keys and use these keys to protect group communication. Since the key generated using PUF can only be used to prevent the key from being stolen, it can be discarded from system memory after use with all associated data. Since keys are not stored anywhere in the system, the adversary cannot obtain it using conventional key theft attacks. Because the PUF concept is rooted in the physical world, when an adversary wishes to attack a key mechanism, it must physically access the device and a dedicated probe/device to facilitate the evacuation. Even with this resource, the adversary will most probably fail because the PUF identity is not based on device serializations or predictable features.

In Chapter 3, key exchange between two parties is discussed. Group communication and group key exchange setup is also discussed in this chapter. The limitations of the existing group key exchange and distribution is also discussed in this chapter. This chapter also presents a novel scheme for group key generation that is based on the characteristics of PUF. The results and analysis of the time required for different operations of our scheme are also discussed in this chapter

Chapter 4 discusses key distribution and authentication mechanisms. The limitations of existing key distribution and authentication schemes are also discussed in this chapter. In this chapter detailed discussion about our key generation and authentication scheme for group communication based on PUF has been presented. The results and analysis of the time required for different operations of the scheme are also discussed in this chapter. The comparison of time required for key generation between the schemes with and without key authentication are also discussed in this chapter

In summary, this study shows that PUF provides a new core of trust based entirely on the physical attributes of the device. Conventional cryptography is based on algorithmic intractability which is not sufficient in securing the key. Through this research an attempt has

been made to shows that PUF ID can be alternatively of the stored key for cryptographic operations and secure service provisions. As a result, the key generation scheme for group communication can be based on PUF.

5.2 Future Work

The presented work has been simulated using the latest cryptographic libraries but lacks in implementation on physical devices from the IoT ecosystem. A simplified implementation can be done on a testbed of raspberry Pi.

Elliptic Curve Cryptography (ECC) is a public key method, which has attracted much attention because it has more benefits than RSA-based cryptosystems [60]. The future work that can be done with respect to this research is to create asymmetric group key generations using PUF and ECC.

Asymmetric group keys can be used in multiple applications like in digital currency, block chain, electronic voting schemes and many others. By creating an asymmetric group key based on PUF and ECC the security and the efficiency of the group communication can be increased significantly.

References

[1] L. Catarinucci *et al.*, "An IoT-Aware Architecture for Smart Healthcare Systems," *IEEE Internet Things J.*, vol. 2, no. 6, pp. 515–526, 2015.

[2] J. Lee, B. Bagheri, and H. A. Kao, "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems," *Manuf. Lett.*, vol. 3, pp. 18–23, Jan. 2015.

[3] N. Homma, S. Nagashima, Y. Imai, T. Aoki, and A. Satoh, "High-resolution side-channel attack using phase-based waveform matching," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006, vol. 4249 LNCS, pp. 187–200.

[4] P. Samarati *et al.*, "Kerckhoffs' Principle," in *Encyclopedia of Cryptography and Security*, 2011.

[5] "Managing Security Issues and the Hidden Dangers of Wearable Technologies - Google Books." [Online]. Available: https://books.google.com.pk/books?hl=en&lr=&id=yYXvDAAAQBAJ&oi=fnd&pg=PR1&dq=A.+Marrington,+D.+Kerr,+and+J.+Gammack,+Managing+Security+Issues+and+the+Hidden+Dangers+of+Wearable+Technologies+IGI+Global&ots=wmvfdP3Ns5&sig=j_4yea0pVXZy1GP1s7CTx5KZTig#v=onepa. [Accessed: 29-Apr-2020].

[6] J. A. Halderman *et al.*, "Lest we remember," *Commun. ACM*, vol. 52, no. 5, p. 91, May 2009.

[7] D. Genkin, L. Pachmanov, I. Pipman, A. Shamir, and E. Tromer, "Physical key extraction attacks on PCs," *Commun. ACM*, vol. 59, no. 6, pp. 70–79, 2016.

[8] C. J. Xue, G. Xing, Z. Yuan, Z. Shao, and E. Sha, "Joint Sleep Scheduling and Mode Assignment in Wireless Cyber-Physical Systems," 2009, pp. 1–6.

[9] J. Cao and H. Li, “Energy-efficient structuralized clustering for sensor-based cyber physical systems,” in *UIC-ATC 2009 - Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing in Conjunction with the UIC’09 and ATC’09 Conferences*, 2009, pp. 234–239.

[10] R. Baheti and H. Gill, “Cyber-physical Systems,” *Impact Control Technol.*, no. 1, pp. 161--166, 2011.

[11] R. H. Rawung and A. G. Putrada, “Cyber physical system: Paper survey,” in *Proceedings - 2014 International Conference on ICT for Smart Society: “Smart System Platform Development for City and Society, GoeSmart 2014”, ICISS 2014*, 2014, pp. 273–278.

[12] W. Wolf, “Cyber-physical systems,” *Computer (Long. Beach. Calif).*, vol. 42, no. 3, pp. 88–89, Mar. 2009.

[13] R. Baheti and H. Gill, “Cyber-physical Systems,” *Impact Control Technol.*, no. 1, pp. 161--166, 2011.

[14] J. Shi, J. Wan, H. Yan, and H. Suo, “A survey of Cyber-Physical Systems,” in *2011 International Conference on Wireless Communications and Signal Processing, WCSP 2011*, 2011.

[15] D. Davies, “A brief history of cryptography,” *Inf. Secur. Tech. Rep.*, vol. 2, no. 2, pp. 14–17, Jan. 1997.

[16] Y. Kumar, R. Munjal, and H. Sharma, “Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures,” *IJCSMS Int. J. Comput. Sci. Manag. Stud.*, vol. 11, no. 03, pp. 2231–5268, 2011.

[17] N. I. of S. and T. NIST, “Data Encryption Standard (DES),” *Fed. Inf. Process. Stand. Publ. (FIPS PUB 46-3)*, vol. 25, no. 10, pp. 1–22, 1999.

[18] J. Daemen, V. Rijmen, and K. U. Leuven, “AES Proposal : Rijndael,” *Complexity*, pp. 1–45, 1999.

[19] National Institute of Standards and Technology (NIST), *Federal Information Processing Standards Publication 197: Announcing the ADVANCED ENCRYPTION STANDARD (*

AES). 2001, p. 51.

[20] C. F. Kerry and P. D. Gallagher, “Digital Signature Standard (DSS),” 2013.

[21] S. H. Weingart, “Physical security devices for computer subsystems: A survey of attacks and defenses,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2000, vol. 1965 LNCS, pp. 302–317.

[22] F. Koeune and F.-X. Standaert, “A Tutorial on Physical Security and Side-Channel Attacks,” in *Springer*, 2005, pp. 78–108.

[23] V. Pasupathinathan, “Hardware-based Identification and Authentication Systems,” no. December, 2009.

[24] O. A. Osanaiye, “Short Paper: IP spoofing detection for preventing DDoS attack in Cloud Computing,” in *2015 18th International Conference on Intelligence in Next Generation Networks, ICIN 2015*, 2015, pp. 139–141.

[25] M. Mavani and K. Asawa, “Modeling and analyses of IP spoofing attack in 6LoWPAN network,” *Comput. Secur.*, vol. 70, pp. 95–110, Sep. 2017.

[26] E. Blumenthal and E. Weise, “Hacked home devices caused massive Internet outage,” *USA TODAY*, 2016. [Online]. Available: https://www.usatoday.com/story/tech/2016/10/21/cyber-attack-takes-down-east-coast-netflix-spotify-twitter/92507806/. [Accessed: 03-Aug-2018].

[27] L. Mathews, “A Frightening New Kind Of DDoS Attack Is Breaking Records,” *Forbes*, 2018. [Online]. Available: https://www.forbes.com/sites/leemathews/2018/03/07/a-frightening-new-kind-of-ddos-attack-is-breaking-records/#721558f878e0. [Accessed: 03-Aug-2018].

[28] A. K. Lenstra and E. R. Verheul, “Selecting cryptographic key sizes,” *J. Cryptol.*, vol. 14, no. 4, pp. 255–293, 2001.

[29] T. Lupu and V. Parvan, “Main Types of Attacks in Wireless Sensor Networks,” *WSEAS Int. Conf. Proceedings. Recent Adv. Comput. Eng.*, pp. 180–185, 2009.

[30] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, “Silicon physical random functions,” in *Proceedings of the 9th ACM conference on Computer and communications security - CCS ’02*, 2002, p. 148.

[31] C. Herder, M. D. Yu, F. Koushanfar, and S. Devadas, “Physical unclonable functions and applications: A tutorial,” *Proceedings of the IEEE*, vol. 102, no. 8. pp. 1126–1141, 2014.

[32] G. E. Suh and S. Devadas, “Physical unclonable functions for device authentication and secret key generation,” in *Proceedings - Design Automation Conference*, 2007, pp. 9–14.

[33] N. Beckmann and M. Potkonjak, “Hardware-based public-key cryptography with public physically unclonable functions,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, vol. 5806 LNCS, pp. 206–220.

[34] J. Murphy, G. Howells, and K. D. McDonald-Maier, “Multi-factor authentication using accelerometers for the Internet-of-Things,” in *Proceedings - 2017 7th International Conference on Emerging Security Technologies, EST 2017*, 2017, pp. 103–107.

[35] S. Feng and P. A. Lee, “Mesoscopic conductors and correlations in laser speckle patterns,” *Science (80-.).*, vol. 251, no. 4994, pp. 633–639, Feb. 1991.

[36] S. Devadas, E. Suh, S. Paral, R. S.- RFID, 2008 IEEE, and undefined 2008, “Design and implementation of PUF-based" unclonable" RFID ICs for anti-counterfeiting and security applications,” *people.csail.mit.edu*.

[37] H. Delfs and H. Knebl, “Symmetric-Key Cryptography,” Springer, Berlin, Heidelberg, 2015, pp. 11–48.

[38] E. Fujisaki and T. Okamoto, “Secure integration of asymmetric and symmetric encryption schemes,” *J. Cryptol.*, vol. 26, no. 1, pp. 80–101, 2013.

[39] W. Diffie, W. Diffie, and M. E. Hellman, “New Directions in Cryptography,” *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[40] S. (Shafi) Goldwasser, L. Strawczynski, W. Diffie, and M. Wiener, *Advances in cryptology--CRYPTO ’88 : proceedings*. Springer-Verlag, 1990.

[41] I. Ingemarsson, D. Tang, and C. Wong, “A conference key distribution system,” *IEEE Trans. Inf. Theory*, vol. 28, no. 5, pp. 714–720, Sep. 1982.

[42] H. Harney and C. Muckenhirn, “Group key management protocol (GKMP) architecture,” 1997.

[43] M. Steiner, G. Tsudik, and M. Waidner, “Diffie-Hellman key distribution extended to group communication,” in *Proceedings of the 3rd ACM conference on Computer and communications security - CCS ’96*, 1996, pp. 31–37.

[44] K. Arnold, *The Java Programming Language, Second Edition*. 1998.

[45] Oracle Corporation, “NetBeans IDE Download,” *NetBeans IDE 8.2 Download*, 2012. [Online]. Available: https://netbeans.org/downloads/. [Accessed: 28-Oct-2018].

[46] “hjp: doc: RFC 2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile.” [Online]. Available: http://www.hjp.at/doc/rfc/rfc2459.html. [Accessed: 15-Feb-2020].

[47] D. Davis and R. Swick, “Network security via private-key certificates,” *Oper. Syst. Rev.*, 1990.

[48] M. Bala Krishna and M. N. Doja, “Symmetric key management and distribution techniques in wireless Ad Hoc networks,” in *Proceedings - 2011 International Conference on Computational Intelligence and Communication Systems, CICN 2011*, 2011.

[49] J. J. Tardo and K. Alagappan, “SPX: Global authentication using public key certificates,” *J. Comput. Secur.*, 1992.

[50] M. Just, “Needham–Schroeder Protocols,” in *Encyclopedia of Cryptography and Security*, 2006.

[51] Avi Kak, “No Lecture 10: Key Distribution for Symmetric Key Cryptography and Generating Random NumbersTitle,” 2020. [Online]. Available: https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture10.pdf.

[52] MIT, “Kerberos: The Network Authentication Protocol,” *2013*. 2013.

[53] J. Steiner, B. Neuman, and J. Schiller, “Kerberos: An Authentication Service for Open

Network Systems.," *USENIX Winter*, 1988.

[54] "Kerberos Authentication Overview | Microsoft Docs." [Online]. Available: https://docs.microsoft.com/en-us/windows-server/security/kerberos/kerberos-authentication-overview. [Accessed: 11-Apr-2020].

[55] M. Soria-Machado, "CERT-EU Security Whitepaper 2014-007 Kerberos Golden Ticket Protection Mitigating Pass-the-Ticket on Active Directory," 2016.

[56] I. Ingemarsson, D. T. Tang, and C. K. Wong, "A Conference Key Distribution System," *IEEE Trans. Inf. Theory*, vol. 28, no. 5, pp. 714–720, 1982.

[57] A. Shaikh, "PPT: Group Key Management Protocol (GKMP)." .

[58] I. Kizhatov, "Physical Security of Cryptographic Algorithm Implementations," *PhD Thesis*, 2011.

[59] Y. Xiao and Y. Pan, *Security in Distributed and Networking Systems*. 2007.

[60] K. Lauter, "The Advantages of Elliptic Curve Cryptography for Wireless Security," *IEEE Wireless Communications*, vol. 11, no. 1. pp. 62–67, 2004.