

Name: **PANDU RANGA REDDY KONALA**

Student ID: **35238559**

Dissertation Title: **Cryptographically Secure On-Line Identity System**

Module: **SCC.420: DISSERTATION**

Date: **01-09-2020**

I certify that the material contained in this dissertation is my own work and does not contain unreferenced or unacknowledged material. I also warrant that the above statement applies to the implementation of the project and all associated documentation. Regarding the electronically submitted version of this submitted work, I consent to this being stored electronically and copied for assessment purposes, including the Department's use of plagiarism detection systems in order to check the integrity of assessed work. I agree to my dissertation being placed in the public domain, with my name explicitly included as the author of the work.

Date: 01-09-2020

Signed: Pandu Konala

TABLE OF CONTENTS

ACKNOWLEDGEMENT	i
ABSTRACT	ii
LIST OF TABLES	iii
LIST OF FIGURES	iv
Chapter 1: Introduction	1
1.1 Overview	1
1.2 Project Prologue & Purpose	1
1.3 Project Aim & Objectives	2
1.4 Project Structure	3
Chapter 2: Background Research	3
2.1 Introduction of PUF	3
2.2 Characteristics of a PUF.....	3
2.2.1 Challenge and Response	4
2.2.2 Features of a PUF	4
2.2.3 Environmental factors effects on PUF's	6
2.3 Types of PUF's	6
2.3.1 Non-Silicon PUF's	7
2.3.2 Delay based Intrinsic PUF's	8
2.3.3 Memory based Intrinsic PUF's	10
2.4 Security Level of PUF's	12
2.4.1 Strong PUF's	12
2.4.2 Weak PUF's	13
2.4.3 Controlled PUF's	13
2.5 Security application of PUF's	14
2.5.1 Device Identification	14
2.5.2 Key generation and storage	14
2.5.3 IC Protection	14
2.5.4 Protocols which uses CRP's	14
2.5.5 Timed Authentication	15
2.5.6 Software Licensing	15
2.6 Security Schemes for IoT Devices	15
2.6.1 Authentication Protocols	16

2.6.2 Types of IoT Authentication Schemes	21
2.7 Usage of PUF's for Authentication	21
2.7.1 Challenges faced by PUF based authentication	24
2.8 Quick Summary	25
Chapter 3: Design and Implementation	26
3.1 Introduction	26
3.2 About PUF designed by Quantum Base	26
3.3 Project Design	26
3.3.1 High level Overview	26
3.3.2 Overview of Message Sequence	28
3.3.3 Project Architecture	32
3.3.4 About the Data Stores	43
3.4 Cryptographic Standards implemented in the Framework	45
3.5 Implementation	46
3.5.1 System Specifications	46
3.5.2 Implemented Framework	47
Chapter 4: Framework Analysis and Testing	49
4.1 Framework evaluation	49
4.2 Performance evaluation	49
4.3 Network Analysis	54
4.4 Security Analysis	55
4.4.1 Overview of STRIDE and DREAD	55
4.4.2 STRIDE Analysis	55
4.4.3 DREAD Analysis	57
Chapter 5: Conclusion	60
5.1 Project Implications	60
5.2 Limitations & Future Work	61
References	62

ACKNOWLEDGEMENT

I express my gratitude to **Lancaster University** for providing a bright academic climate which has made this entire task appreciable. This acknowledgement is intended to be thanks giving measure to all those people involved directly or indirectly with my project.

I feel extremely grateful to **Quantum Base** for funding this project. I sincerely thank **David Howarth** for his support, feedback and encouragement which helped me to complete the project in time.

I would like to thank my supervisor, **Dr. Daniel Prince**, whose expertise was invaluable in formulating the research and methodology. Your support and feedbacks had pushed me to sharpen my thinking and brought my work to a higher level.

I would like to extend my sincere thanks to my family and friends for helping and motivating me during the course of the project. Last but not the least, I thank **God** for his blessings which made my project a success.

ABSTRACT

A new era of communication among the devices paved way for a modern technology called Internet of Things (IoT). In the initial days, it was confined with a single device by sharing the data within its internal components which was termed as embedded systems. As it evolved, the scope of IoT has expanded by transferring data through various networks. Since the IoT devices consume less power and low bandwidth, the researcher not paid attention towards privacy, security and efficient use of the devices which has lead to compromise between data security and usage of resources.

In traditional approach, Symmetric and Asymmetric algorithms were used to provide data security. In Symmetric Key Cryptography, if the master key is compromised which leads to the failure of the protocol similarly by using Asymmetric Key Cryptography is not resource efficient for IoT devices. The framework allows the use of the Quantum Base's unique and Unclonable QID for authenticated cryptographic operations.

A framework is proposed for providing a secure way of authenticated data transfer to the consumer. This framework utilizes QIDs for generating master key which will be used for generating various cryptographic keys for secure transmission. One of the core feature of this frame work is its modularity as users can choose between different security profiles installed in the frame work. The project is designed to utilize the unique properties of QIDs for various cryptographical processes in order to securely fetch the data present in the QID Device. To achieve this, the project is focused on the three entities such as the Consumer, QB Server and QID Device which possess multiple functionalities.

The Cryptographically Secure On-Line Identity System was evaluated for network analysis to determine whether an attacker is able to recreate the cryptographic keys through the response packets of the QID device. When analysed it is found that the attacker was unsuccessful in recreating the cryptographic keys. Similarly to identify the vulnerabilities and threats posed for this frame work, a security analysis is carried out using STRIDE and DREAD analysis. In the security analysis it is found that the QB server is more likely to be targeted by the attacker. The memory utilization is very low for the operations that are taken place in the QID devices which shows that the resource utilization and performance is found to very efficient.

LIST OF TABLES

Table 2.1: Usage of PUF's in different schemes

Table 4.1: STRIDE Analysis

Table 4.2: DREAD Analysis

LIST OF FIGURES

- Figure. 2.1: Challenge and Response
- Figure. 2.2: Inter Distance Measure
- Figure. 2.3: Intra Distance Measure
- Figure. 2.4: Unpredictability Model
- Figure. 2.5: Optical PUF
- Figure. 2.6: Ring Oscillator PUF
- Figure. 2.7: Arbiter PUF
- Figure. 2.8: SRAM PUF
- Figure. 2.9: SRAM PUF
- Figure. 2.10: SAML Model
- Figure. 2.11: OpenID + OAuth2 Model
- Figure. 2.12: RADIUS Authentication Model
- Figure. 2.13: SQRL Authentication Model
- Figure. 2.14: PUF Authentication
- Figure. 2.15: PUF Authentication using homomorphic encryption
- Figure. 3.1: High Level Design Diagram
- Figure. 3.2: Sequence Diagram
- Figure. 3.3: Consumer to QB Server
- Figure. 3.4: QB Server to QID Device
- Figure. 3.5: QID Device to QB Server
- Figure. 3.6: QB Server to Consumer
- Figure. 3.7: Consumer to QB Server and vice versa
- Figure. 3.8: Project Architecture

Figure. 3.9: Stage – 1 Data Process

Figure. 3.10: Stage – 2 QB Server Mapping and Transmitting

Figure. 3.11: Data Fetching

Figure. 3.12: Cryptographic Operations

Figure. 3.13: Stage – 3 QID Device Operations

Figure. 3.14: Stage – 4 QB Server Operations

Figure. 3.15: Stage – 5 Final Operations

Figure. 3.16: QB Server Storage

Figure. 3.17: QID Device Storage

Figure. 3.18: Consumer Interface

Figure. 3.19: Data Display to Consumer

Figure. 3.20: QB Server Test Run

Figure. 3.21: QID Device Test Run

Figure. 4.1: Single Request analysis of profile-1

Figure. 4.2: Multi Request analysis of profile-1

Figure. 4.3: Single Request analysis of profile-2

Figure. 4.4: Multiple request analysis of profile-2

Figure. 4.5: Single request analysis of profile-3

Figure. 4.6: Multiple request analysis of profile-3

Figure. 4.7: Single request analysis of profile-4

Figure. 4.8: Multiple request analysis of profile-4

Figure. 4.9: STRIDE Threat Model

Chapter 1 : Introduction

1.1. Overview

With the advent of Internet for communication and devices becoming smarter had led to a new technology called Internet of Things (IoT). In the initial days, it was confined to a single device by sharing the data within its internal components. As it evolved the scope of IoT has expanded by transferring data through various networks. Since the IoT devices consume less power and low bandwidth, the researcher not paid attention towards privacy and security. There are various methods for provisioning of data security in IoT, but which are high energy consuming. This chapter discusses the overall idea of the project, which includes the aim, objectives, prologue and an overview of each chapter.

1.2. Project Prologue & Purpose

The IoT is rapidly changing the surroundings of the world we live in. Sophisticated sensors and digital control systems are embedded in the physical chips around us where each chip transmits valuable data. This data helps us better understand the collective usage of IoT systems. However, IoT is becoming more and more complex and may have serious consequences in the areas of privacy and security. The current way of the data transfer between IoT device and the consumer is expensive. Since IoT devices are resource constrained, due to this data security is often given least important or by maintaining with low data security standards. To overcome these issues, it is proposed to use a Quantum Base's unique and unclonable functions to achieve the data security at a higher level with a low computational cost.

Traditionally, the IoT systems use a Symmetric key/Asymmetric key encryption for cryptographic process. Authorization, Authentication, Encryption and Digital Signature are performed on IoT devices. When analyzed these operations consumes lot of resources. The designed framework of this project can be used on various IoT devices. With low overhead and high data security, This project is suitable for various types of IoT applications such as smart city or smart home, wearable automation, and much more.

1.3. Project Aim & Objectives

This project is to develop and implement a cryptographically secure on-line identity system which deploys Quantum Base's unique and unclonable quantum identities to digital identities. The objectives of this project are as follows:

- To design a framework that uses Quantum Base's quantum identities for IoT
- Produce a proof of concept of the designed framework
- Conduct performance and security analysis on designed framework

1.4. Project Structure

Chapter 1: Introduction – This chapter provides an overview of IoT, the problem definition and project objectives.

Chapter 2 : Background Research – This chapter showcases the research undertaken to design the framework. Investigates existing techniques of quantum identities in order to develop the framework.

Chapter 3 : Design & Implementation – The lifecycle of the framework is briefly discussed in this chapter along with the implementation.

Chapter 4 : Framework Analysis and Testing – This chapter focuses on threat modelling of the framework with necessary recommendations to limit the impact of the threats.

Chapter 5 : Conclusion – In this chapter, The complete analysis of the progress is discussed. Limitations and Further developments are also discussed in this chapter.

Chapter 2 : Background Research

2.1. Introduction of PUF

Physically Unclonable Function (PUF) [1] is a concept of implementing various critical security concepts such as authentication and identification through hardware properties present in a system. PUF's are used in various sectors such as technology, manufacturing and defence sector where the sensitivity of the data is crucial. The generic way of authenticating the edge devices pose various security challenges. Traditionally, these cryptographic schemes are used to store the underlying cryptographic secrets i.e. Private keys, Hard coded passwords. Security architects used technologies like Trusted Platform Module (TPM), Hardware Security Module (HSM) which are used for secure key storage. These platforms are very expensive and had a vulnerability in the design aspect. In some designs the private keys were stored in the EEPROM's of the edge device and used public key cryptography for data security and communication. This system design is not suitable for several small scale devices due to high energy consumption of the cryptographic process. To cover up all the drawbacks of the traditional system, PUF's were used for secure key identity and generation process. The unique characteristics of PUF makes it as an alternative to the above mentioned traditional security modules and resistant to various hardware security attacks. The advantages of PUF's are as follows:

- No stored keys
- No public key cryptography
- Cannot be cloned / copied
- Uses nano-scale variations which is unique for each PUF
- Cost efficient

This chapter focuses a variety of aspects of a PUF such as the characteristics, types of PUF's, classification and corresponding application in the field of computer security. A study had been performed on widely used authentication protocols in the current world and the usage of PUF's for authenticated cryptographic operations in the realm of IOT technologies.

2.2. Characteristics of a PUF

PUF's possess distinctive characteristics that make it exclusive compared with traditional methodologies. These traits make it useful for devices which requires security from the hardware level. In Sections 2.2.1, 2.2.2, 2.2.3 highlights the various characteristics of the PUF's.

2.2.1. Challenge and Response

A PUF can be defined as a unique function which takes in a challenge and gives out a unique response. This means that the output depends on the input as well as the device executing it. This makes the responses unique for every device thus useful in generating unique identities which acts like a physical fingerprint of the device as shown in Figure 2.1.

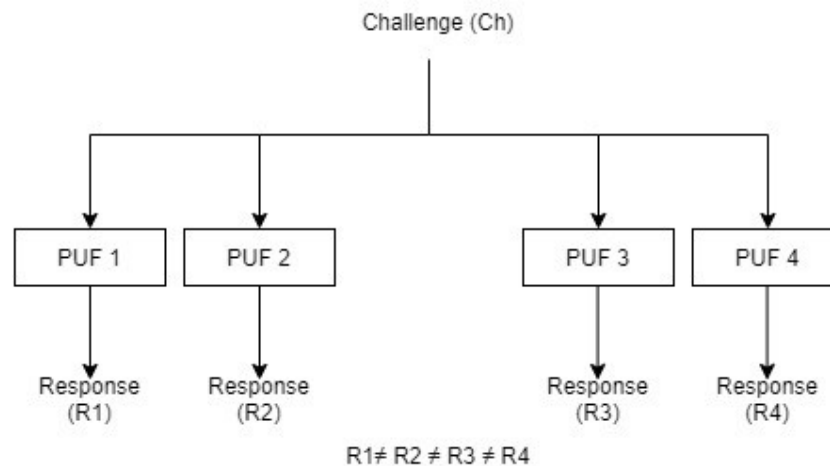


Figure.2.1: Challenge and Response

The challenge is an electrical stimulus which is applied to the physical structure of the PUF. After the complex interactions with physical structure of the PUF's, a unique and Unclonable response is obtained as an interaction. The applied challenge and the obtained response are known as challenge-response pairs (CRP). The response depends on various factors like manufacturing process, input electric stimulus and the device in which responses are being generated this makes the output unpredictable and unclonable. Since there is no need for any stored keys which enables a secure binding of secrets with various software, protocols and underlying hardware. Also due to the low overhead of PUF's makes it very useful in IOT applications. In current world, PUF's are used in anti-counterfeiting, protection of Intellectual Property (IP) and Radio frequency identification systems (RFID).

2.2.2. Features of a PUF

Fundamentally, The PUF's are used for unique identification. During the manufacturing process, Some PUF's might be faulty and might not give out an intended response when an electrical stimulus is applied. In order to test the quality of the PUF's, Manufacturers test the PUF's to possess features such as Uniqueness, Reliability and Unpredictability. These three

features can be measured by observing physical properties of PUF's such as Inter and Intra Distances.

- **Uniqueness**—Uniqueness can be measured by calculating the Inter Distances. During this process, a common challenge is sent to two or more devices which are exactly identical. The response given out by the PUF in the corresponding devices should be unique. As shown in Figure 2.2.

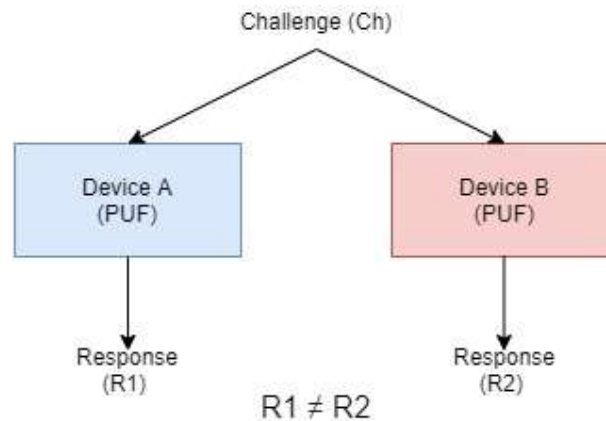


Figure. 2.2: Inter Distance Measure [9]

- **Reliability** –The reliability is measured using Intra-Distances where a challenge is sent to device multiple times to get back similar responses. It is expected that the differences between the responses must be similar. As shown in Figure 2.3.

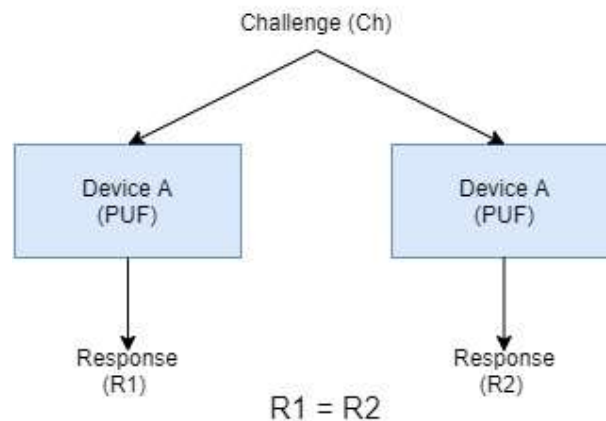


Figure. 2.3: Intra Distance Measure [9]

- **Unpredictability** – The ability to provide a correct response to a challenge requires the presence of the device rather than being generated by the algorithm or other human interaction. This makes it difficult to predict the output of a PUF to a randomly chosen challenge. As shown in Figure 2.4.

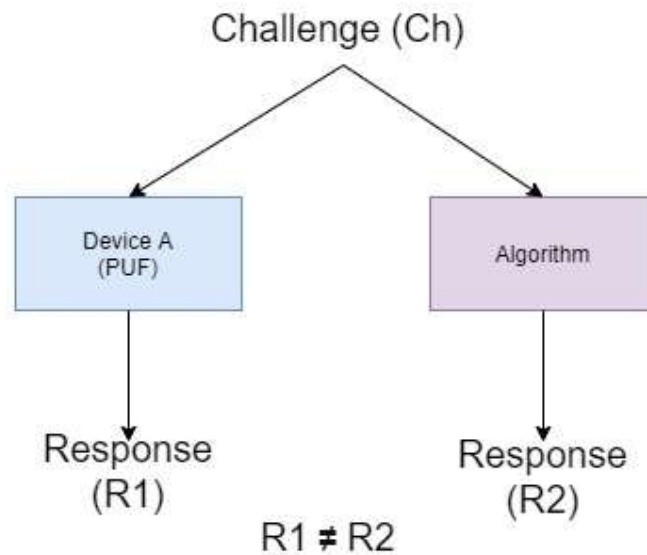


Figure. 2.4: Unpredictability Model [9]

2.2.3. Environmental factors effects on PUF's

PUF's use their physical properties to generate the response. The generated response depends on factors like temperature, power supply and crosstalk [2]. These factors interfere with the features of PUF's pointed in Sect. 2.2. The interferences cause disturbances during the intra distance measure phase which makes the responses appear scrambled. These environmental factors can be reduced using technical methods such as compensation [3], pre-response selection [4] and using onboard sensors [5]. However, there are various types of PUF's available in the market which are resistant to some of the environmental factors.

2.3. Types of PUF's

PUF's were initially developed to prevent the exploitation of authentication by malicious actors back in 1983 [6]. After its initial success in the field of computer security many institutions came up with various types of PUF's. Based on their physical properties, working mechanisms and architecture, PUF's can be classified into three types[9][14]:

- Non-Silicon PUF's
- Delay based Intrinsic PUF's
- Memory based Intrinsic PUF's

2.3.1. Non-Silicon PUF's

These PUF's fall into a category where their functions or the architecture to generate a response which does not depend on the electronics. However most of their functionality depends on the nature of random primitive generation and other electronic components are used for processing and storage purposes.

- **Optical PUF**

This is a technique which is achieved by using laser on a material to cause a random optical reflection pattern. This obtained pattern is known as a "Reflective Particle Tag (RPT)" which is unique for every iteration. This method is widely used to securely generate a PUF. Optical PUF's offer high security against cloning and modelling attacks. In an economic point of view the cost per piece is inexpensive. As it uses plastic platelet which does not contain any electronic or circuitry to output a complex optical interference process [7] [15].

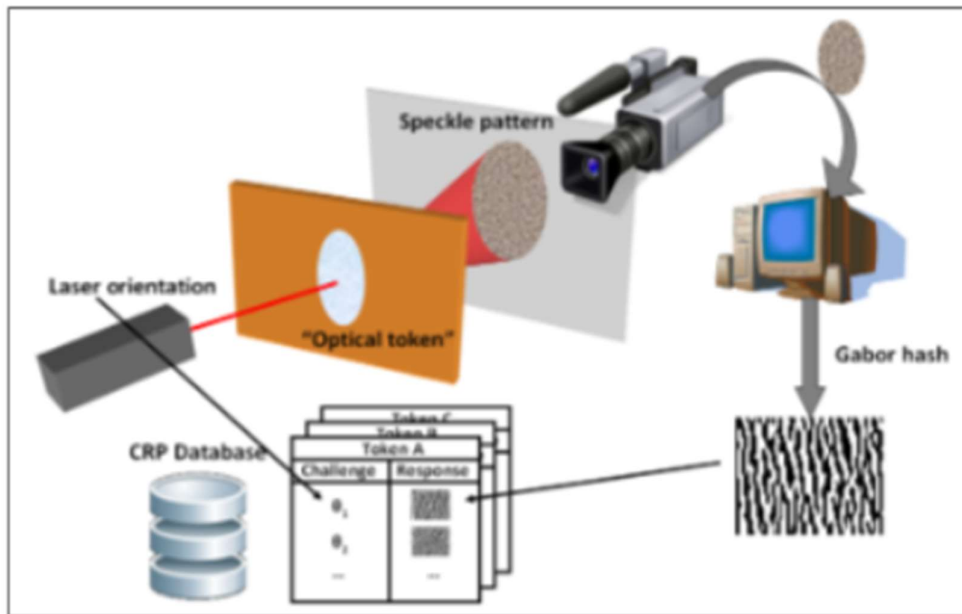


Figure. 2.5: Optical PUF [9]

However, There is a minute disadvantage to the outputs generated by the optical PUF's. In order to obtain the responses, a malicious actors must establish a same distance for the light scattering process, laser beam, charge coupled device camera for each and every response in an attempt to clone a single PUF as shown in Figure 2.5. This process is complex, expensive, time consuming and error prone due to environmental factors[7] [15].

- **Compact Disc PUF**

At the time of manufacturing, Compact Discs have lands and pits which are random and unique for every disc. It is highly unlikely that the lands and pits are common for two discs. This can be taken as an advantage to support the operations performed by a PUF [9].

- **Acoustical PUF**

Generating the responses using physical vibrations made by the delay in electrical signals are known as Acoustical PUF's [8]. Since electric signals and material vibrations being random and unique, Enough entropy can be extracted for responses. However, The false rejection and acceptance rate is arguable making it less useful than the above.

These are some of the most commonly used Non Electronic PUF's. There are few PUF's developed but not used in the real world because of high production costs and complexity. Few notable mentions are Paper PUF's, Magnetic PUF's and Radio frequency PUF's [8].

2.3.2. Delay based Intrinsic PUF's

A lot of PUF's that are proposed in the last 15 to 20 years. Most common are the Intrinsic PUF's which are often used in modern world technology. These are the PUF's that can be completely implemented within a chip i.e. The PUF function, measurement circuit and post processing is also present within that single chip e.g. Most silicon based PUF's.

- **Ring Oscillator PUF**

This type of PUF constitutes of an "ENABLE" and an AND gate. Whenever there is a signal set to "ENABLE" to set its value to 1, The corresponding AND gate would pass the signal through [9]. Also, There are odd number of inverters that are connected to a feedback from output to the initial AND gate. This function changes the output continuously from 0 to 1 and vice versa. So, It essentially creates a periodic output of 1 and 0. As shown in Figure 2.6.

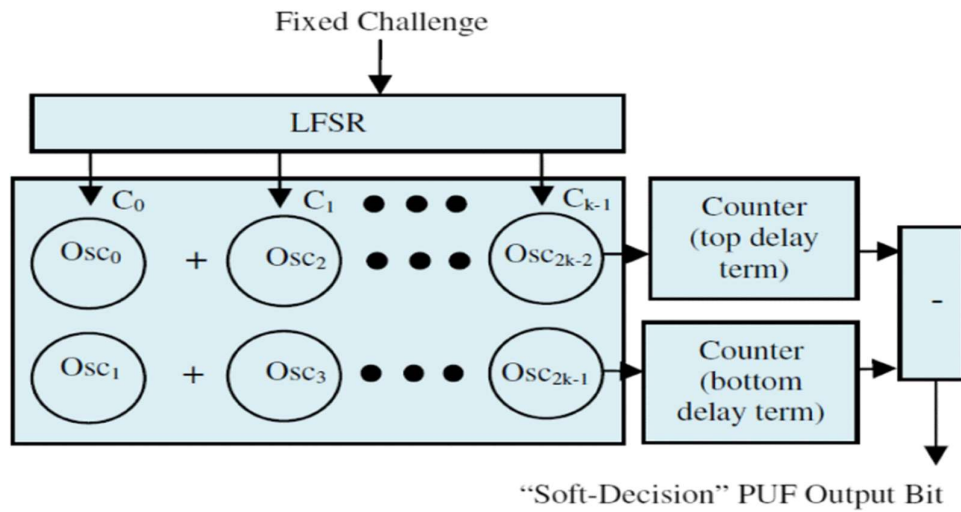


Figure. 2.6: Ring Oscillator PUF [15]

The frequency of the output varies on multiple factors. Firstly, It depends on number of NOT gates that are present in this ring oscillator i.e. More inverters correspond to lower frequency because the signal takes a longer time to propagate to the output. Secondly, The delay in each stage also effects the output. For example, If the delay of each inverter present in the circuit is very short then the signal would take a shorter time to reach the output and as a result higher frequency can be achieved. On the contrary, If the delay of each inverter is a long then the frequency of the ring oscillator output will be much smaller [15]. These variations can also occur at the time of fabrication process of the gates and circuit.

- **Arbiter PUF**

Fundamentally, The working principle of an Arbiter PUF depends on a switch. The switch contains two multiplexers. The two multiplexers are given same input. Based on the input either 0 or 1, it will switch that corresponding input value to that output. For example, If the multiplexer select line is 0 then the input at the multiplexer is sent to the output. On the other hand, If the select line if the multiplexer is set to 1 then the input present at the 2nd input would be switched at the output as shown in Figure 2.7. So this component called arbiter switch is used to build an Arbiter PUF [16]. The fundamental feature about this particular switch that makes it interesting for the Arbiter PUF is that these outputs whether 1 or 0 depends on the characteristics of these PUF's.

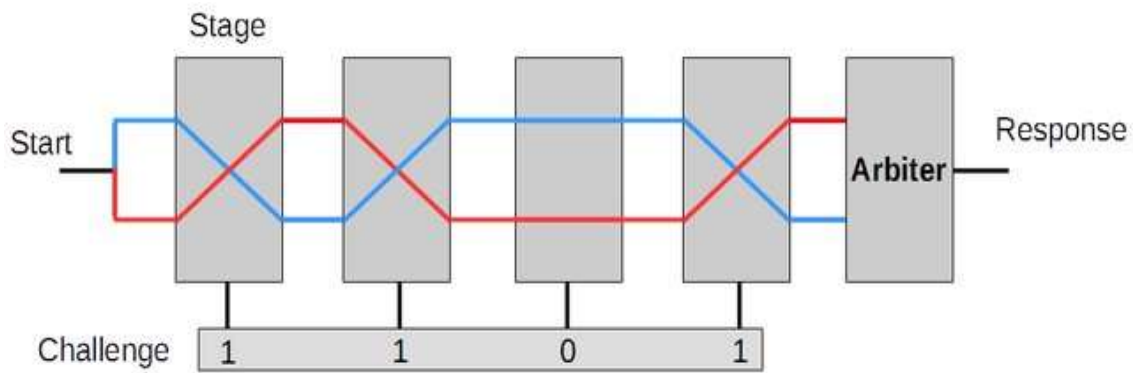


Figure. 2.7: Arbiter PUF [16]

Generally, An Arbiter PUF consists of multiple switches connected to each other and a single input which is fed to both switches. The challenge play an important role for operation of the switches. So, There are essentially two aspects that make this design interesting for the use as a PUF. Firstly, The signals propagating through the switches differs on the nano scale variations of their design [15]. These variations effect the transmission speed of the signals which is similar to the Ring Oscillator PUF's. The delays provided by each of the multiplexers is influenced by the manufacturing processes and various intrinsic properties of the silicon and process used to create the PUF. Secondly, A "D Flip-Flop "acts like an arbiter is present at the output of the switches to measure which transmission line is faster [15]. The flip-flop passes the value with fastest transmission rate and the value changes on factors like temperature, electric charge and challenge used.

2.3.3. Memory based Intrinsic PUF's

These are silicon based PUF's which take the advantage of the memory states of a digital circuit. Static Random Access Memory (SRAM) PUF's and Latch PUF's fall under this category.

- **SRAM PUF**

A simple SRAM consists if a cross coupled inverters [17] as shown in Figure 2.8. For example, If the input to the inverters is 1 the output has to be 0. This process is achieved by stabilizing the circuit using feedback loops back to the input of the inverters and similarly with input 0. The idea behind SRAM is during the system's initial power off stage the input signals to the SRAM are random therefore the output will also be random. Also the initial value of each cell in the SRAM will also be random and this can be used to build an SRAM PUF's.

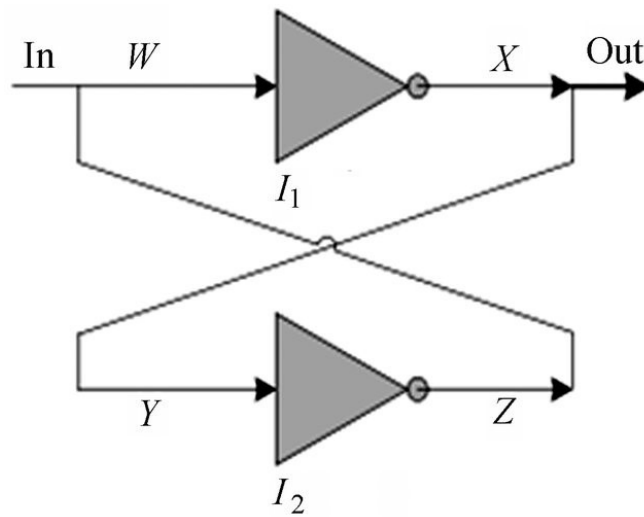


Figure. 2.8: SRAM PUF [17]

- **Latch PUF**

Similarly like an SRAM PUF architecture, Latches can be used to define a path bit. Latch PUF uses a symmetric circuit to change between different state [17]. This symmetric nature produces different output values with even with a small change in input as shown in Figure 2.9. Also, Latch consists of minimum of 2 gates with different intrinsic properties and transmission speed thus creating a randomness for each state.

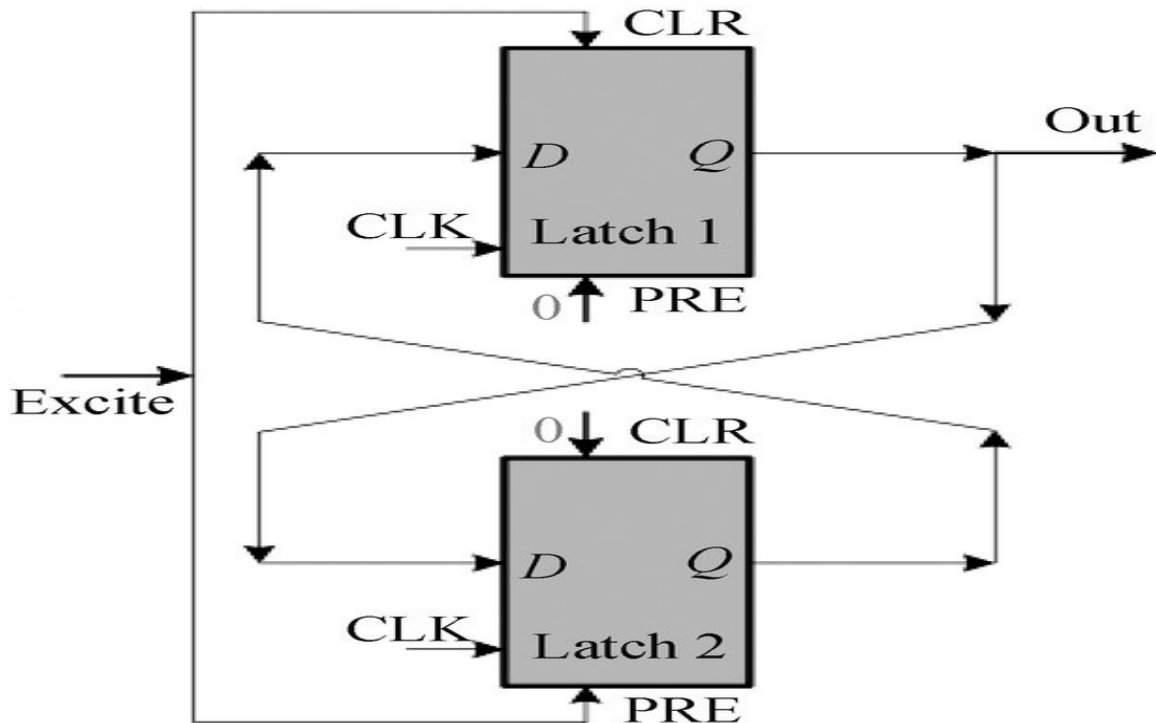


Figure. 2.9: SRAM PUF [17]

- **Butterfly and Flip-Flop PUF's**

The disadvantage with the above mentioned memory based PUF's is that they cannot be used for Field Programmable Gate Array (FPGA) due to high initial power up values. Since during an FPGA start-up the configuration stream will hard reset all the SRAM cells to zero. This removes randomness required to construct a PUF. In order to make a PUF that work for FPGA, Researchers have come up with Butterfly [17] and Flip-Flop PUF's [9].

The Butterfly PUF mimics the behaviour of the SRAM cells to bring back the randomness before the start up sequence of FPGA's. This is achieved by cross coupling latches using a FPGA reconfigurable logical circuit. The randomness stored by the Butterfly PUF depends on factors like temperature and fabrication processes. Similar to Butterfly PUF's, The Flip-Flop PUF's uses the concept of flip-flop instead of latches to mimic the behaviour of SRAM cells.

2.4. Security Level of PUF's

As discussed above the widely used PUF's in the current world, Each of the PUF's possess various properties and are useful for a dedicated task alone. The security levels of a PUF can be classified into three types[9] [15].

- Strong PUF's
- Weak PUF's
- Controlled PUF's

2.4.1. Strong PUF's

For a PUF to fall into this category it should be able to solve complicated challenges and produce unique response [15]. The properties as discussed in Sect.2.2 should thoroughly follow. Also if a malicious actor has complete access to the CRP data, It has to be impossible for him to reproduce similar response for a given challenge for given time frame.

An example for a Strong PUF's is an Arbiter PUF. In an Arbiter PUF, Select lines are used to choose a challenge and if there are N such switches which are present then the number of challenges possible are 2^N . Also, The CRP's are exponentially related to the number of components in the case of a strong PUF [15]. Strong PUF's are used for authentication, identification and token generation for high level applications. Some of the properties of a strong PUF's are as follows:

- They possess a huge number of CRP's
- It is assumed that an attacker cannot enumerate all the CRP's within a fixed time interval. Therefore CRP's can be made public.
- An attacker given a polynomial sized sample of adaptively chosen CRP's cannot predict the response to a new randomly chosen challenge.
- Also, there is no need for a cryptographic scheme because there is nothing secret which needs to be stored.

2.4.2. Weak PUF's

Weak PUF's means that they support very limited number of challenges [15]. The responses obtained from these PUF's can also be effected by environmental factors. This increases the rate of errors of a PUF. Weak PUF's are used for cryptographic key derivations which acts as a secure secret key known only to the system in order for storage and communication for a high level application.

An example for a Weak PUF's is an Ring Oscillator PUF. As mentioned above in Sect.3.2.1, Ring oscillator PUF constitutes of multiplexers and an N-bit challenge. Essentially the N-bit challenge is choosing a pair of multiplexers and therefore the number of challenges possible are $C(n,2)$ i.e. N choose 2. The generated CRP's are linearly related to the number of components in an Ring Oscillator PUF [15]. Some of the properties of a weak PUF's are as follows:

- They have very good inter and intra differences.
- They have comparatively few CRP's. As discussed in the case of ring oscillator.
- The CRP's have to be kept secret and of the low number. Measures have to be taken to ensure that an attacker should not be able to enumerate all possible CRP's.
- Typically used along with a cryptographic scheme like encryption, HMAC, etc. In order to hide the CRP to maintain confidentiality.

2.4.3. Controlled PUF's

The concept of controlled PUF is to use a strong PUF with the combination of a control circuits. An additional logic can be used to control the CRP's obtained from the PUF. This method only allows instructions from explicitly authorized trusted entities [10]. These are proven to be cost effective and provide resistance to various attacks. Cryptographic functionalities such as key exchange and mutual authentication is supported [10].

2.5. Security application aspects of PUF's

After understanding the behaviour and properties of various PUF's, Their applications in the field of computer security can be revolutionary. Some of the major security applications of PUF's that are currently being are as follows:

- Device Identification
- Key generation and Storage
- IC Protection
- Protocols which uses CRP's
- Timed Authentication
- Software licensing

2.5.1. Device Identification

PUF's can be used for device identification [18]. This is based on the observation that the same PUF circuitry generates different and unique PUF data on different chips. This PUF data can be used to differentiate between two and more chips. This process is similar to a biometric identification scheme.

2.5.2. Key generation and storage

Using PUF's for key generation and storage is much more secure than storing the keys in the memory because the keys in the memory are vulnerable to various physical memory attacks. By using PUF's to generate the keys, The attacker can still use an invasive attack on the chip in order to analyse the structure of the PUF what is being used on the device [13]. However, The attackers would not be able to figure out the keys in the memory. Since, the PUF has to be functional in the first place. Also, In order to use the PUF generated key in a cryptographic process as a cryptographic secret key there is a need for post-processing to make the key reliable, robust and random.

2.5.3. IC Protection

PUF's can be handy in the case of IC Protection. For example, They can be combined with a finite state machine for active IC metering schemes [11]. The traditional ways of securing IC metering is expensive but with the use of PUF's the cost can be significantly reduced.

2.5.4. Protocols which uses CRP's

As mentioned in Sect.3.2.2, In an arbiter PUF there are CRP's which can be used for implementation of several security protocols [19]. For example, In a device authentication scheme the user will have a pair of challenge and response. When the user wants to authenticate the device, The user sends a challenge to the device and the will return with a response. If the response is correct then the system will be authenticated else not. This pair can also be used for encryption. For example, The data can be encrypted using PUF as the secret key and whoever possess the public key can decrypt the message.

2.5.5. Timed Authentication

This technique takes advantage of the time taken by a device to authenticate [20] i.e. A genuine device response time to solve the challenge is much lower when compared with the response time from a model building attack or any kind of software / hardware emulation.

2.5.6. Software licensing

PUF's can also be use in software licensing. By binding them to a particular chip because each chip have a different PUF ID which are unique and can be used for software licensing the information [12].

2.6. Security Schemes for IoT Devices

Any device that is connected to the internet needs to be protected. Since past decade, We used internet through connected things such as PC's, laptops, smartphones and tablets but in this new era where there are devices connected to the internet that are not being used or interacted by the people all the time. For example security cameras, smart air conditioners, etc. are IOT applications. These critical applications if not safeguarded, poses a security risk for an organisation or an individual's when exploited by hackers. Due to various constraints, IOT devices do not possess malware detection and prevention software's which makes it an easy target to attack. The main security concerns for IOT devices are as follows [21]:

- **Authentication** – A process to uniquely identify a device in a network with which the IOT device is interacting.
- **Authorization** – Asks the device what permissions does the user possess. The identity can be anonymous, a user or a device to get access to the resources of a device.
- **Non-Repudiation** – A method to guarantee an action taken on a device is permanent and cannot to rolled back.

- **Confidentiality** – The data shared by the device has to be accessible to authorised parties. Encryption is used to maintain confidentiality.
- **Integrity**–Assures that the confidential data is not tampered or altered. Hashing is a technique used to ensure Integrity.
- **Availability** – A user must be able to access the IOT device objects at any time form anywhere around the world. The use of cloud based solutions ensures availability.
- **Privacy** – Ensures the data is not accessible to public, untrusted third parties and malicious users.

With the usage of PUF's in IOT devices, Some of the above mention IOT security concerns can be resolved. Currently, Authenticating an IOT device using energy efficient solutions is a major research area in IOT. The usage of PUF's paves way to new methodologies and schemes for device authentication. In the following section we focus to discuss various authentication schemes used in current world.

2.6.1. Authentication Protocols

In the field of computer security, Authentication is a crucial process of identifying an object uniquely. Authentication asks an entity to make a provable claim about the presented identity. The claim can often be an email address or a username i.e. when a person logs in to a website, the user claims the email or the username for themselves. Then the system responds to submit a proof for that claim which is often known as a password. If the proof is correct then the device or a server validates the claim and allows users to access its resources. Also, The proof can include 2 factor and 3 factor authentication to increase assurance. Devices such as hardware tokens, biometric reader, One Time Pad (OTP), Google authenticator, etc techniques for multi factor authentication of an object. However, Authentication is also possible for a system to system and not always implies a requirement of a person. An entity such as a person or a system can have different sets of identities for authentication.

In the present world, there are various authentication protocols which promise a high degree of security of applications. Some of the well-used framework for authentication process are as follows:

- **SAML/Shibboleth**

SAML stands for “Security Assertion Markup Language” and was created in 2001 form combining various intellectual property of several corporate business technologies. It is a high level protocol which is designed for exchanging security assertions. SAML is based on

XML which was extensively used back in 2001. However, In recent times XML poses various complications while implementing schemas and parsers. Shibboleth is a java based authentication implementation for SAML. Finally, SAML is a tightly bound, strongly checked protocol which in theory provides a strong security but complicated for implementation. SAML is an open standard protocol which is often used to provide single sign-on to web based applications. The protocol can be used for both authentication and authorization. This protocol constituents of three entities i.e. User Agent, Service Provider (SP) and Identity Provider (IDP) as shown in Figure 2.10.

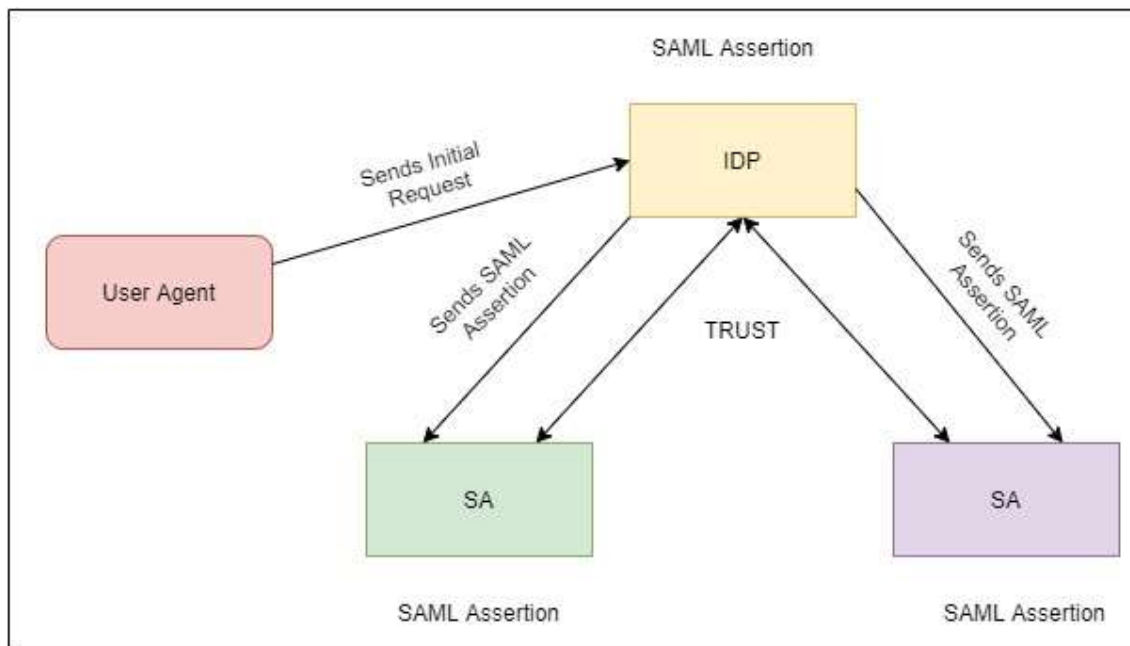


Figure. 2.10: SAML Model [22]

When configuring SAML federation, A trust based relationship is established between the SP and IDP. A user who wants to access a service provider must first authenticate into the IDP. If the user manages to successfully authenticate and authorized then the IDP generates a SAML assertion. The assertion is sent to the application and since the application trusts the IDP, The user is allowed access. Since the user is already authenticated into the IDP, the user can single sign-on to other applications. Also, SAML uses digital signatures to ensure integrity of the certificates.

- **OpenID Connect + OAuth2**

OpenID Connect (OIDC) is an open source authentication protocol and it is designed especially for authentication. Created in 2014 with a decentralisation as a core idea but introduced a new challenges in the field of trust and privacy. The original scheme is designed

in a way that the OIDC provider could be used on any services that support OIDC but lacks in the area of providing trust to the OIDC provider. Being decentralized, A discovery protocol was incorporated which make easy for the users to contact the service provider. OIDC is currently adopted by companies like Google, Microsoft and PayPal.

OIDC is implemented on an existing technology known as OAuth2 [23]. OAuth2 is a framework used for authorization protocol designed for API access. OAuth2 allows unattended access without authentication i.e. API's can be used for period of set time after the user authorization. This also uses TLS which is quite popular in current world for simple data transfer. By implementing TLS, It is not mandatory to use digital signatures thus reducing computational power of the server.

By leveraging the existing process that are part of OAuth2, It is easy to integrate OIDC with the current applications. The combination of these two technologies can solve various authentication and authorization problems. Also, OIDC has various flows for the user to interact with OIDC service providers.

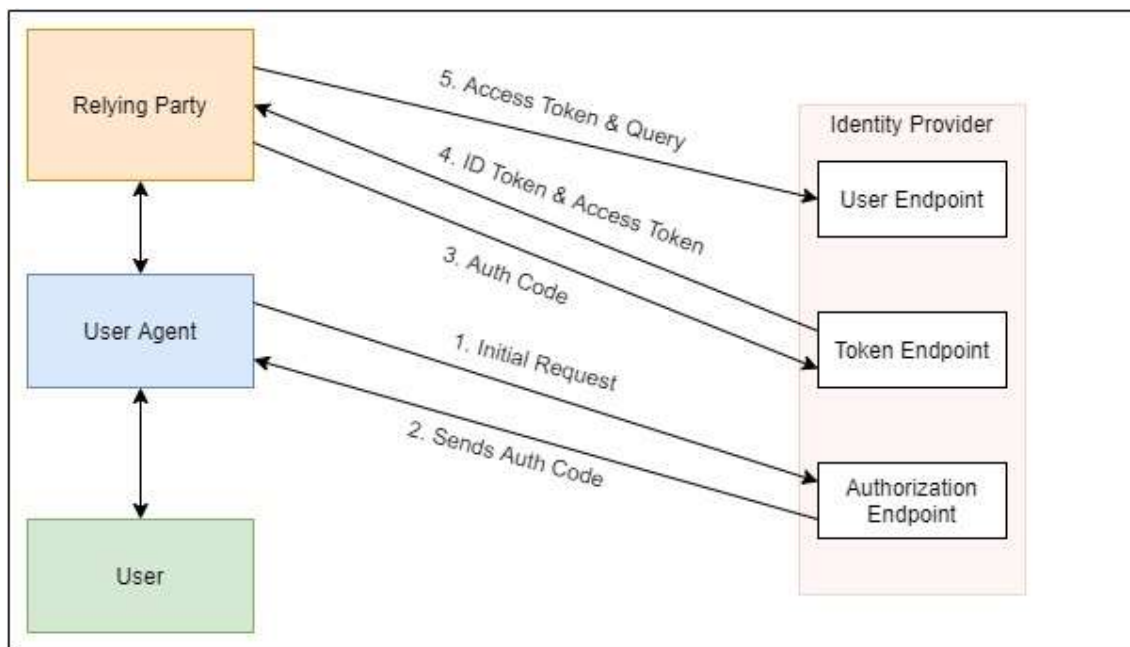


Figure. 2.11: OpenID + OAuth2 Model [23]

Firstly, The user connects to the application using a user agent such a web browser. The user who wants to authenticate to the application sends request to the authorization endpoint of the identity provider. The authorization endpoint sends an authorization code back to the user agent. The Relying Party (RP) then makes a request for a token with the identity provider's token endpoint. The identity provider authenticates the client using the

client ID and secret provided in the previous step. After a successful validation, The identity provider sends back ID Token and Access Token back to the RP which is later validated by the user thus proving the identity of the user. Finally, The access token is used to request resources and this token also works as a single sign-on. A pictographic representation is shown in Figure 2.11.

- **RADIUS**

RADIUS stands for Remote Authentication Dial-In User Service. It uses a fundamental security framework known as AAA framework. It stands for Authentication, Authorization and Accounting. AAA framework helps to control a user's access to a network, determining access levels, policies, user identity and keeping track of the user activities [24]. The data obtained in this framework can be used for monitoring, statistics, Identity and various other administrative tasks. RADIUS is a client-server protocol system that enables a Network Access Server (NAS) to communicate with a central server for various tasks like authenticating users and authorizing access to the network also keeping track of their activities. In RADIUS, NAS acts like a connecting point between a user and AAA servers. RADIUS allows an entity to maintain user profiles in a central database that all remote servers can share.

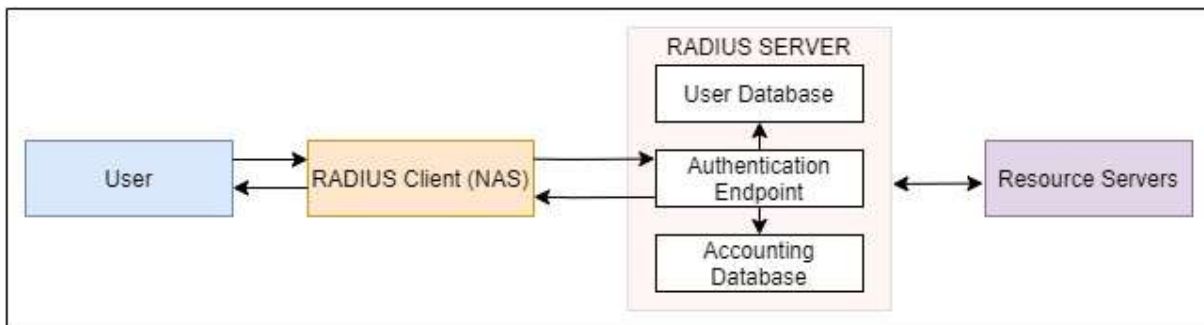


Figure. 2.12: RADIUS Authentication Model [24]

As shown in the Figure 2.12, Firstly a user who want to access the resources of a remote server. In RADIUS, The users device is known as supplicant and the Router is known as a RADIUS Client which acts as a middle man between the user and AAA servers. The user initially sends username and password to the RADIUS Client where encryption is taking place only of password. Secondly, The RADIUS Client send the username to the RADIUS server's database to check whether the user is valid or not. If the user exists, The server checks against the credentials. After validating the credentials, The RADIUS Server sends the acknowledgement containing access permissions, session token and other security profile back to the RADIUS Client also adding an entry to the accounting database in the

RADIUS Server. This keeps track of the activities during the session. Finally, The RADIUS Client send the packet to the user device which allows the device to access network resources until the user logs out or when session is expired.

- **SQRL**

Secure Quick Reliable Login (SQRL) is an alternative for a traditional authentication process. Invented in 2015 by researcher Steve Gibson and is being developed from past 5 years. SQRL offers an authenticated identity based system for user to securely login to the services without the need for any secret object being stored in the servers [25]. Traditional ways are more susceptible to massive breaches occurring globally in the recent times. Thus revealing sensitive credentials to malicious parties. Since with the use of SQRL, In an event of a data breach malicious actors can only access the identities of the users accessing the resources. SQRL allows servers to store the users public keys as identities which is not sensitive. Also, It allows QR codes for easy login to services.

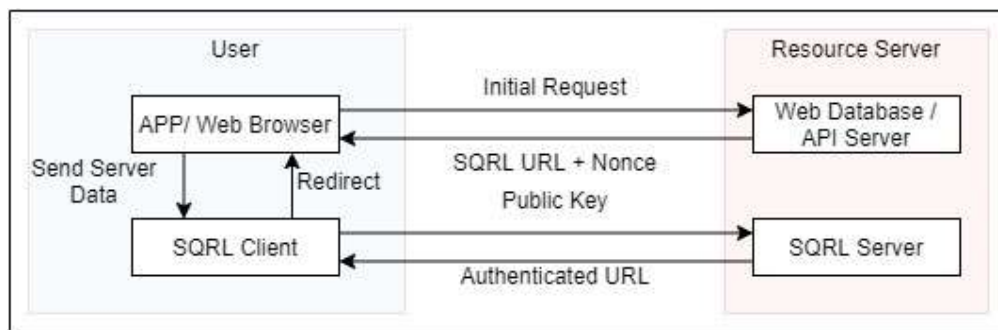


Figure. 2.13: SQRL Authentication Model [25]

As shown in the Figure 2.13, Initially the user requests a website for sign-in. The web server returns the page with an SQRL URL and a Nonce. This data is sent to SQRL Client present with the user. SQRL Client has a private key which is unique. This private key is used to generate identities of the user to access the web server. The SQRL Client sends the user's public key to the web server which is signed by user's private key. Finally, User's identity is authenticated and the website sends the URL to SQRL Client which is later redirected to the users browser with authenticated signed-in session. This architecture has several advantages like [25]:

- User keys never leave the device.
- Only public key is required to verify the user identity.
- User needs to remember just a single master password to access all services.

- Uses site specific credentials which are based on asymmetric cryptography. Thus providing resistance against common attacks such as brute force, password spraying, etc attacks.

There are other various authentication protocols such as Lightweight Directory Access Protocol (LDAP) [26] and Kerberos [26] whose security standards are similar to the above discussed protocols. Thus, No discussion shall happen about these protocols.

2.6.2. Types of IoT Authentication Schemes

IOT Devices are resource constrained and the type of authentication schemes must be chosen very carefully. There are many schemes for authentication of a device which offers various security standards at various levels. Based on resource usage and security, IOT authentication schemes can be classified into six types [21]:

- **Hardware Authentication**– Authentication using physical characteristics of hardware. PUF's is one of an example for hardware authentication.
- **Use of Tokens** - This process uses tokens for user / device authentication. Services such as OpenID Connect and OAuth2 uses tokens in their framework.
- **Authentication Procedures** – Procedures include One-way, Mutual and Three-way authentication steps. In One-way authentication, Only one of the party authenticates and the other is unauthenticated. Scenario where both parties authenticate with each other is known as Mutual Authentication. Three-way authentication includes the use of Central Authority (CA) to authenticate both parties.
- **Architecture based Authentication** – The authentication schemes can be either flat or hierarchical. Which can work on both distributed and centralized architectures.
- **Layer based Authentication** – Allows for the developer to choose at which layer does the authentication process of an application. IOT architecture consists of perception layer, network layer and application layer which are arranged hierarchically. This provides security at various layers of the IOT device.
- **Authentication Factor**–Uses identity or contextual data for authentication. Identity based schemes tend to use cryptographic algorithms for authentication. Whereas contextual data can be physical or behavioural security features like biometric scans, retina scans, keystrokes, access time patterns, etc.

2.7. Usage of PUF's for Authentication

As discussed, PUF's can be used as digital fingerprinting techniques that is used to achieve authentication and identification without using secret keys stored in a device. Each PUF should possess good intra and inter chip variations to be efficiently be useful for cryptographic processes.

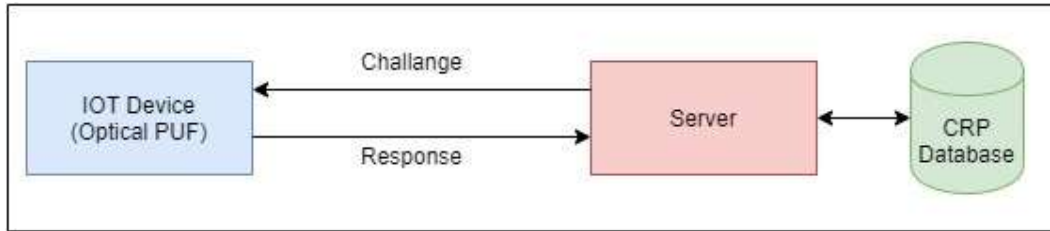


Figure. 2.14: PUF Authentication [15]

The setup mentioned in the Figure 2.14 constitutes a server and an edge device which has a strong PUF i.e. An Optical PUF or Arbiter PUF [15]. At the time of manufacturing of the PUF, The manufacturer would create a database for the challenges and its corresponding responses which is a CRP database of the particular device that would be stored in the server. When the edge device is deployed, the server would require the device to be authenticated. During this phase the server would send a random challenge from the CRP database to the edge device. The edge device would then use the obtained challenge on the PUF which generates the corresponding response which is send to the server for validation. Once the server receives the response from the edge device, It checks and compares the obtained response with the CRP database. The comparison of the responses is measuring the hamming distances between the response present in the CRP against the obtained response. This determines if the response has actually originated from the specific device. Thus, authenticating the edge device successfully.

In order to reduce the resource consumption of the IOT devices, Homomorphic encryption can be used for PUF based authentication process [15] [27] [28]. Initially, The CRP's which are generated for a particular PUF present in the IOT deice is encrypted and stored in an un-trusted environment as shown in Figure 2.15. The property of homomorphic encryption allows validation of the particular response on an encrypted data. This means that the server could run a program in the un-trusted cloud environment that works on the manufacturers encrypted CRP database allowing to obtain whether the response is valid or not without decrypting form the CRP database.

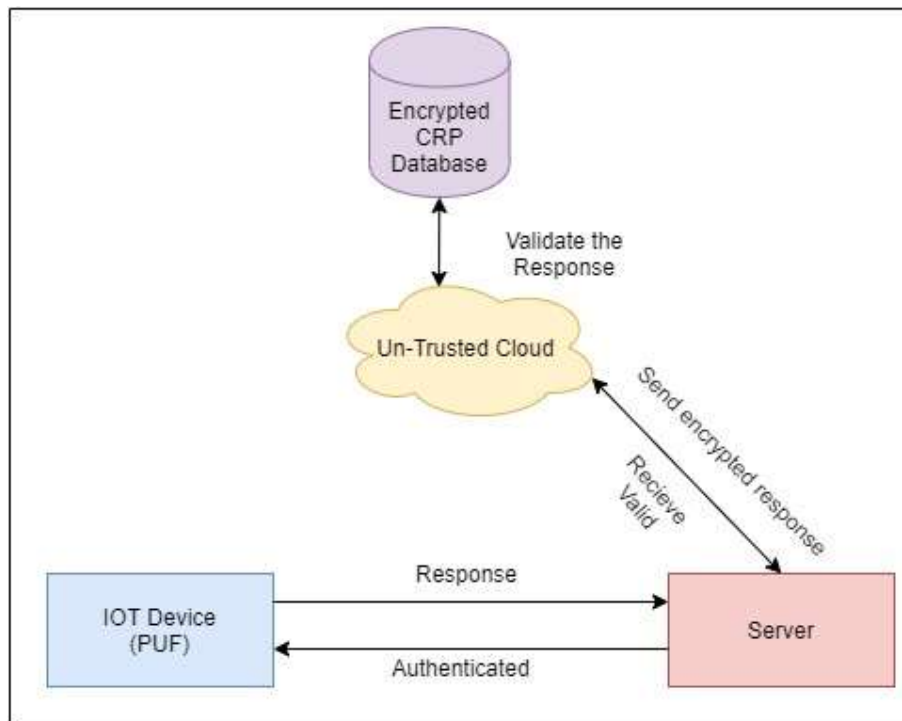


Figure. 2.15: PUF Authentication using homomorphic encryption [28]

Several researchers have used PUF's for cryptographic operations. The Table 2.1 showcases some of the implementations and their drawbacks.

Author's	PUF's usage	Limitations
Braeken, A. [31]	Key Agreement, Authentication	Too many cryptographic operations.
Yilmanz, Y., Gunn, R., Halak, B. [32]	Authentication	Almost 40% of RAM of the IoT device is consumed.
Wallrabenstein, R. [33]	Enrolment and Authentication	Limited to 64-bit operations.
Chatterjee, U., Subhra, R., Mukhopadhyay, D. [34]	Authentication and Key Sharing	High cost of cryptographic operations.

Table 2.1: Usage of PUF's in different schemes

These implementations gives an idea on the ways that a PUF can be used. However, The IoT devices run on limited resources. Care must be taken at the designing phase of the project by focusing on optimising the computational cost (Encryption & Hashing) and network cost (Network Interactions) such that the designed project will be available for various types of devices.

2.7.1 Challenges faced by PUF based authentication

Regardless of being a strong or a weak PUF's, there are various techniques used by malicious actors to hinder the security of the PUF eco-system. Some of the common ways of exploitation of PUF's are as follows:

- **Masquerading the response's**

Considering a scenario where an attacker is trying to masquerade the edge device [29] [30]. When a server sends the challenge, the properties of the PUF would make it difficult to predict the response and the generated rouge response would be quite different that the original response that was expected by the server. After validation the server would reject or drop the device.

- **Man In The Middle Scenario**

PUF based authentication is susceptible to Man-in-the-Middle (MITM) attacks. Since due to resources constraints in the IOT devices, The communication for the authentication is sent in clear text and therefore anyone could view the CRP. If the server sends the same challenge to the device, the attacker would be actually respond to that corresponding challenge without actually forwarding the challenge to the device. To prevent the MITM attack, The CRP's should not be used more than once i.e. the challenge should be either marked as used or deleted from the CRP database. Therefore enough CRP's must be present in the database that can sustain the device through its lifetime which ends up having a very large CRP tables.

- **CRP Database Theft**

Each CRP table is device specific. Therefore if multiple devices are being managed to the server there would be multiple CRP databases that are required to be stored. In a scenario where CRP tables are stolen or privacy of the server gets breached then the entire security of the PUF's corresponding to the edge devices would be lost.

To solve this issue, Researchers came up with solution known as "Secret Model of PUF" [15]. At the time of manufacture instead of making the CRP database, the manufacturer would study the properties of the PUF and would create a model for that particular PUF. For example, the server builds a database of gate delays of each component in the PUF. At the time of deployment the server would pick a random challenge and constructs its expected response from the secret model. Later, Queries the device and validates the response. However, this works for the PUF's which can be modelled such as an Arbiter PUF and Ring

Oscillator PUF. Also, this model still requires secure bootstrapping and secure storage to store the model.

These are some of the challenges that are faced by the PUF's in current world. However, Miniaturization of technology made the possibility to use PUF's for several security based applications mainly in the field of IOT security where often security is neglected.

2.8. Quick Summary

As from above discussion, it can be noted that the applications on the use of PUF's on small scale devices are limitless. PUF's can significantly reduce the cost of processing and communications in an IoT device drastically. also, ensuring an unparalleled identity and authentication of the device. However, the security of an application or the device depends upon the purpose of the applications and type of PUF's that is being considered. The authentication protocols discussed above uses a traditional means of proving its identity. This leaves a window of opportunities to exploit the current authentication protocols. The use of PUF's as a substitute for identity and authentication protocols proved to be resource efficient for IOT Devices.

Chapter 3 : Design and Implementation

3.1. Introduction

The Usage of PUF's in the IOT devices for various cryptographic process are efficient due its unique characteristics elaborated in the review literature. For the design of this project, A strong optical PUF which was designed by Quantum Base organisation is taken into the account for the designing of the framework. This chapter discusses about the proposed design and implementation of the framework.

3.2. About PUF designed by Quantum Base

The PUF that is designed by Quantum Base organization is a strong PUF which uses resonant tunnelling diode (RTD) [35]. RTD's acts like a unique fingerprint which are naturally occurring atom scale variations formed during the manufacturing process. This can result in producing a high number of CRP values. Thus the properties of the Quantum Base designed PUF makes it highly useful in the field of resource constraint devices such as IOT and embedded systems. Which aims to provide secure authenticated data transfer between the Quantum Base's Server's and the IOT Device which holds the Quantum Identity (QID)

3.3. Project Design

This project is designed by keeping modularity as one of the core features. The user has a complete freedom to choose the security standards. The framework allows the use of the unclonable QID for authenticated cryptographic operations. Thus linking the digital identities with the QID's.

3.3.1. High level Overview

The main focus of the project design is to utilize the unique properties of QID for various cryptographical processes in order to securely fetch the data present in the QID Device in order to serve the consumer. To achieve these objectives, the project is divided between four entities which possess multiple functionalities.

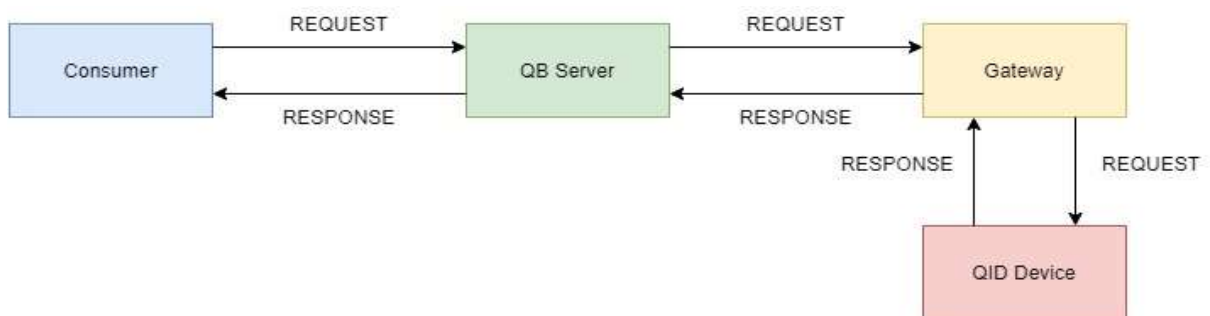


Figure. 3.1: High Level Design Diagram

The Figure 3.1 describes the various entities present in the project. Each of these entities have a unique security functionalities that aid with the data processing.

- **Consumer** – The Consumer is an entity which initiates the communication. After establishing a secure communication with the QB Server, Consumer requests the data that is to be fetched by the QB Server. The functionalities of the Consumer application are as follows.
 1. Establish secure communication with the QB Server
 2. Send Data to the QB Server
 3. Verify the response got back from the QB Server.
- **QB Server** – QB Server is considered to be an important node in the entire framework. It handles various tasks from multiple nodes to efficiently process the data. Also, It contains the CRP database's which is crucial for cryptographic operations. The following are the functionalities for the QB Server application.
 1. Verify incoming data packet
 2. Filter the incoming data
 3. Establishing security association (SA) in the SA database
 4. Apply Cryptographic operations
 5. Prepare the outgoing data packet
 6. Logging and Accounting
- **Gateway** – In this application, The gateway functions as a protocol translator between various modes of communication protocol and does not use any of the properties of the QID. Thus, Gateway is not considered during the architecture and the implementation phase of the framework.
- **QID Device** – QID Device contains the PUF designed by Quantum Base and has the data which was requested by the Consumer. The functionalities of the QID Device are as follows:
 1. Verify incoming data packet
 2. Filter the incoming data
 3. Check the CRP
 4. Fetch the data
 5. Apply cryptographic operations
 6. Prepare the outgoing data packet

The above mentioned functionalities are an overview of the structure of the framework. Detailed description of the entities will be provided in the upcoming chapters.

3.3.2. Overview of Message Sequence

Message sequence is broken down in to 5 stages for better understanding of the design. This sections aims to provide details if the various operations involved in the message flow. A complete diagram of the process sequence is represented in Figure 3.2. The functionalities of each stage is detailed in the upcoming sections. The stages are as follows:

- Stage – 1 (Consumer → QB Server)
- Stage – 2 (QB Server → QID Device)
- Stage – 3 (QID Device → QB Server)
- Stage – 4 (QB Server → Consumer)
- Stage – 5 (Consumer → QB Server → Consumer)

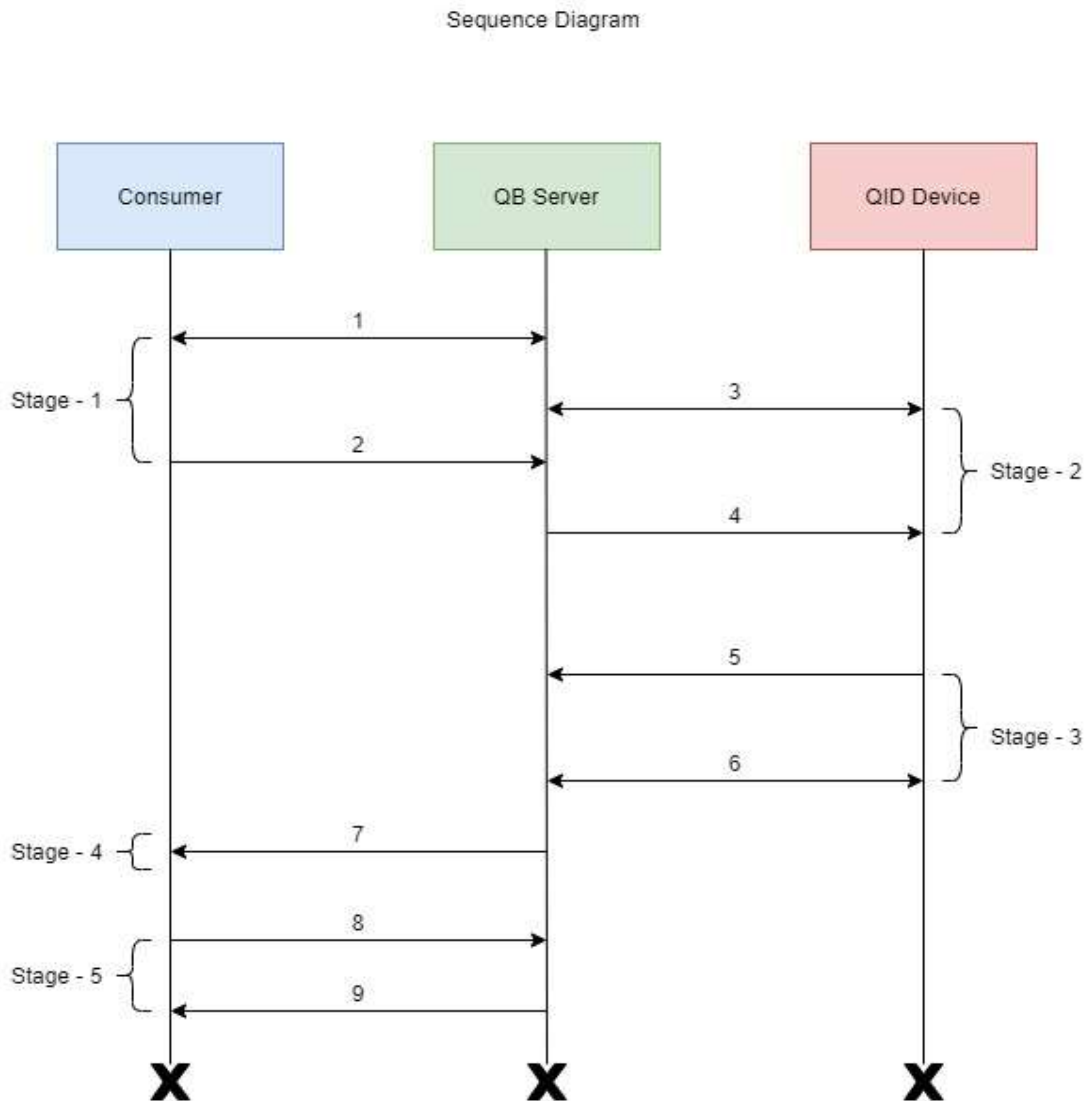


Figure. 3.2: Sequence Diagram

- **Stage – 1**

This stage involves communication of data between Consumer and the QB Server. The Figure 3.3 shows and the description showcases the type of interactions with the parties.

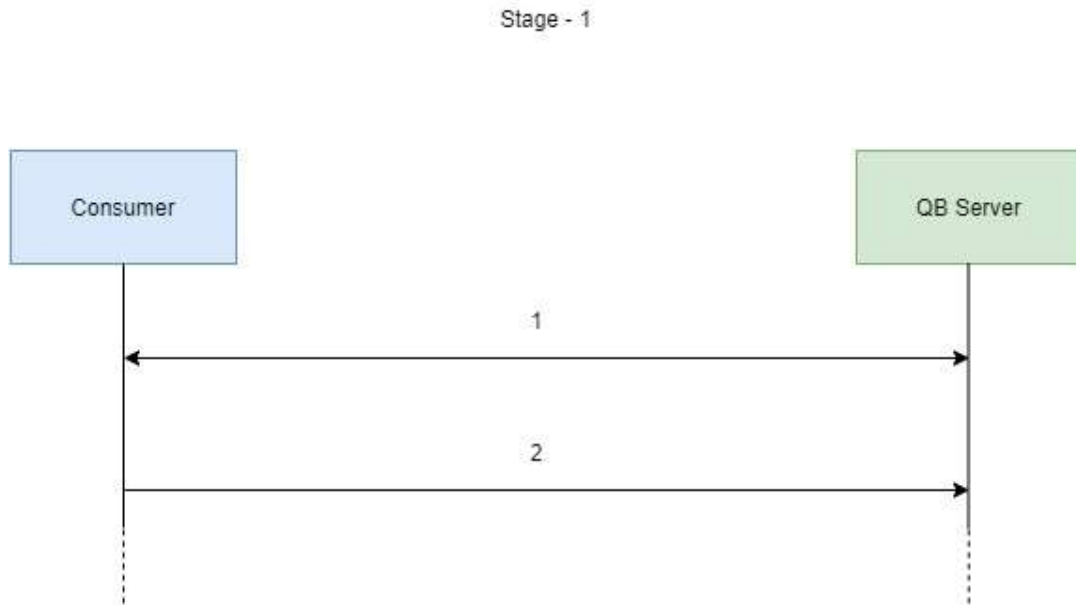


Figure. 3.3: Consumer to QB Server

- 1.) Establishing a Secure Communication
- 2.) Data Request which involves the user to send QID Device ID, Encryption Algorithm and Signature Algorithm the person wish to use.

- **Stage – 2**

After receiving the initial data request from the consumer, The QB server creates a security association which is needed for drawing out a relationship between a the Consumer and QID Device. The QB server checks all the required fields and prepare the outgoing data packet to QID Device.

Stage - 2



Figure. 3.4: QB Server to QID Device

3.) Establishing a secure connection between QB Server and QID Device

4.) Send the data request to the QID Device.

- **Stage – 3**

The QID Device process the server request and uses the QID properties for cryptographic process. After successful mapping the challenge which is provided by the server to the CRP database present in the QID Device, The data is fetched and necessary cryptographic process which are requested by the consumer are applied and send to the QB Server. After a successful transmission, The connection between the QB Server and QID Device is closed.

Stage - 3

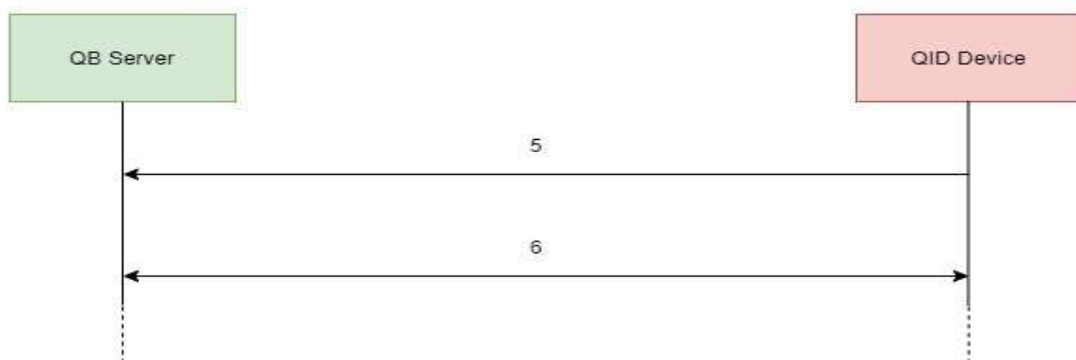


Figure. 3.5: QID Device to QB Server

- 5.) Send the response data back to the QB Server.
- 6.) Close the connection between QID Device and QB Server.

- **Stage – 4**

After receiving the data from the QID Device, QB Server reverses the cryptographic operations to verify the authenticity. After successful verification, The QB Server signs and transmits the requested data to the consumer device.

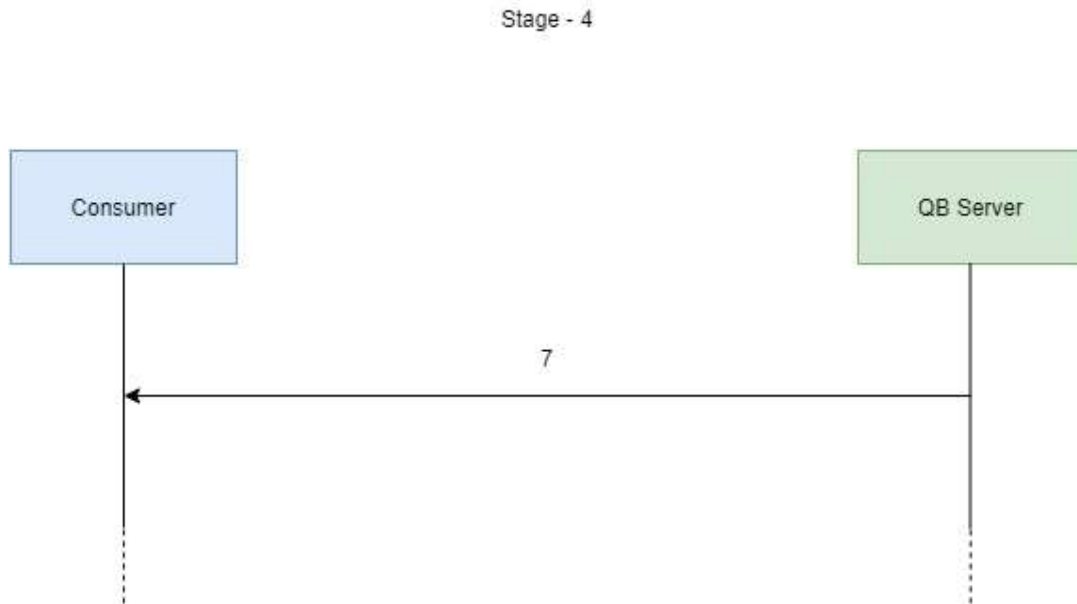


Figure. 3.6: QB Server to Consumer

- 7.) Send requested data to Consumer

- **Stage – 5**

A hash is generated of the data received by the consumer which is sent to the QB Server for integrity check. Upon receiving the hash, QB Server checks the hash values and respond with ACK or NACK. Once a response is received the data is either displayed the consumer else data is requested again and the cycle continues.

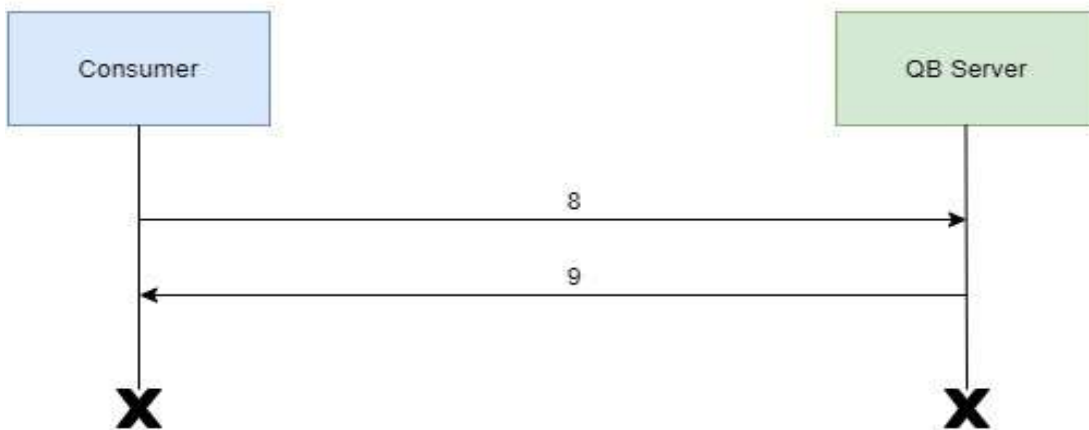


Figure. 3.7: Consumer to QB Server and vice versa

- 8.) Send the hash value of received data to QB Server.
- 9.) Send ACK / NACK to Consumer.

3.3.3. Project Architecture

The Figure 3.8 represents a detailed description of the frameworks architecture. This section focuses to provide in depth details about each and every operation taking place in the framework. A staged approach is incorporated while designing. This allows modularity for future improvements.

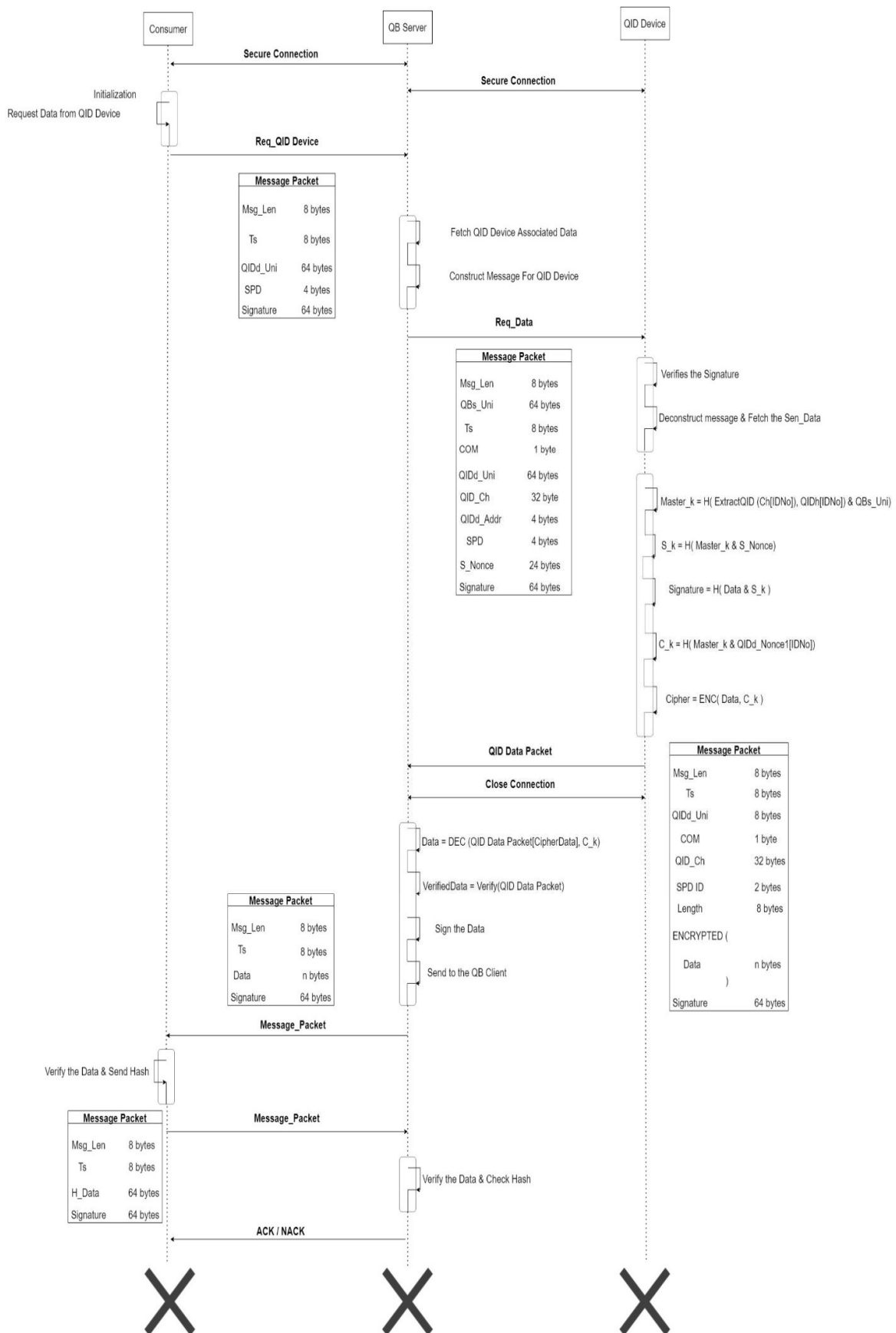


Figure. 3.8: Project Architecture

For better understanding of the architecture, the process is briefed in five stages.

- **Stage – 1**

This stage is known as initialization phase .After establishing a secure connection, The consumer is prompted to provide QID Device unique ID along with the security options. The security options include the type of encryption algorithm for the data encryption and type of signature algorithm for signing the data.

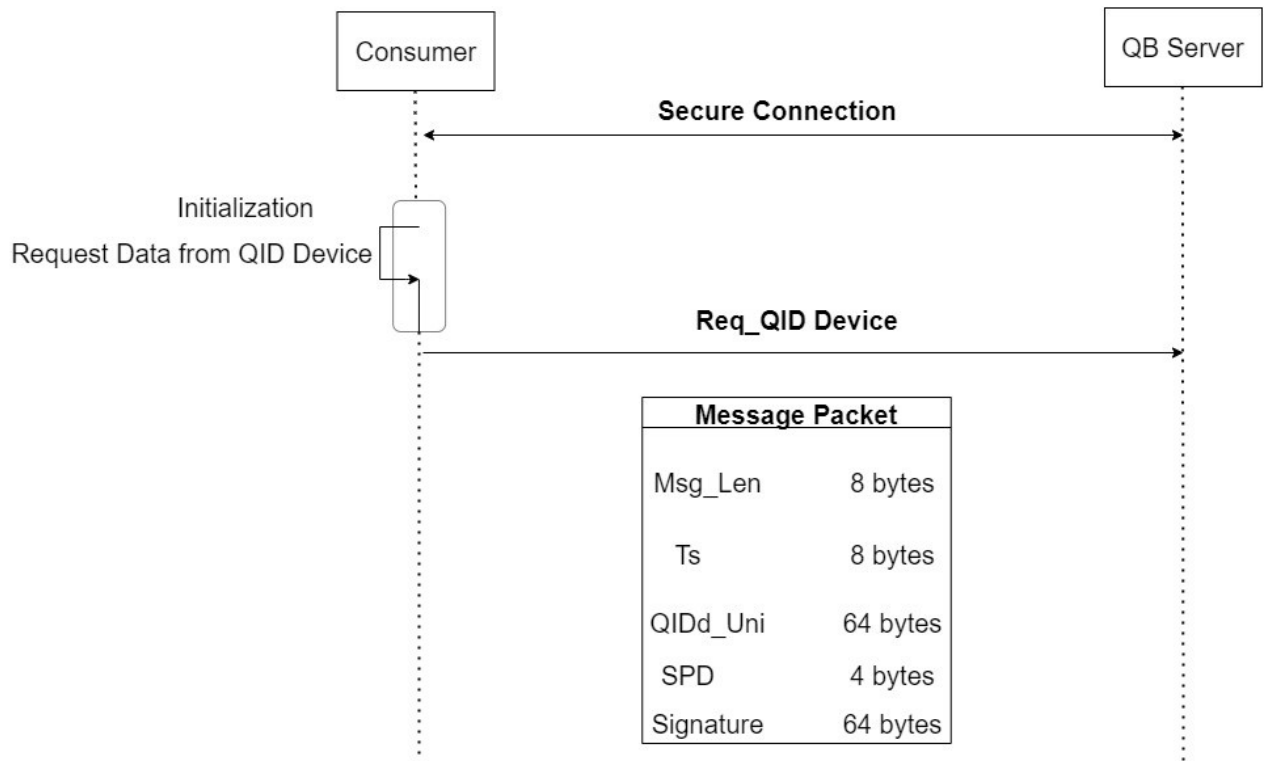


Figure. 3.9: Stage – 1 Data Process

The consumers request is then send to QB Server for further processing. The following data fields are present in the request data packet as shown in the Figure 3.9:

1. **Msg_Len** – The total length of the packet without the signature is considered. The size of this field is 8 bytes. This field ensures the check for data loss of the packet.
2. **Ts** – Ts denotes the timestamp of the consumer device. This field is 8 bytes in size and it is attached at the time of packet creation for transmission.
3. **QIDd_Uni** – This is an mandatory field required by the QB Server. QID Device name or Unique ID is taken from the consumer which is later mapped by the QB Server. This field takes up to 64 bytes.

4. **SPD** – This field is known as Security Profile Data. It carries options which are used for cryptographical process at both QID Device and QB Server. The information about the data encryptions and signing is carried out by this field. It requires 4 bytes.
5. **Signature** – Once the required data is prepared, A combined signature is generated for the fields of Ts, QIDd_Uni and SPD using ECDSA signature algorithm. Cryptographic process will be discussed in Section: 4.1.2.1.

- **Stage – 2**

After the data received by the consumer, QB Server verifies the data packet and maps the QIDd_Uni provided by the consumer with the entry of QID Unique ID present on the Security Association Database (SAD) in the QB Server. Details regarding SAD will be provided in Section: 4.3.2.1. The QB Server randomly picks a challenge associated with the QID Device unique identity from the corresponding CRP database and stores the SPD value for further use in the SAD database. Also, The QB Server will randomly generate a nonce which is will be sent to QID Device along with other data entries.

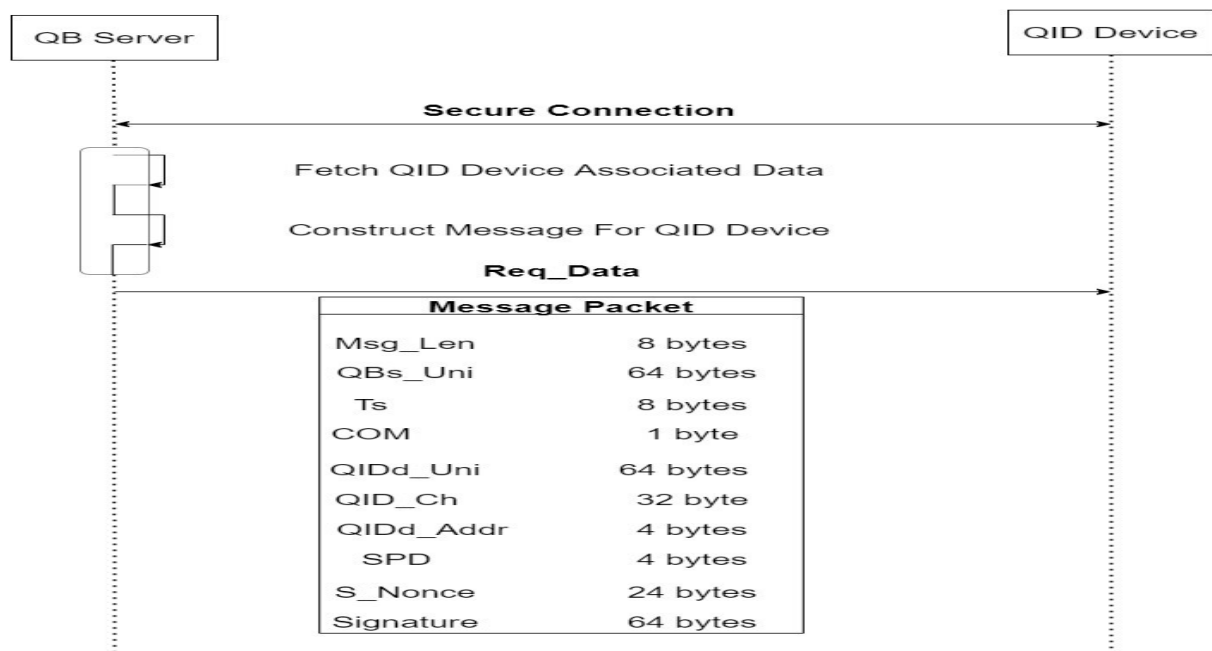


Figure. 3.10: Stage – 2 QB Server Mapping and Transmitting.

The following data fields are present in the request data packet to QID Device as shown in the Figure 3.9:

1. **Msg_Len** – The total length of the packet without the signature is considered. The size of this field is 8 bytes. This field ensures the check for data loss of the packet.
2. **QBs_Uni** – Bears the value of QB Server unique ID which will be later used by QID Device. This field is 64 bytes.
3. **Ts**–Ts denotes the timestamp of the QB Server. This field is 8 bytes in size and it is attached at the time of packet creation for transmission.
4. **COM** – This field represent the type of communication protocol being used by the QB server for communication. This field will be used by the gateway for translation between different protocols and requires 1 byte.
5. **QIDd_Uni** – This field bears the value of QID Device unique ID which is present in the database of the QB Server. This field is 64 bytes in size.
6. **QID_Ch** – QB Server randomly selects a challenge from its CRP database. This challenge transmitted to QID Device whose response acts as authentication. The size of this field is 32 bytes.
7. **QIDd_Addr** – This field holds the address of QID Device. Helps the QB Server to locate the QID Device. The size of this field is 4 bytes.
8. **SPD** – This field holds the security profile values. During Stage – 1, The acquired security options are verified by the QB Server and transmitted to the QID Device. This field is 4 bytes in size.
9. **S_Nonce** – This field is a randomly generated 24 bytes value that is used to generate a signature key in the QID Device.
10. **Signature** – ECDSA is used as a signature algorithm. The public key, algorithm used and packet signature is transmitted along with the data. The size of this field is 64 bytes.

- **Stage – 3**

This stage is divided into 3 phases because of the multiple operations involved. The following phases are as follows :

- i. Data Fetching Phase
- ii. Cryptographic Operations Phase
- iii. Transmission Phase

i. Data Fetching Phase

Once the request packet is received from the QB Server, The QID Device verifies the packet with ECDSA verify operation. If the packet verification is failed, The packet is discarded. Upon successful verification, The packet is deconstructed and mapping of packet content takes place with the data present in QID Device i.e. Checking QIDd_Uni, QID_Ch, S_Nonce, SPD and QBS_Uni.

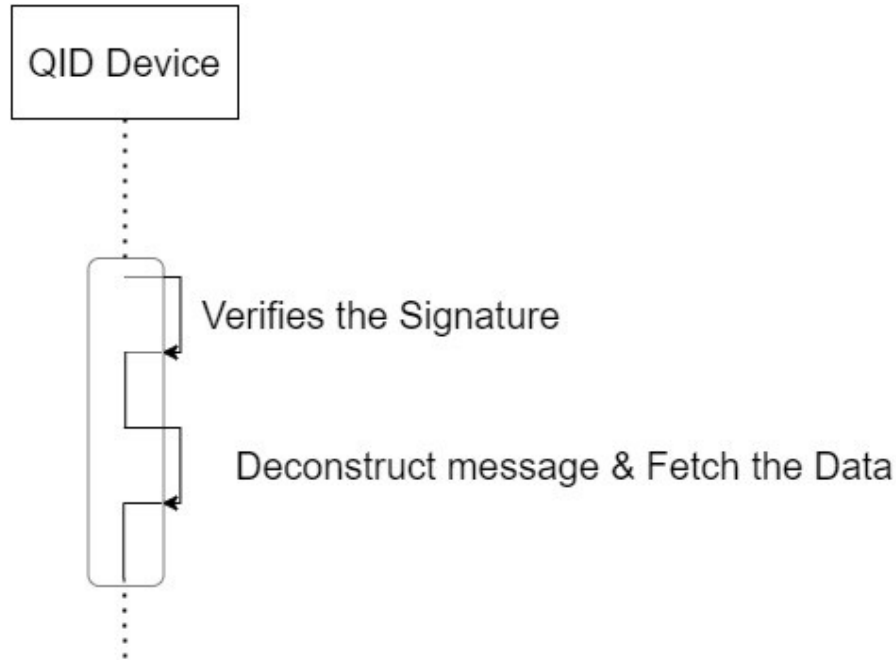


Figure. 3.11: Data Fetching.

Finally, The requested data is fetched and proceeded to the next phase.

ii. Cryptographic Operations Phase

This phase involves in generating key material for data signing and encryption i.e. Generation of Master Key, Signature Key and Cipher Key. These generated keys will be used for data signing and encryption process as shown in the Figure 3.12.

- **Generation of Master Key** – As shown in the Figure 12, Master key (Master_k) is generated by hashing the Response and QB Server unique ID (QBs_Uni). The response is obtained by passing the acquired Challenge (QID_Ch) by the QB Server and comparing it with the CRP Database present in the QID Device with the help of ID No and helper data of the PUF. Then the values of response and QBs_Uni are hashed with a SHA-256 algorithm which is known as Master Key
- **Generation of Signature Key** –Based upon the SPD data, If the user chooses to apply a signature to the data then the Signature Key (S_k) is generated by hashing

Master_k and S_Nonce. The algorithm used for generating a signature key is SHA-256 or the algorithm of the consumers choice.

- **Signing the data** – After generating the S_k, The data and S_k is combined and hashed using SHA-256 algorithm or the algorithm of the consumers choice.
- **Generation of Cipher Key** –The generation of Cipher Key (C_k) acts like a secret key for encryption of data. Master_k and a pre-present Nonce (QIDd_Nonce1) is combined to generate a hash value which will be used as a C_k. Note that QIDd_Nonce1 is present in both QB Server and QID Device CRP databases and it is equal.
- **Encrypting the data** – The data is encrypted using an AES-256-GCM encryption algorithm or the algorithm of the consumers choice. Here, C_k is used as a key for generating the Cipher.

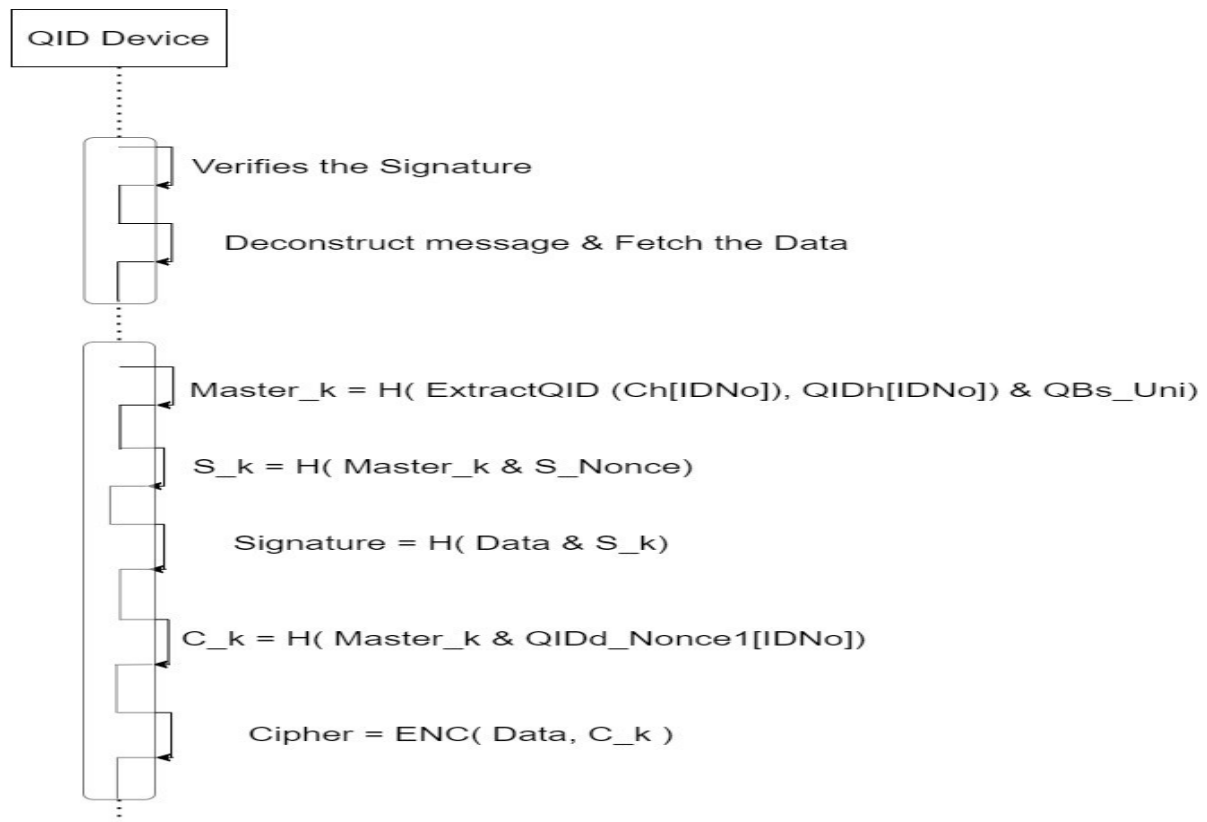


Figure. 3.12: Cryptographic Operations.

iii. Transmission Phase

After successfully generating the required material, The packet is constructed and the data is transmitted to the QB Server. After a successful transmission the connection is closed. The figure below shows the combined operations in this stage.

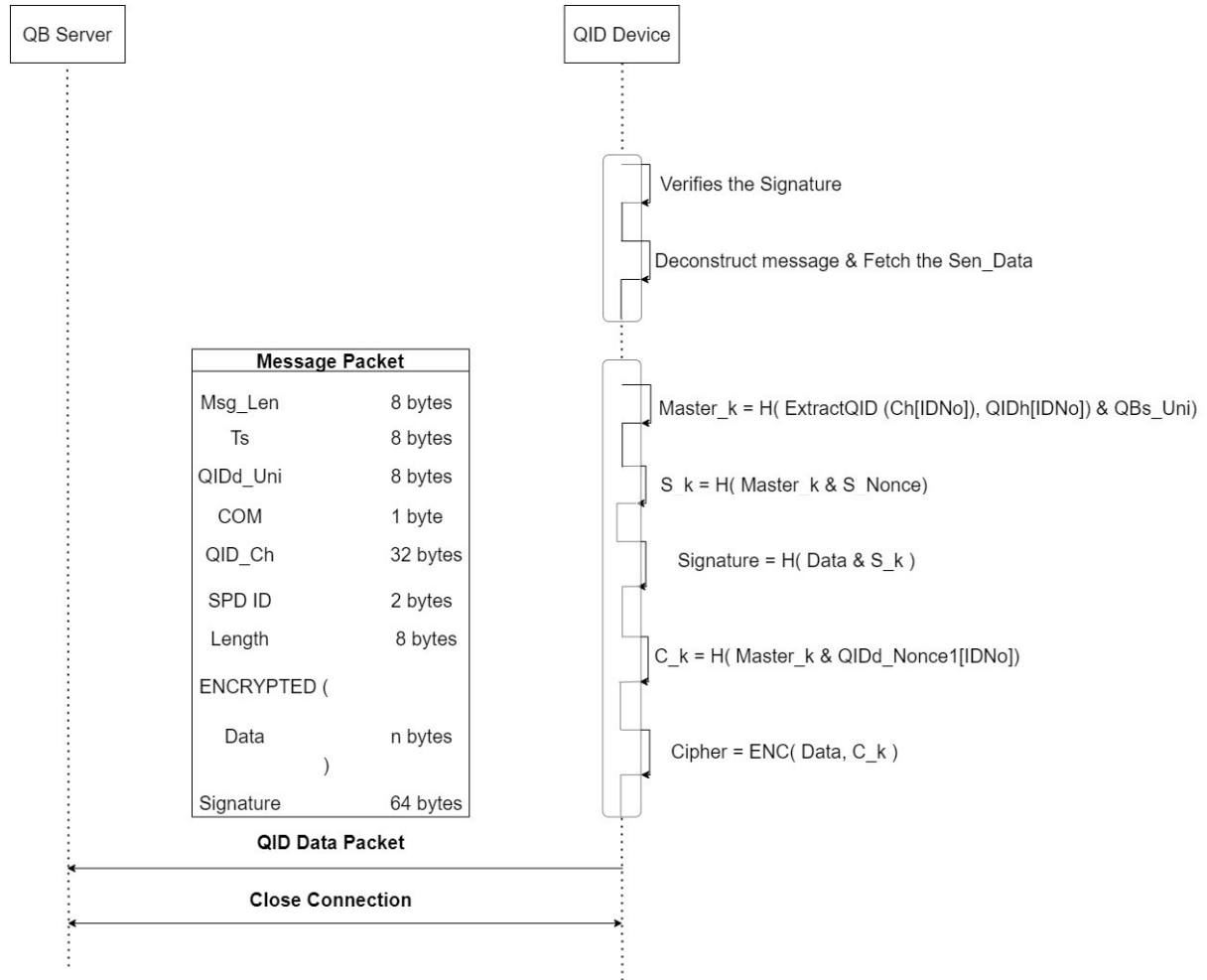


Figure. 3.13: Stage – 3 QID Device Operations.

The response data packet contains the following data fields as shown in the Figure 3.13:

1. **Msg_Len** – The total length of the packet without the signature is considered. The size of this field is 8 bytes. This field ensures the check for data loss of the packet.
2. **Ts** – Ts denotes the timestamp of the QID Device. This field is 8 bytes in size and it is attached at the time of packet creation for transmission.
3. **QIDd_Uni** – This field bears the value of QID Device unique ID. This field is 64 bytes in size.

4. **COM** – This field represent the type of communication protocol being used by the QID Device for communication. This field will be used by the gateway for translation between different protocols and requires 1 byte.
5. **QID_Ch** – The challenge that is used by the QID Device is present in this field. The size of this field is 32 bytes.
6. **SPD ID** – This field holds the security profile ID that is used by the QID Device. This contains information related to data signing and encryption. This field is 4 bytes in size.
7. **Length** – This field contains the length of the encrypted data packet. The size of this field depends on the size of the ENCRYPTED Content.
8. **ENCRYPTED Content** – Encrypted data packet.
9. **Signature** – Holds the value of Signature which is generated in the Cryptographic Operations Phase. This field is 64 bytes in size.

- **Stage – 4**

Upon receiving of the data packet from the QID Device, The QB Server checks for the necessary information before applying for cryptographic operations. Message length, Timestamp, QIDd_Uni, SPD ID and the length of the signature is checked. The process of cryptographic operations is a 3 phase process which involves the following:

- i. Recreation Phase
- ii. Decryption and Verification Phase
- iii. Transmission Phase

i. Recreation Phase

In this phase the data packet is segregated and necessary information is collected. Using the QIDd_Uni, SPD ID and QID_Ch data is compared with the SAD. Each and every field should match with the SAD else the packet is discarded and retransmission takes place. Based on the SPD ID, The respective keys recreation process takes place.

- **Master Key** – The data required for this field is the response value from the CRP database and the QB Server unique ID. These two field are present with the QB Server.
- **Signature Key** – The generated Master Key is used along with the server nonce (S_Nonce) to generate the Signature Key. Note that S_Nonce is already stored at the time of data request to the QID Device. If the user requests for 'No Signature' option then this step is skipped.

- **Cipher Key** – The creation of Cipher Key uses Master Key and Nonce (QIDd_Nonce). The QIDd_Nonce comes along with the CRP database which is equal at bot QB Server and QID Device. This key will be later used to decrypt the data.

ii. Decryption and Verification Phase

After successful recreation of the necessary keys, The decryption of the encrypted data packet takes place. The decryption process takes the recreated Master Key and the Cipher Key to retrieve the QID Device data requested by the Consumer.

The decrypted QID data is then verified. Here the recreated Signature Key is used along with the decrypted QID data to generate a signature. If the generated signature matches with the Signature present with the QID response packet then it goes to the next phase else the data is identified as tampered and discarded.

iii. Transmission Phase

The decrypted data is signed by the QB Server using ECDSA signature algorithm and transmitted to the Consumer. The figure below shows the entire operations taking place is Stage – 4.

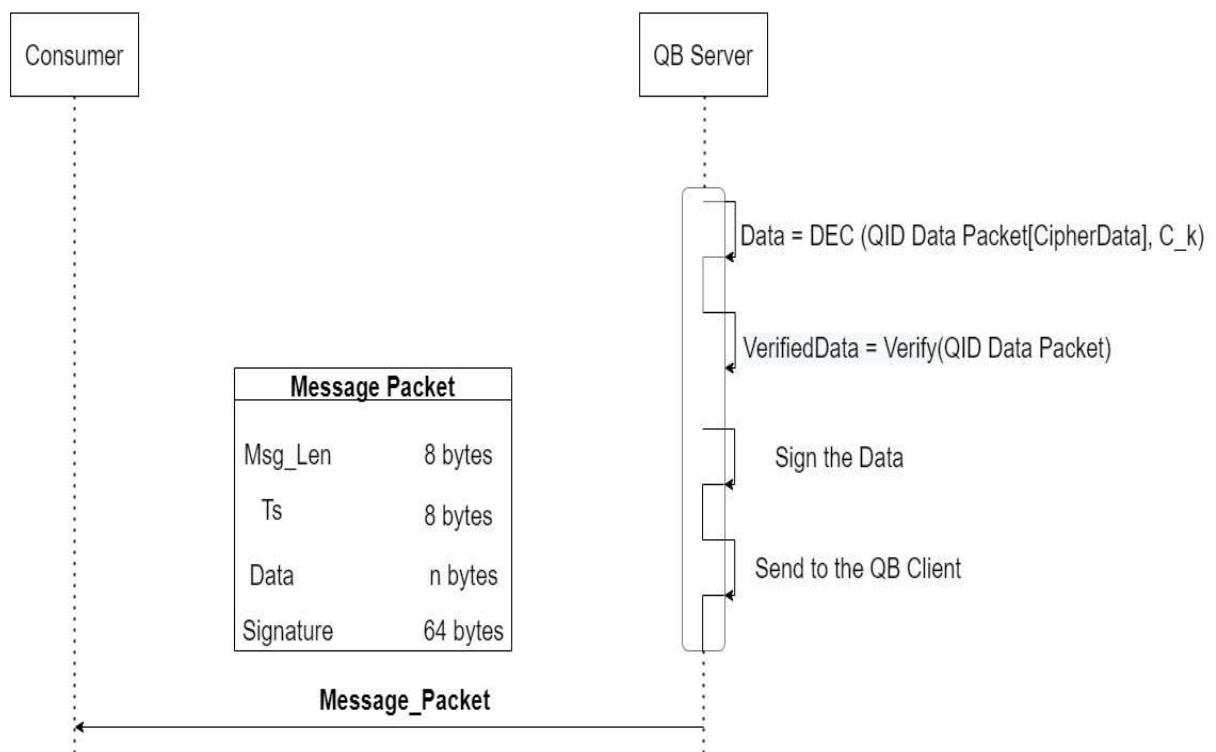


Figure. 3.14: Stage – 4 QB Server Operations.

The response data packet to the Consumer contains the following data fields as shown in the Figure 3.14:

1. **Msg_Len** – The total length of the packet without the signature is considered. The size of this field is 8 bytes. This field ensures the check for data loss of the packet.
2. **Ts** –Ts denotes the timestamp of the QB Server. This field is 8 bytes in size and it is attached at the time of packet creation for transmission.
3. **Data** – The QID Device data which is requested by the consumer.
4. **Signature** – The public key, algorithm used and packet signature is transmitted along with the data. The size of this field is 64 bytes.

• Stage – 5

At this stage, The data is received by the Consumer is verified using ECDSA algorithm. After successful verification, The QID Device Data is hashed using a hashing algorithm i.e. SHA-256 and transmitted back to the server for verify the data integrity. The following field are present in the request packet to QB Server as shown in the Figure 3.15:

1. **Msg_Len** – The total length of the packet without the signature is considered. The size of this field is 8 bytes. This field ensures the check for data loss of the packet.
2. **Ts** – Ts denotes the timestamp of the Consumer. This field is 8 bytes in size and it is attached at the time of packet creation for transmission.
3. **H_Data** – Hash value of the QID Device data. The size of this field is 64 bytes.
4. **Signature** – The public key, algorithm used and packet signature is transmitted along with the data. The size of this field is 64 bytes.



Figure. 3.15: Stage – 5 Final Operations.

Upon receiving the request, The QB Server verifies the response packet and matches the hash value received by the Consumer with the hash value stored in the QB Server. If successful, The QB Server transmits ACK and the Data is displayed to the Consumer else QB Server transmits NACK and the QID Device Data is retransmitted to the Consumer.

3.3.4. About the Data Stores

In this framework, The QB Server and QID Device play a major role. The information present in the device storage that are vital for authentication and signing operations of the PUF. This section focuses about the critical fields present in the storage system of QB Server and QID Device.

- **QB Server Storage**

QB Server is the entity which manages various devices and controls the crypto operations. QB Server has four data stores which are as follows:

1. **General Store** – This stores the generic information like server address, server unique ID , public keys, private key, etc which is mentioned in the below figure.
2. **CRP Database** – Holds the device specific CRP tables. During the initial request the QB Server randomly selects a challenge form this data base and transmits to the QID Device.
3. **SAD Database** – This holds various profile data i.e. The type of encryption algorithm and signing algorithm requested by the consumer. This database also linked with the challenge that is currently being used and timeout period associated with it.
4. **Accounting and Logging** – QB Server logs data for every transaction. This helps to easy the troubleshooting process and to keeps tacks of any failures in the process.

QB SERVER STORAGE						
Parameters			Description			
QBs_Uni			Unique ID for QB Server			
QBs_Add			Server Address			
QBs_Ts			QB Server Timestamp			
QBs_PVK			QB Server Private Key for Signature			
QBs_PUB			QB Server Public Key for Signature			
SAD			Security Association Database			
QIDd_Uni	QIDd_PUB	QID_No	QID_Qb	QID_Ch	QID_Rp	Nonce
0x32974...	an2M8B54jKIOe..	1	0x823784...	0x827482...	0x98213...	0x82738...
.
0x32974...	an2M8B54jKIOe..	N	0x712673...	0x716371...	0x87102...	0x93849...

Figure. 3.16: QB Server Storage.

The Figure 3.16 is a schematic of the QB Server storage. The terms in the storage is described as following:

1. **QID_No** – The ID no associated with the challenge.
2. **QID_Qb** – ID of QID identity in the Quantum Base servers.
3. **QID_Ch** – The unique challenge that is given to the QID Device of authentication.
4. **QID_Rp** – The expected data response received by the QB Server from the QID Device is cross checked with this field. This field is used to recreate the Master Key
5. **Nonce** – This variable is present in the CRP Databases of both QB Server and QID Device with identical values. This is used to recreate cipher key.

- **QID Device Storage**

QID Device is a resource constrained hardware. Thus, The functionality of this device is designed as energy efficient as possible. Functions such as data handling and crypto operations are handled by this device. QID Device has two data stores which are as follows:

1. **General Store** – This stores the generic information like server address, server unique ID, QB Server unique name, public keys, private key and security association ID which is mentioned in the below figure.
2. **CRP Database** – Holds the Challenge-Response pairs and corresponding information. During the initial request the QB Server randomly selects a challenge form this data base and transmits to the QID Device.

QID DEVICE STORAGE					
<u>Parameters</u>			<u>Description</u>		
QIDd_Uni			Unique ID for QID Device		
QBs_Add			Server Address		
QBs_Uni			Unique ID for QB Server		
QIDd_Ts			QID Server Timestamp		
QIDd_PVK			QID Server Private Key for Signature		
QIDd_PUB			QID Server Public Key for Signature		
QBs_PUB			QB Server Public Key for Signature		
SA ID			Security Profile Data Number		
QID_No	QID_Qb	QID_Ch	QID_Rp	QID_h	Nonce
1	0x823784...	0x827482...	0x98213...	0x23782...	0x82738...
.
N	0x712673...	0x716371...	0x87102...	0x34891...	0x93849...

Figure. 3.17: QID Device Storage.

The Figure 3.17 is a schematic of the QID Device storage. The terms in the storage is described as following:

6. **QID_No** – The ID no associated with the challenge.
7. **QID_Qb** – ID of QID identity in the QID Device.
8. **QID_Ch** – The unique challenge that is present in the QID Device. Later, Compared with the challenge received by the QB Server.
9. **QID_Rp** – After successful comparison of the QID_Ch, The corresponding response is used to create the Master Key. This response is used for data encryption and signing.
10. **Nonce** – This variable is present in the CRP Databases of both QB Server and QID Device with identical values. This is used to create cipher key.

3.4. Cryptographic Standards implemented in the Framework

Cryptographic standards plays a crucial role in the energy efficiency of resources constraint devices. However in order to achieve energy efficiency, IOT application designers tend to neglect data security. So to close this gap between energy efficiency and data security, I had identified some of the encryption, hashing and signing algorithms which is perfectly suitable for this project. However since this framework offers modularity, There is a provision to include future energy constraint algorithms. The identified cryptographic standards are as follows:

- **Encryption** – This is one of the key operation that consumes high amount of energy. Cryptographic algorithms such as AES-256-GCM and AES-256-CBC are chosen in this framework because of their better performance and energy efficiency compared with their peers. [36] [37] [38] [44]
- **Hashing** – The designed framework uses hashing operations from integrity check to generating signatures. To aid these operations, hashing algorithms such as SHA-256 and SHA-3 are chosen due to their performance and speed. [39] [40] [41]
- **Signing** –In this framework data signing uses hashing functions for the operations. However, ECDSA algorithm is used for operations that does not involve the use of PUF's. ECDSA algorithm offers better security and performance when compared with RSA. [42][43]

Thus the combination of the PUF based identity along with the identified modern cryptographic standards ensures a balance of data security, speed and performance in resource constraint devices.

3.5. Implementation

The implementation of the above discussed framework is developed in Java Programming language. Java being modular and easy of use helps for fast implementation of the project. This section showcases the working of the protocol and the complex functions that are involved in developing the framework.

3.5.1. System Specifications

The program for Consumer, QB Server and QID Device are developed in a single host system. For this implementation, Raw TCP socket is used as a communication channel between the entities. The following are the software specifications the framework is developed on:

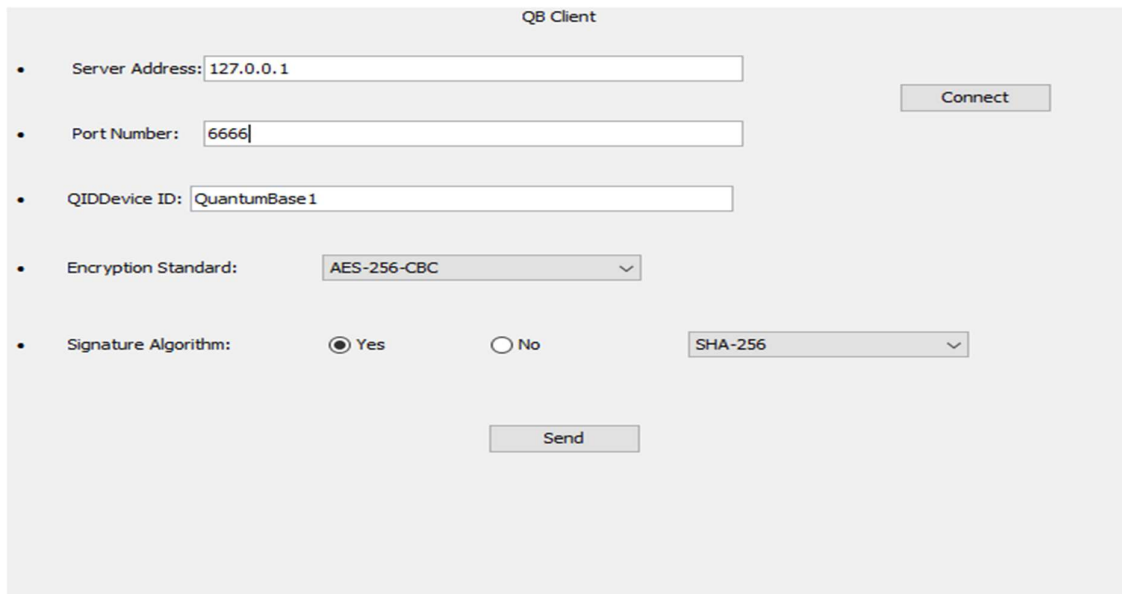
- **Operating System** – Windows 10 64-bit
- **JDK Version** – 1.8.0_251
- **JRE Version** – Build 1.8.0_251-b08
- **IDE Details** – Apache NetBeans IDE 11.3 [45]

3.5.2. Implemented Framework

This section showcases the designed user interfaces and operation of protocol at a practical level. Also gives an idea of the working of the framework from the perspective of a consumer or an administrator.

- **Consumer Application**

Consumer application is an interface which allows connection to the QB Server and choose from various options. The Figure 3.18 showcases the user interface of the consumer application in which it is connected to QB Server which is running on local host.



The screenshot displays the 'QB Client' application window. It contains several input fields and buttons. The 'Server Address' field is set to '127.0.0.1'. The 'Port Number' field is set to '6666'. The 'QIDDevice ID' field is set to 'QuantumBase1'. The 'Encryption Standard' is set to 'AES-256-CBC'. The 'Signature Algorithm' has 'Yes' selected with a radio button, and the 'SHA-256' option is selected in the dropdown menu. There are 'Connect' and 'Send' buttons.

Figure. 3.18: Consumer Interface.

The QID Device data will be displayed to the consumer once the cycle is completed. In this test run the sample data that is present in the QID Device is represented in the Figure 3.19.

```
Welcome to QB Client
Connected to: 127.0.0.1:6666

To Server:QuantumBase1

QuantumBase1:21
Verified
Successfully wrote to a file
BUILD SUCCESSFUL (total time: 25 seconds)
```

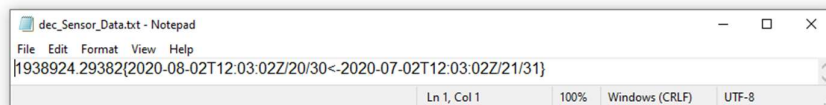


Figure. 3.19: Data Display to Consumer.

- **QB Server Application**

In this implementation QB Server is hosted on localhost and on a custom port. The QB Server is modular and integration of cryptographic standards are made simple. In Figure 3.20, a showcase of one of the test run is detailed. The text which is represented in the figure is generated for each and every operations thus satisfying logging and accounting feature.

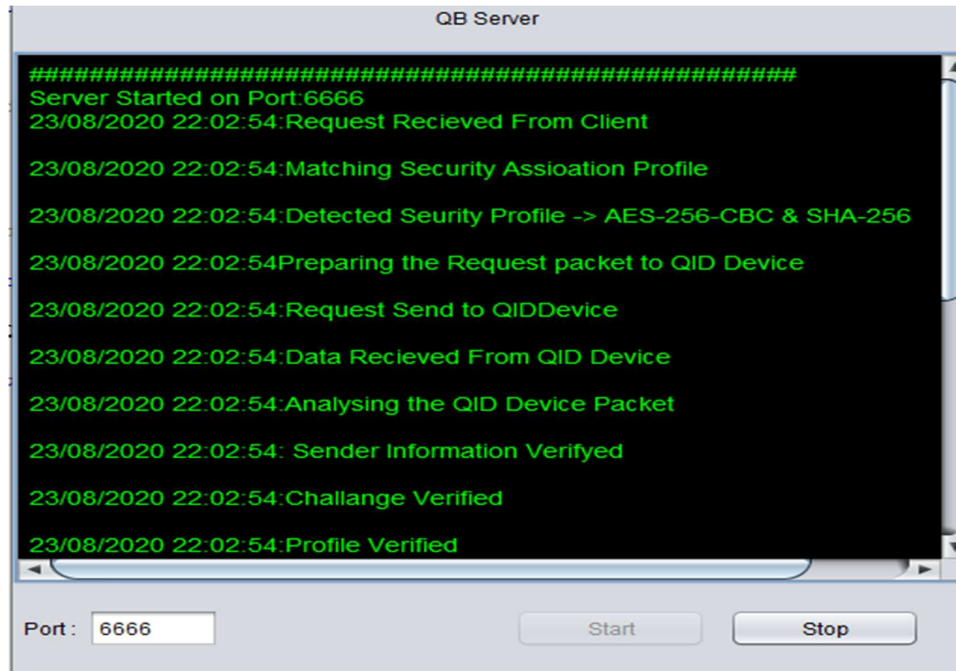


Figure. 3.20: QB Server Test Run.

- **QID Device**

The application for QID Device is designed as a console application since most of the IOT devices are automated. The application has the capability to connect the QB Server, handle request and responses. The primary goal of this application is to successfully transfer data to the QB Server in a safe and secure manner by following the designed framework.

```
deps-jar:
Updating property file: E:\NetBeans-11.3\NetBeans Projects\QIDDevice\build\build-jar.properties
Compiling 1 source file to E:\NetBeans-11.3\NetBeans Projects\QIDDevice\build\classes
compile-single:
run-single:
Welcome to QID Device
Detected Profile:11:Encryption: AES-256-CBC & Signature: SHA-256
Data Send & Connection Closed
```

Figure. 3.21: QID Device Test Run.

Chapter 4 : Framework Analysis and Testing

4.1. Framework evaluation

This chapter focus on the evaluation and analysis of the framework which include performance, network and security evaluation. This analysis is aimed to reduce vulnerabilities in the design and to minimise the risks posed by the threats.

4.2. Performance evaluation

Performance analysis is crucial when any software is designed. It gives useful information about the resource consumption of various entities. This information can be later used to optimize the program which improves the overall experience of the end users. In this case performance analysis is performed on the QID Device program. Since the program runs on resource constraint devices proper optimization is required.

In the implementation, The QID Device is capable of encrypting the data with two encryption standards and a single signature standard. Since IOT devices run on various types of hardware, A memory analysis is performed on the QID Device program using JProfiler tool. This analysis showcases the utilization of RAM for a single request and for 10000 data request instances. Here a request is defined as a complete operation cycle i.e. From filtering the incoming data to transmitting outgoing data.

- **Using AES-256-CBC & SHA-256 (Profile -1)** – This is the default option that is preferred for the consumer. As seen in the below image, The signing and encryption operations peaked at 4MB. However during the initial data segregation and creation of keys, Only 3MB of memory was utilized.

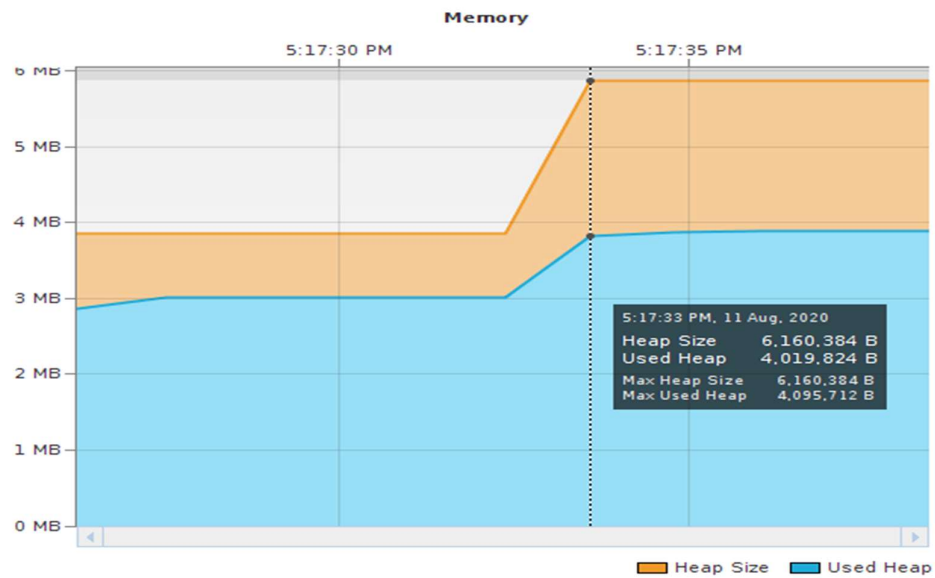


Figure. 4.1: Single Request analysis of profile-1.

For 10000 data requests, The peak memory utilization is around 5.5 MB as shown in the Figure 4.2. The average utilization is around 4.5 MB.

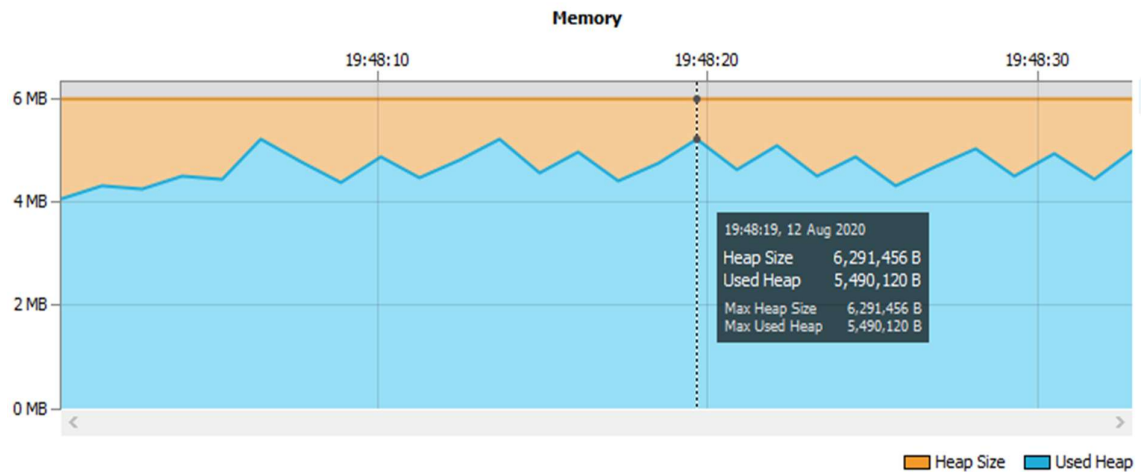


Figure. 4.2: Multi Request analysis of profile-1.

- **Using AES-256-GCM & SHA-256 (Profile -2)** – The memory performance is similar to AES-256-CBC. However, Time taken to finish processing this security option is faster which is shown in the graph below but consumes slightly more RAM than the Profile-1.

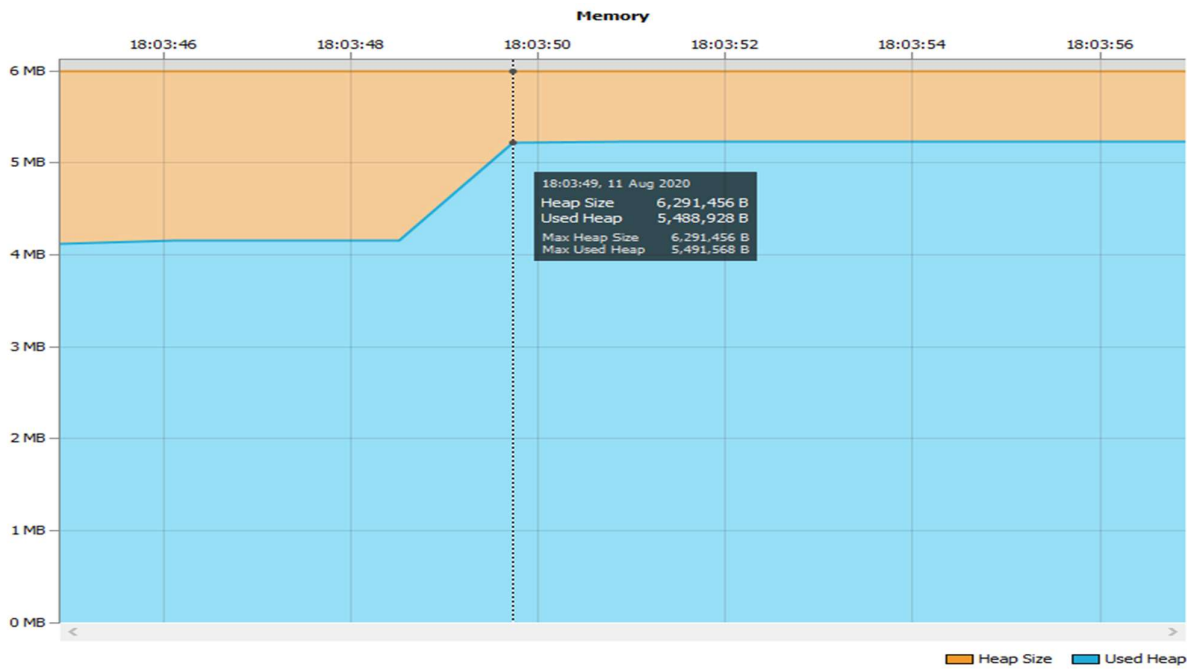


Figure. 4.3: Single Request analysis of profile-2.

AES-256-GCM combined with the SHA-256 is faster in handling bulk requests. Approximately 30 seconds is the time taken for processing the data. The program utilized a maximum of 5.5 MB which is ideal for IOT devices operating with minimum space as shown in the Figure 4.4.

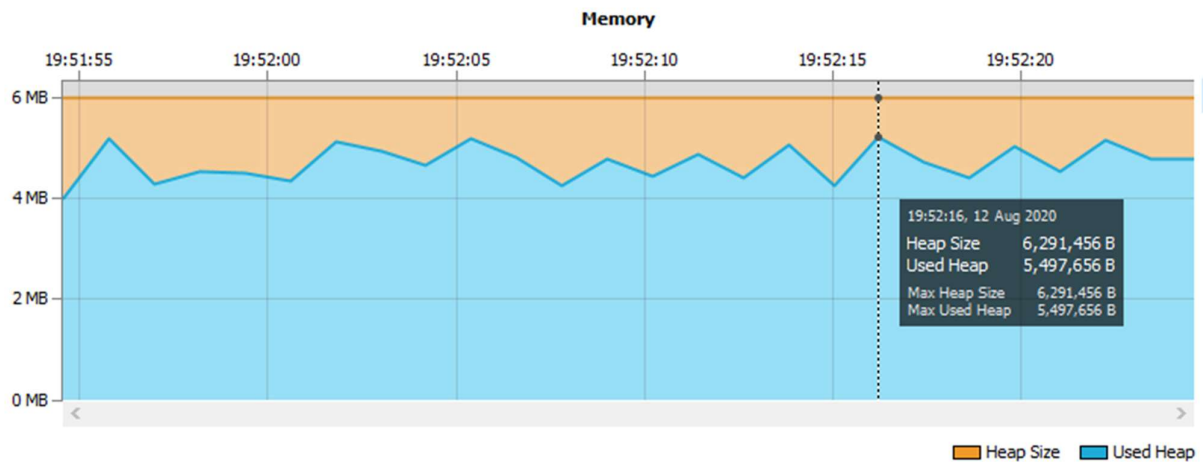


Figure. 4.4: Multiple request analysis of profile-2.

- **Using AES-256-CBC without Signature (Profile -3)** –This option is preferable if the end user requests device status and informative services. Only 3.9 MB of memory is utilized for its operations.

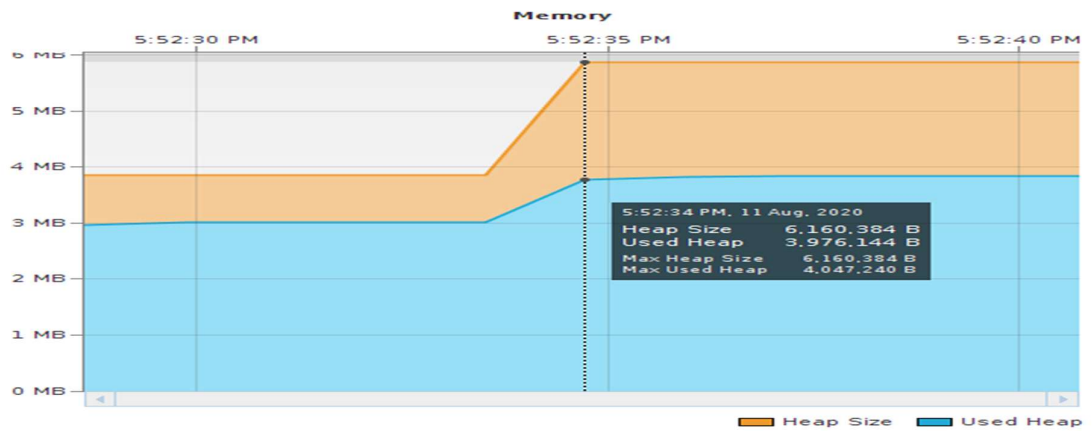


Figure. 4.5: Single request analysis of profile-3.

For 10000 requests, Comparing between profile-1 with profile-3 it can be observed that the resource utilization is almost equal.

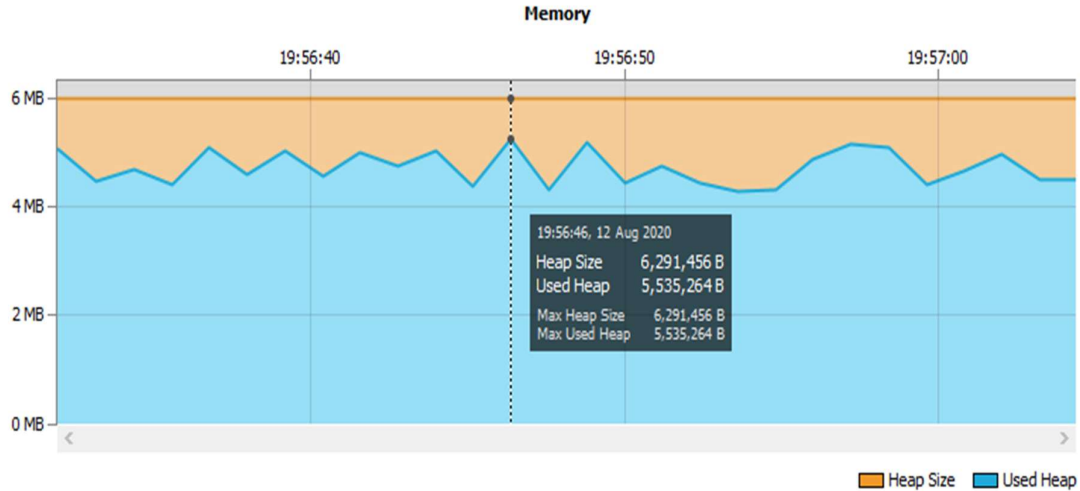


Figure. 4.6: Multiple request analysis of profile-3.

- **Using AES-256-GCM without Signature (Profile -4)** – This profile utilizes 5.4 MB of memory which is significantly high compared to profile-3. Also the time taken for computation is almost similar to profile-3.

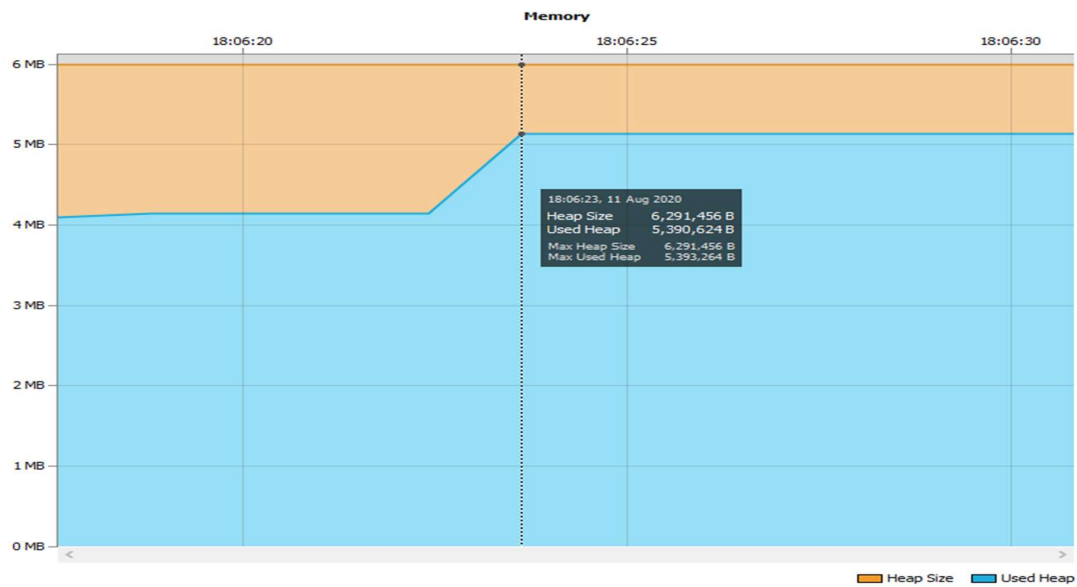


Figure. 4.7: Single request analysis of profile-4.

Since AES-256-GCM is obviously faster, The same can be observed in the Figure 4.8. The peak memory utilization was around 5.5MB which is similar to the performance of multiple requests in profile-3.

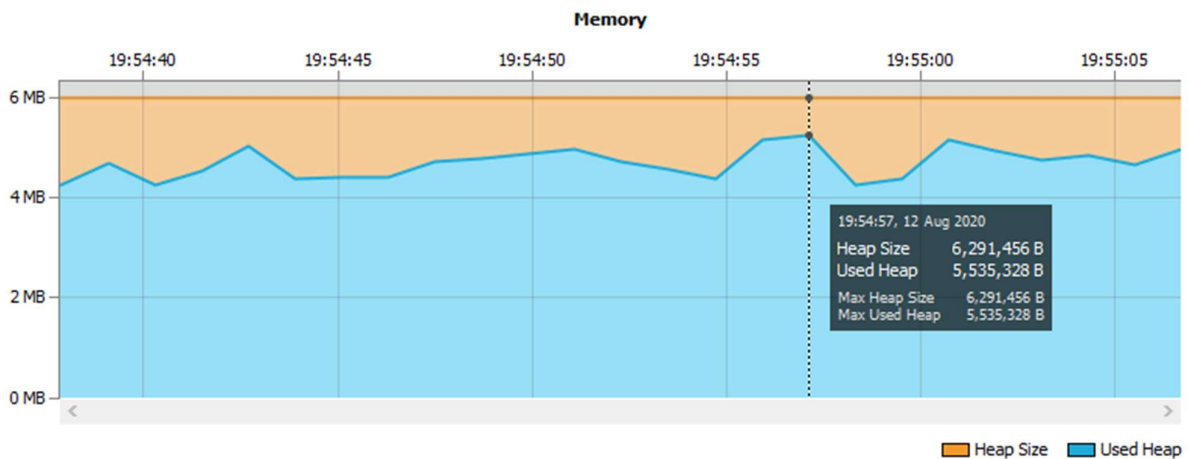


Figure. 4.8: Multiple request analysis of profile-4.

The performance analysis showed the resource utilization of the designed protocol. Overall, The protocol showed efficient results and can be implemented in IOT Devices.

4.3. Network Analysis

Network analysis tends to identify the network vulnerabilities in a system. Wireshark tool is used to sniff the packets that are transmitting between various entities present in the framework. The aim of this analysis is to determine whether a malicious actor is able to recreate the cryptographic keys and to check if any disclosed information compromises the protocol. Also, Analysis is conducted between the communication of QB Server and QID Device. Since the connection between the Consumer and QID Device is assumed to be trusted for analysis.

After a analysis, The identified data fields that a malicious actor can sniff through the network is as follows:

- **Time Stamps** – The header information is transmitted in plain text. So attacker can note the time intervals.
- **Unique ID of QID Device& QB Server** – This variable is transmitted to the QID Device by the QB Server. This aids attacker to tag the QID Device and QB Server in the network.
- **Requested Security Options** – The Malicious actor knows the security options chosen for the data security.
- **Challenge**
- **Server Nonce** – This field is randomly generated by the server and used generate the signature key.
- **Encrypted Data**–Attacker can perform crypt analysis on the encrypted data.
- **Unsigned header information from QID Device to QB Server** –Since signature for the header information is absent, Malicious attacker can read and modify the data packet.

From the above network analysis it can be observed that the with the give information disclosed in the network, It is not possible for the attacker to recreate the cryptographic keys. However, it is easy for the attacker to modify the data packet because of the lack of signature for the body of the packet transmitting from QID Device to QB server. But the attacker will be unsuccessful to read the data.

4.4. Security Analysis

This section focuses on assessing the different risks that the framework is vulnerable to. This process helps to provide a solution for mitigation of threats that can be harmful for the framework. The methodologies that are used for risk analysis are STRIDE and DREAD.

4.4.1. Overview of STRIDE and DREAD

STRIDE was developed by Microsoft as a thread modelling framework. It classifies threats as Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege. Each of these threat poses risk to the system such as data tampering takes place if there is lack of integrity check.

Similar to STRIDE, DREAD is a risk assessment methodology. DREAD focuses of Damage, Reproducibility, Exploitability, Affected users and Discoverability of attacks. Each threat is categorised and a value is calculated which is a rating that indicates the priority of the necessity changes that need to take place.

4.4.2. STRIDE Analysis

Using Microsoft Threat Modelling tool [46], A model of the framework is designed as shown in the Figure 4.9. This model is created using the generic objects present in the tool which helps with the analysis. As shown in the below figure, The trust boundary marks the entities that are trusted by the QB Server. It can be observed that the response and request packets from the QID Device and QB Server are not trusted by default. Whereas in this scenario it is assumed that QB Server trusts the Consumer. The Mobile Client mention in the model is the Consumer node which initiates the data request. QB Server has the following databases that are required for various operations. Also it acts like a proxy between Consumer and QID Device. QID Device is the IOT Device which has the PUF and the CRP database.

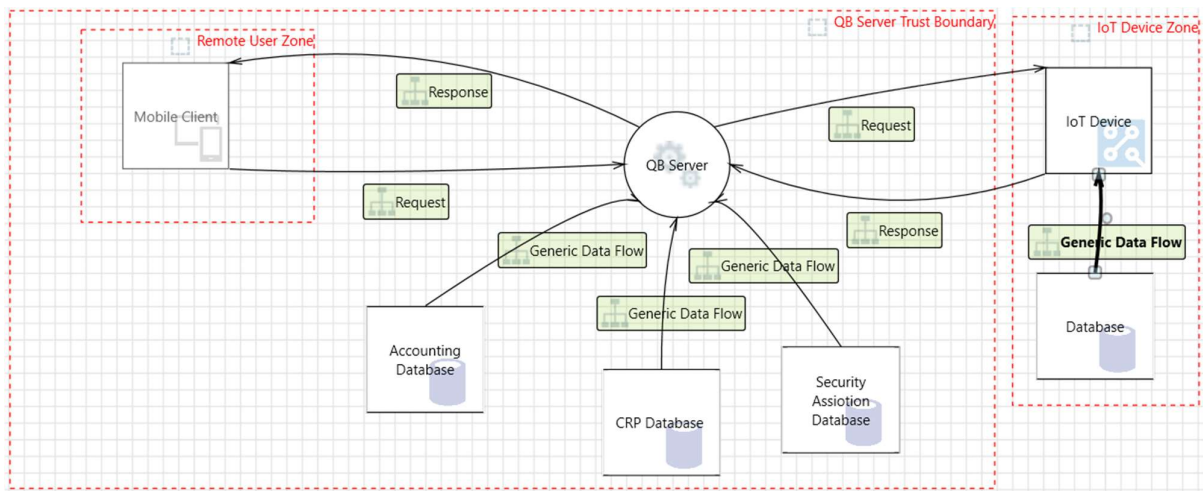


Figure. 4.9: STRIDE Threat Model.

With the combination of this information, STRIDE analysis has been performed. The key areas of focus are as follows:

- **Process** – Focuses on the operation such as protocol, cryptographic functions and log processing of QB Server and QID Device
- **Store** – Focuses on databases, log storage and data storage of QB Server and QID Device.
- **Flow** – Communications between QB Server and QID Device.

The threats that are identified in the STRIDE analysis are shown in the Table 4.1.

Framework	Process	Store	Flow
Spoofing	-	-	- Response packet from QID Device to QB Server
Tampering	-	-	- Header information altering from QID Device to QB Server.
Repudiation	-Logs not present on QID Device	-Logs not present on QID Device	-Logs not present on QID Device.
Information Disclosure	-Crypto keys are stored in Memory	-Data is stored in plain text.	-Information such as Challenge, Timestamps, Security Profile Data, QB Server - QID Device unique names, Signature & Address

Denial of Service	-	-	- Flood QB Server and QID Device with requests and responses.
Elevation of Privilege	-Runs as Admin	-Runs as Admin	-Runs as Admin - Possibility to execute a BOF on QB Server via QID Device response.

Table 4.1: STRIDE Analysis

After the analysis, It had been identified that the framework possess vulnerabilities. The controls that were identified for the vulnerabilities are as follows:

- Implementation of signing of response packet to QB Server from QB Client to prevent spoofing and tampering.
- Implementation of different run levels and user management to prevent elevation of privilege.
- Encrypt the databases in both QB Server and QID Client to prevent information disclosure.
- Securely flushing out generated cryptographic keys from the memory.
- Whitelisting to block Denial of Service attacks from malicious actors.
- Data logging at QID Device. Optionally encrypt the logs for security by compromising on the energy of the device.

With the above mentioned controls in place the identified threats can be minimized.

4.4.3. DREAD Analysis

For the DREAD Analysis, OWASP top 10 list of vulnerabilities [47] and OWASP IOT top 10 list of vulnerabilities [48] are considered. Scales are required for DREAD analysis which differ from various organizations. The scales of the five categories for assessing this framework is as follows [49]:

- **Damage Potential**

0 – No Damage.

3 – Single user data is compromised, affected or availability denied.

5 – A group of users data is compromised, affected or availability denied.

7 – All user data is compromised, affected or availability denied.

8 – Availability of server components is denied.

9 – Underlying server management system and infrastructure data is compromised or affected.

10 – Complete system or data destruction, failure or compromise

- **Reproducibility**

0 – Very hard.

5 – Using tools or custom scripts.

10 – Can exploit using a web browser.

- **Exploitability**

0 – Exploitable but takes a lot of time, skills and computation.

1 – Even with direct knowledge of the vulnerability we do not see a viable path for exploitation.

2 – Advanced techniques required, custom tooling.

5 – Exploit is available/understood, usable with only moderate skill.

7 – Exploit is available/understood, usable by non-authenticated users.

10 – Exploitable just by web browser or Simple tool.

- **Affected Users**

0 – None

5 – Users of specific component or entity.

10 – All the users.

- **Discoverability**

0 – Very hard to detect.

5 – Can figure out by analysing the network traces.

9 – Discoverable by using search engine or exploit databases.

10 – Information is available everywhere.

Using these values the risk is calculated by summation of scales of Damage Potential, Reproducibility, Exploitability, Affected Users and Discoverability. The summated value is then divided by 5. The output value is known as DREAD Score. Based on this score, The severity is classified as following:

- **Trivial** – Score ranging from 0 – 3
- **Important** – Score ranging from 4 – 7
- **Critical** – Score ranging from 8 – 10

The following are the threats arranged based on their severity that are identified which posses a potential risk to the framework which is represented in Table 4.2.

Threat	D	R	E	A	D	Total	Rating
DDOS on QB Server	10	5	10	10	10	45	9
DDOS on QID Device	8	5	10	5	10	38	7.6
QB Server takeover through Buffer overflow	10	5	2	10	5	32	6.4
Data Breach of QB Server through SQL Injection	7	5	5	10	5	32	6.4
Flaws in Signature Scheme	7	5	5	5	9	31	6.2
Flaws in Encryption Standards	10	5	1	10	0	26	5.2
Flaws in QID Authentication	10	0	1	10	0	21	4.2
Side Channel Attack on QID Device	10	0	1	10	0	21	4.2

Table 4.2: DREAD Analysis

Based on the values, There are two threats that range in “**Critical**” category which needs an immediate fix i.e. DDOS on QB Server and QID Device. The remaining threats range in “**Important**” category that can be fixed as a priority. These threats can be controlled by implementing measure such as:

- The discoverability of DDOS is very high. So a firewall with an access control list can be installed in QB Server to filter the false messages. Also, A access control whitelist can be used for QID Device to prevent this attack.
- Proper filter program and implementation of canaries can prevent buffer overflow attacks.
- Sanitization of the incoming data packet fields must be kept in check. This prevents a malicious attacker from exploiting the QB Server.
- By using stable libraries and constantly updating the libraries can nullify the flaws in cryptographic libraries.
- Side channel attacks can be prevented by taking proper physical security measures for QID Device.

STRIDE and DREAD analysis helps to identify and mitigate threats and risks that are potentially harmful to the system. Using these outputs, The framework can be enhanced with better security controls thus making it harder for malicious actor to compromise the system.

Chapter 5 : Conclusion

The designed framework ensures secure transfer of data from the QID Device to the consumer. This is achievable through the unique properties of the PUF. The PUF's are extremely useful mechanisms to achieve various cryptographic operations like authentication, secret key generation with a very small fingerprint. Also, a presence of a PUF in an IoT device ensures the characteristics of unclonability. The framework utilizes these characteristics for efficient resource consumption. Also, the designed framework proves to have a low resource cost compared with the other implementations which were discussed in the earlier sections. In addition, The proposed framework gives freedom for the users to choose wide range of security standards. This allows easy of use for various applications and IoT devices present in the market.

The System was evaluated for network analysis to determine whether an attacker is able to recreate the cryptographic keys through the response packets of the QID device. A renowned network analysis tool called WireShark is used to capture the network traffic flowing from QB server to QID device and Vice versa. When analysed it is found that the attacker was unsuccessful in recreating the cryptographic keys. Similarly to identify the vulnerabilities and threats posed for this frame work, a security analysis is carried out using STRIDE and DREAD analysis. During the STRIDE analysis the attacker was able to modify the header information of the data packet which is transmitted from QID device to QB server. Also there is a lack of log management in the QID device. In the event of a data breach, it is very hard to track and trace the incident. It has been identified in the DRED analysis that the DDOS attack is very effective in disrupting the frame work. In the security analysis it is found that the QB server is more likely to be targeted by the attacker. The memory utilization is very low for the operations that are taken place in the QID devices which shows that the resource utilization and performance is found to very efficient. To test the performance of the developed programs for the IoT devices, A memory performance test is conducted where 10000 requests are transmitted to the QID Device. However, Since the memory capacity and processing power of the devices differ, The results of the experiment are limited to the test system only.

5.1. Project Implications

The framework is designed by upholding modularity as a primary objective. This gives wide range of options for the users to choose the security standards. The use of PUF's for an authenticated cryptographic data transfer is not something new. As mentioned in Table 1,

There are various implementations of this idea each with its own merits and demerits. However with the use of the PUF developed by Quantum Base, A robust, multi-functional framework is developed where the consumer has complete control over the data of QID Device. With this framework, Several smart devices can be securely managed over the internet. The following are the merits of the framework:

- Low computational and network cost
- Cryptographic keys are unique for each transaction
- Supports various cryptographic standards
- Faster data processing
- User friendly

5.2. Limitations & Future Work

Although this framework guarantees various merits, There are a handful of limitations that need to be considered. The following are some of the identified limitations of the designed framework:

- If QB Server gets compromised, Entire system is ineffective.
- Requires data signing from transmissions from QID Device to QB Server
- Framework need to be tested on real-time IoT Devices
- There is no log management system present in the QID Device. So, In the case of a breach it is very hard to investigate.

The limitations can be resolved by conducting research in the area of data privacy and server management. These limitations can also be solved in future work . This framework can be enhanced by focusing on the following:

- Conduct energy profile analysis on various cryptographic standards that might be suitable to add to the framework. This gives user more freedom and control over their data.
- A tracking and control mechanism should be implement for the data stored in different entities for data privacy.
- A protocol for mutual authenticated data transfer between IoT Devices.

References

1. Pappu, R., 2001. Physical One-Way Functions. Available at : <https://pdfs.semanticscholar.org/3097/28aa9bc6052c242d9217b750059ce7cc84cd.pdf>
2. Wang, X., Tehranipoor, M., 2010. Novel Physical Unclonable Function with process and environmental variations. Available at : https://www.researchgate.net/publication/221340113_Novel_Physical_Unclonable_Function_with_process_and_environmental_variations
3. Gassend, B., 2003. Physical Random Functions. Available at : <http://dspace.mit.edu/handle/1721.1/37606?show=full>
4. Suh, G.F., Devadas, S., 2007. Physically unclonable functions for device authentication and secret key generation. Available at : <https://dl.acm.org/doi/10.1145/1278480.1278484>
5. Yin, C.E., Qu, G., 2009. Temperature-aware cooperation ring oscillator PUF. Available at : https://www.researchgate.net/publication/224584886_Temperature-aware_cooperative_ring_oscillator_PUF
6. Bauder, D.W., 1983. An anti-counterfeiting concept for currency systems. Research report PTK-11990 Sandia National Labs.
7. Ruhrmair, U., Hilgers, C., Urban, S., Agnes, W., Dinter, E., Forster, B., Jirauschek, C., 2013. Optical PUF's Reloaded. Available at : <https://eprint.iacr.org/2013/215.pdf>
8. Vrijaldenhoven, S.C.B.J., 2004. Acoustical physical unclonable functions. Available at : <https://research.tue.nl/en/studentTheses/acoustical-physical-uncloneable-functions>
9. Maes, R., 2010. Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions. Available at : https://www.researchgate.net/publication/226371108_Physically_Unclonable_Functions_A_Study_on_the_State_of_the_Art_and_Future_Research_Directions
10. Gao, Y., Ma, H., Said, F., Abbot, D., Ranasinghe, C., 2017. PUF-FSM: A Controlled Strong PUF. Available at : <https://ieeexplore.ieee.org/document/8010824>
11. Che, E., Lin, Y., Pan, A., Zhang, J., 2013. A Robust Hierarchical FSM Structure for Active IC Metering. Available at : <https://scialert.net/fulltext/?doi=itj.2013.1107.1115>

12. Rishab, N., John, S., 2012. A Theoretical Analysis: Physical Unclonable Functions and the Software Protection Problem. Available at :
<https://www.infona.pl/resource/bwmeta1.element.ieee-art-000006227678>
13. Hutle, M., Kammerstetter, M., 2015. Resilience Against Physical Attacks. Available at : <https://www.sciencedirect.com/science/article/pii/B9780128021224000043>
14. Wachsmann, C., Sadeghi, A., 2014. Physically Unclonable Functions (PUFs): Applications, Models, and Future Directions. Available at :
https://www.researchgate.net/publication/280323819_Physically_Unclonable_Functions_PUFs_Applications_Models_and_Future_Directions
15. Herder, C., Yu, M., Koushanfar, F., Devadas, S. 2014. Physically Unclonable Functions and Applications: A Tutorial. Available at :
<https://ieeexplore.ieee.org/document/6823677>
16. Tobisch, J., Becker, G.T., 2015, On the Scaling of Machine Learning Attacks on PUF;s with Applications to Noise Bifurcation. Available at :
https://www.researchgate.net/publication/299854285_On_the_Scaling_of_Machine_Learning_Attacks_on_PUFs_with_Application_to_Noise_Bifurcation
17. Zhang, J., Qu, G., Lv, Y.Q., Zhou, Q., 2014. A Survey on Silicon PUF's and Recent Advances in Ring Oscillator PUFs. Available at :
https://www.researchgate.net/publication/271997658_A_Survey_on_Silicon_PUFs_and_Recent_Advances_in_Ring_Oscillator_PUFs
18. Kurra, K., Nelakuditi, U., 2019. A secure arbiter physical unclonable functions (PUFs) for device authentication and identification. Available at:
<http://section.iaesonline.com/index.php/IJEEI/article/view/614>
19. Ruhrmair, U., Dijk, M., 2013. PUF's in Security Protocols: Attack Models and Security Evaluations. Available at : <https://ieeexplore.ieee.org/document/6547116>
20. Majzoobi, M., Elnably, A., 2010. FPGA Time-Bounded Unclonable Authentication. Available at : https://www.researchgate.net/publication/220722318_FPGA_Time-Bounded_Unclonable_Authentication
21. El, M., Fadlallah, A., Chamoun, M., Serhrouchni, A., 2019. A Survey of Internet of Things (IoT) Authentication Schemes. Available at :
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6427355/>
22. Overview of SAMLbyonelogin Developers. Available at:
<https://developers.onelogin.com/saml>
23. Dodanduwa, K., Kaluthanthri, I., 2018. Role of Trust in OAuth 2.0 and OpenID Connect. Available at :

- https://www.researchgate.net/publication/327392445_Role_of_Trust_in_OAuth_20_and_OpenID_Connect
24. How Does RADIUS Work? By Cisco, 2006. Available at :
<https://www.cisco.com/c/en/us/support/docs/security-vpn/remote-authentication-dial-user-service-radius/12433-32.html>
25. Gibson, S. 2019. Welcome to SQRL. Available at :
https://www.grc.com/sqrl/SQRL_Explained.pdf
26. Comparing and Contrasting the Function and Purpose of Authentication Services by simplilearn. Available at : <https://www.simplilearn.com/comparing-and-contrasting-the-function-tutorial>
27. A PRACTICAL USE CASE OF HOMOMORPHIC ENCRYPTION by Kontron.
Available at :
https://www.kontron.de/download/download?filename=/downloads/white_papers/usecase_homomorphic_encryption-web.pdf
28. Huynh, D., 2020. Homomorphic Encryption intro: Overview and Use Cases.
Available at: <https://towardsdatascience.com/homomorphic-encryption-intro-part-1-overview-and-use-cases-a601adcff06c>
29. Chen, S., Li, B., Cao, Y., 2019. Intrinsic Physical Unclonable Function (PUF) Sensors in Commodity Devices. Available at :
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6603541/>
30. Shoukry, Y., Martin, P., Tabuada, P., Srivastavan, M., 2013. Non-invasive Spoofing Attacks for Anti-lock Braking Systems. Available at :
https://link.springer.com/chapter/10.1007/978-3-642-40349-1_4
31. Braeken, A., 2018. PUF Based Authentication Protocol for IoT. Available at :
https://www.researchgate.net/publication/327131986_PUF_based_authentication_protocol_for_IoT
32. Yilmaz, Y., Gunn, R., Halak, B., 2018. Lightweight PUF-Based Authentication Protocol for IoT Devices. Available at : <https://www.c-iot.ecs.soton.ac.uk/sites/www.c-iot.ecs.soton.ac.uk/files/Yildiran%20Yilmaz%20C-IoT.pdf>
33. Wallrabenstein, R., 2015. Implementing Authentication Systems Based on Physical Unclonable Functions. Available at :
https://www.researchgate.net/publication/284188873_Implementing_Authentication_Systems_Based_on_Physical_Unclonable_Functions

34. Chatterjee, U., Subhra, R., Mukhopadhyay, D., 2016. A PUF-based Secure Communication Protocol for IoT. Available at : <https://eprint.iacr.org/2016/674.pdf>
35. Roberts, J., Bagci, E., Zawawi, M., Sexton, J., Hulbert, N., Noori, J., Young, M., Woodhead, S., Missous, M., Migliorato, A., Roedig, U., Young, R., 2015. Using Quantum Confinement to Uniquely Identify Devices. Available at : <https://rdcu.be/b6grj>
36. Venckauskas, A., Jusas, N., Kazanavicius, E., Stuikys, V., 2015. An Energy Efficient Protocol For The Internet Of Things. Available at : <https://doi.org/10.1515/jee-2015-0007>
37. Toldinas, J., Stuikys, V., Damasevicius, R., Ziberkas, G., Banionis, M., 2011. Energy Efficiency Comparison with Cipher Strength of AES and Rijndael Cryptographic Algorithms in Mobile Devices. Available at : https://www.researchgate.net/publication/265866230_Energy_Efficiency_Comparison_with_Cipher_Strength_of_AES_and_Rijndael_Cryptographic_Algorithms_in_Mobile_Devices
38. Rana, S., Hossain, S., Shoun, I., 2018. An Effective Lightweight Cryptographic Algorithm to Secure Resource-Constrained Devices. Available at : https://thesai.org/Downloads/Volume9No11/Paper_37-An_Effective_Lightweight_Cryptographic_Algorithm.pdf
39. Kumar, S., Gupta, P., 2014. A Comparative Analysis of SHA and MD5 Algorithm. Available at : https://www.researchgate.net/publication/263656830_A_Comparative_Analysis_of_SHA_and_MD5_Algorithm
40. Nakajima, J., Matsui, M. Performance Analysis and Parallel Implementation of Dedicated Hash Functions. Available at : https://link.springer.com/content/pdf/10.1007%2F3-540-46035-7_11.pdf [ff] [40]
41. Chandran, R., Manuel, M., 2015. Performance Analysis of Modified SHA-3. Available at: <https://core.ac.uk/download/pdf/82095933.pdf>
42. Bafandehkar, M., Yasin, M., Mahmood, R., Zurina, H., 2013. Comparison of ECC and RSA Algorithm in Resource Constrained Devices. Available at : https://www.researchgate.net/publication/286734225_Comparison_of_ECC_and_RSA_Algorithm_in_Resource_Constrained_Devices
43. Albela, M., Lamas, F., Carames, F., 2018. A Practical Evaluation on RSA and ECC-Based Cipher Suites for IoT High-Security Energy-Efficient Fog and Mist Computing Devices. Available at : <https://europepmc.org/article/PMC/6264011> [ii] [43]

44. Shah, A., Engineer, M.,2019. A Survey of Lightweight Cryptographic Algorithm's for IoT-Based Applications. Available at :
https://www.researchgate.net/publication/329072888_A_Survey_of_Lightweight_Cryptographic_Algorithms_for_IoT-Based_Applications_Proceedings_of_ICSI CCS-2018
45. NetBeans 11.3 by Apache.2020. Available at :
<https://netbeans.apache.org/download/nb113/index.html>
46. Microsoft Threat Modelling Tool by Microsoft. 2016. Available at :
<https://www.microsoft.com/en-in/download/details.aspx?id=49168>
47. OWASP Top Ten by The OWASP Foundation. 2020. Available at :
<https://owasp.org/www-project-top-ten/>
48. Misra, N.OWASP Top 10 IoT Vulnerabilities by Device Authority. Available at :
<https://www.deviceauthority.com/blog/owasp-s-top-10-iot-vulnerabilities>
49. OSSA-Metrics by OpenStack. Available at :
<https://wiki.openstack.org/wiki/Security/OSSA-Metrics>