Islamic University of Technology (IUT)

Department of Computer Science & Engineering

# Integrating authentication among IoT Devices having Perfect Forward Secrecy

**By**

**Md. Rahat Anwar Khan (154431)**

**H.M. Zakaria (154409)**

A thesis submitted to the department of Computer Science & Engineering in partial fulfillment of the requirements for the degree of BSc in Computer Science & Engineering

**Academic Year: 2018-2019**

**November 2019**

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Md. Rahat Anwar Khan and H.M. Zakaria under the supervision of Md. Sakhawat Hossen, Assistant Professor, Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

**Authors:**

___________________________

Md. Rahat Anwar Khan

Student ID: 154431

___________________________

H.M. Zakaria

Student ID: 154409

**Supervisor:**

___________________________

Md. Sakhawat Hossen

Assistant Professor

Department of Computer Science & Engineering

Islamic University of Technology

# Acknowledgement

We would like to express our grateful appreciation for Md. Sakhawat Hossen, Assistant Professor, Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT) for being our adviser and mentor. His motivation, suggestions and insights for this research have been invaluable to us. This research would never have been possible without his support and proper guidance. His valuable opinion, time and input provided throughout the thesis work, from first phase of thesis topics introduction, subject selection, proposing protocol, modification till the finalization which helped us to do our thesis work in proper way. We are really grateful to him.

# Abstract

Internet of Things (IoT), also referred to as the Internet of Objects, is envisioned as a transformative approach for providing numerous services. Compact smart devices constitute an essential part of IoT. They range widely in use, size, energy capacity, and computation power. However, the integration of these smart things into the standard Internet introduces several security challenges because the majority of Internet technologies and communication protocols were not designed to support IoT. Moreover, commercialization of IoT has led to public security concerns, including personal privacy issues, threat of cyberattacks, and organized crime. Security measurements must be taken into serious consideration for the IoT Infrastructure to prevent all kinds of nuisance. Most of the encryption protocols of IoT devices depend on long term secret .It is a major concern for the IOT infrastructure because somehow if the long term secret is compromised, then the intruder can easily decrypt session data. To avoid this adversity, perfect forward secrecy along with private key can make the encryption protocol much more strong. Then even after attaining perfect forward secrecy, the devices are not safe because of no authentication measure. Due to the absence of authentication step the devices can be cloned and replaced inside the infrastructure without admin’s knowledge. So to mitigate this calamity we have tried to integrate an authentication extension inside an existing secret key communication protocol among IoT devices.

# Contents

# 1. Introduction

## 1.1 Overview:

The Internet of Things is the concept of connecting any device (so long as it has an on/off switch) to the Internet and to other connected devices. The IoT is a giant network of connected things and people – all of which collect and share data about the way they are used and about the environment around them. That includes an extraordinary number of objects of all shapes and sizes – from smart microwaves, which automatically cook your food for the right length of time, to self-driving cars, whose complex sensors detect objects in their path, to wearable fitness devices that measure your heart rate and the number of steps you've taken that day, then use that information to suggest exercise plans tailored to you. There are even connected footballs that can track how far and fast they are thrown and record those statistics via an app for future training purposes.

Devices and objects with built in sensors are connected to an Internet of Things platform, which integrates data from the different devices and applies analytics to share the most valuable information with applications built to address specific needs. These powerful IoT platforms can pinpoint exactly what information is useful and what can safely be ignored. This information can be used to detect patterns, make recommendations, and detect possible problems before they occur. For example, if anyone own a car manufacturing business,he/she might want to know which optional components (leather seats or alloy wheels, for example) are the most popular. Using Internet of Things technology, he/she can: Use sensors to detect which areas in a showroom are the most popular, and where customers linger longest; Drill down into the available sales data to identify which components are selling fastest; automatically align sales data with supply, so that popular items don't go out of stock. The information picked up by connected devices enables me to make smart decisions about which components to stock up on, based on real-time information, which helps me save time and money. With the insight provided by advanced analytics comes the power to make processes more efficient. Smart objects and systems mean you can automate certain tasks, particularly when these are repetitive, mundane, time-consuming or even dangerous.

The Next Industrial Revolution which is going to change our lives in ways never imagined before, the last industrial revolution which is nothing but INTERNET the way we communicate and connect with people has changed like never before and also the Internet boom has improved our lives in many ways. Every time an Industrial Revolution happens there will be huge changes in the economy create a whole new level of markets. For humanity, which is moderately clutter by nature, the IoT is an extraordinary advancement. On the other hand, for individuals who esteem their security, the Man to Man helps in interconnecting different electronic gadgets.

IOT has arrived with a highly believable promise of giving individuals few more free hours by automating few tasks and boosting productivity of businesses by making better use of data. IoT security is important because many critical functions are entrusted to connected devices, and a sophisticated attack could easily lead to disastrous consequences. For example, on a smaller scale, hackers could gain entry to a smart-house by remotely disabling the security system. On a larger scale, hackers could gain control of utility grids and shut down electricity in a building or even a neighborhood. The primary reason companies struggle with securing IoT is that in their rush to get IoT devices to market, IoT device vendors may forgo security. Building in IoT security protocols from the beginning would be expensive and more labor-intensive, plus it might compromise the capabilities consumers want most. As a result, companies are forced to deal with devices with fewer built-in security considerations.

Most IoT devices have password authentication and basic security protocols, but that’s not enough. Since IoT devices are so specialized in size, scope, and complexity, many standard PC security solutions won’t work. The methods of network security that MSPs and companies are most familiar with—like firewalls or intrusion software—are built for brick-and-mortar IT infrastructures, not necessarily IoT protocols. Internet of things cybersecurity is also difficult to implement for five major reasons: Not enough resources to create a strong IoT security defense i.e. connected devices are often configured to execute one core process, and there just isn’t enough computing power devoted to securing IoT. Set it and forget it i.e. IoT devices typically go unpatched and update once they are turned on. Lack of established IoT security standards i.e. without a formal infrastructure or framework, the security standards in IoT devices are left up to individual manufacturers. Reliance on default credentials i.e. connected devices only work out-of-the-box if they use stock credentials, which are easily guessed by hackers. Similarly, IoT devices are usually produced en masse—if you can hack one, you can hack them all. Long product lifespan i.e. IoT devices remain in circulation for 15 to 20 years. Due to this long lifespan, they simply won’t be able to keep up with advancing security standards without updates. For these reasons, IoT devices are often left undefended and are easily exploited by bad actors.

Security researchers found that cyberattacks on IoT devices have jumped to 2.9 billion events per period so far in 2019, three times higher than in recent years. Some of these attacks are levied against the devices themselves, but cybercriminals are more likely to target lapses in IoT device security because they can be used as backdoor entry points into larger networks.

## 1.2 Problem Statement:

The primary goal of perfect forward secrecy in the communication of IOT devices is to make the communication much more secure. There won't be any problem even if primary key is compromised. Because perfect forward secrecy ensures that in every session, a new session key is produced which can't be compromised in that session and this key along with long term key is needed to break the system. But if the authentication step is not present then even after attaining Perfect Forward Secrecy the communication is not safe. Because if a cloned device enters into the IoT infrastructure the system loses it can't be identified without authentication procedure. So, we want to integrate authentication among IoT devices having Perfect Forward Secrecy.

# 1.3 Motivation and scopes:

The numbers of IoT devices are increasing day by day. Back in 2007 the number of connected IoT devices in the world surpassed the total number of Human Beings present on Earth. Since then the growth has escalated manifold. Presently for every human being present on earth there are four IoT devices in response. The IoT devices also have widespread uses. Starting from people's phones, tablets, wearables to smart car, smart home, smart grid, smart road, smart retailing system, smart automation everything is now possible due to IoT's growth. The usage sectors of IoT devices can be easily characterized by people, vehicles, homes, towns & cities, commerce and industrial sectors. With the increasing numbers of devices the number of attacks on IoT devices has also increased gradually. In recent years the attacks are skyrocketing and the havoc & devastation it creates are enormous. Few recent examples are: Stuxnet, Mirai botnet, cold in Finland, brickerbot, botnet barrage etc. If necessary preventive steps are not taken into consideration then the rate of attacks will keep on skyrocketing. Considering all these aspects, it is decisive to provide security measures for all the devices. This motivates us to work on the security layer of IoT.

# 1.4 Thesis Outline:

In Chapter 1 we have discussed our study in a precise and concise manner. Chapter 2 deals with the necessary literature review for our study and there development so far. In Chapter 3 we have stated the skeleton of our proposed method, proposed work and how our solution is being integrated inside the protocol. Chapter 4 shows the results and comparative analysis of successful integration of our proposed method. The final segment of this study contains the conclusion section and probable future works on it.

# 2. Literature Review

## 2.1 Few Terminologies

**Heterogeneous IOT Devices:** Heterogeneous IoT refers to the platform (the smart side) to allow communicating with a wide variety of devices using multiple protocols (MQTT, CoAP, Modbus, etc). It is a feature which allows more devices by more vendors to be handled by the platform, and prevents lock-in by the platform vendor

**Perfect Forward Secrecy:** Perfect Forward Secrecy is a feature of specific key agreement protocols that gives assurances your session keys will not be compromised even if the private key of the server is compromised. By generating a unique session key for every session a user initiates, even the compromise of a single session key will not affect any data other than that exchanged in the specific session protected by that particular key. Perfect Forward Secrecy represents a huge step forwards in protecting data on the transport layer and following on from Heartbleed, everyone using SSL/TLS should be looking to implement it.

**Authentication:** Authentication is the process of determining whether someone or something is, in fact, who or what it declares itself to be. Authentication technology provides access control for systems by checking to see if a user's credentials match the credentials in a database of authorized users or in a data authentication server. Users are usually identified with a user ID, and authentication is accomplished when the user provides a credential, for example a password, that matches with that user ID. Most users are most familiar with using a password, which, as a piece of information that should only be known to the user, is called a knowledge authentication factor. .

**Private-public key encryption**: The Public and Private key pair comprise of two uniquely related cryptographic keys (basically long random numbers). Below is an example of a Public Key:

3048 0241 00C9 18FA CF8D EB2D EFD5 FD37 89B9 E069 EA97 FC20 5E35 F577 EE31 C4FB C6E4 4811 7D86 BC8F BAFA 362F 922B F01B 2F40 C744 2654 C0DD 2881 D673 CA2B 4003 C266 E2CD CB02 0301 0001

The Public Key is what its name suggests - Public. It is made available to everyone via a publicly accessible repository or directory. On the other hand, the Private Key must remain confidential to its respective owner.

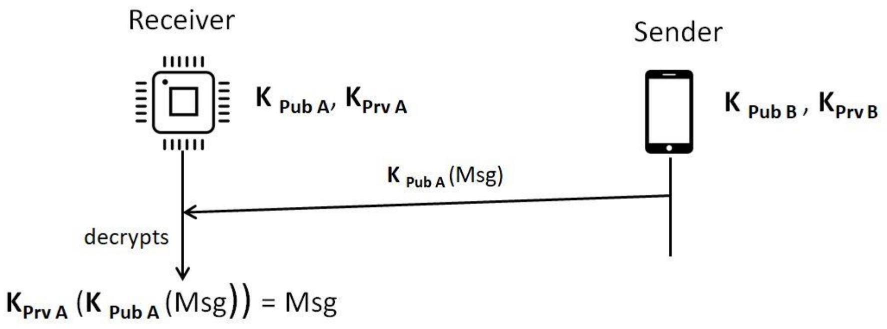


**Figure 1: Public-Private Key Encryption**

Because the key pair is mathematically related, whatever is encrypted with a Public Key may only be decrypted by its corresponding Private Key and vice versa.

For example, if Bob wants to send sensitive data to Alice, and wants to be sure that only Alice may be able to read it, he will encrypt the data with Alice's Public Key. Only Alice has access to her corresponding Private Key and as a result is the only person with the capability of decrypting the encrypted data back into its original form.

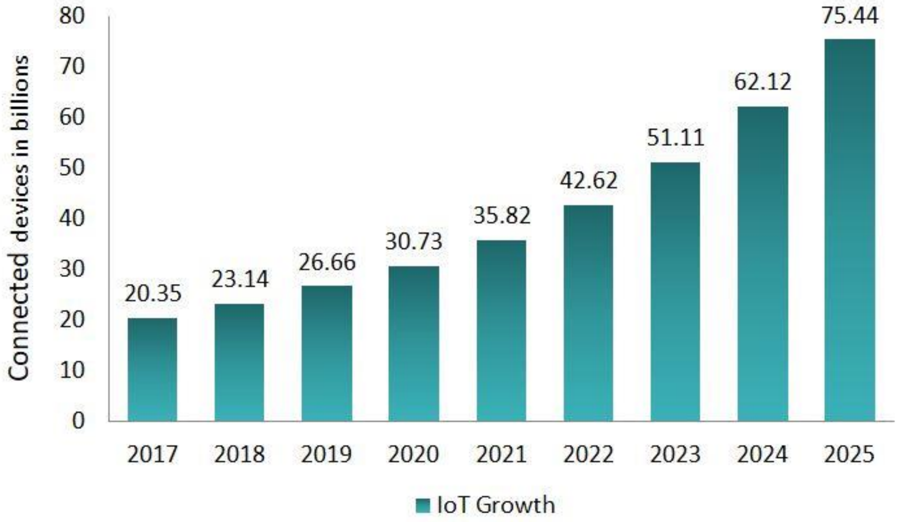
As only Alice has access to her Private Key, it is possible that only Alice can decrypt the encrypted data. Even if someone else gains access to the encrypted data, it will remain confidential as they should not have access to Alice's Private Key.

**Symmetric key:** In cryptography, a symmetric key is one that is used both to encrypt and decrypt information. This means that to decrypt information, one must have the same key that was used to encrypt it. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link. This requirement that both parties have access to the secret key is one of the main drawbacks of symmetric key encryption, in comparison to public-key encryption. Asymmetric encryption, on the other hand, uses a second, different key to decrypt information.

**Hashing:** It is generating a value or values from a string of text using a mathematical function. Hashing is one way to enable security during the process of message transmission when the message is intended for a particular recipient only. A formula generates the hash, which helps to protect the security of the transmission against tampering. Hashing is also a method of sorting key values in a database table in an efficient manner.

## 2.2 IoT and its growth

An important inflection point occurred in 2008, when the number of things connected to the Internet surpassed the human population. The adoption rate of the IoT is trending to be at least five times faster than the adoption of electricity and telephony, shown in Figure 1. This equates to about six things for every person on earth. A interesting trend contributing to the growth of the IoT is the shift from the consumer-based IPv4 Internet of tablets and laptops, that is, Information Technology (IT), to an Operational Technology (OT)-based IPv6 Internet of Machine-to-Machine interactions. This includes sensors, smart objects and clustered systems (for example, Smart Grid).



Source: IHS Statista 2018

**Figure 2: IoT Growth**

From a technology perspective, there are three main drivers that contribute to the growth of the IoT:

- **Ubiquitous Computing:** With intelligence in things at the edge, e.g., lightweight operating systems such as TinyOS running on very small computing platforms
- **Ubiquitous use of IP:** with convergence of protocols to run over IP rather than proprietary transports. Also greater adoption and support for IPv6 in carrier networks
- **Ubiquitous Connectivity:** Including cellular, radio and fixed. This includes low power, personal area wireless mesh networks particularly suited to sensors

Essentially, the enhancements and progress in these technologies have allowed the development of IoT devices such as sensors that have compute, storage and network capabilities built into extremely small form factors with low energy requirements.

The use of IoT has broadened into different sectors. We can divide them into major 4 categories. Starting from a person's wearable watch to hospital sensors, IoT is present everywhere. This distribution is shown in the following figure:

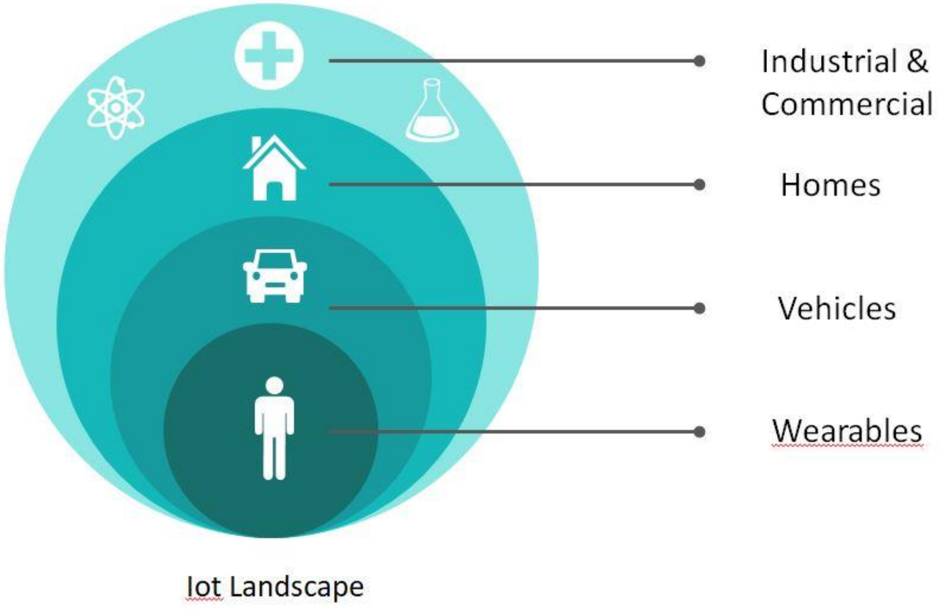


**Figure 3: IoT Distribution**

Researchers and early adopters have been further encouraged by advancements in wireless technologies, including radio and satellite; miniaturization of devices and industrialization; and increasing bandwidth, computing, and storage power. All of this provides an opportunity to reduce management and operational costs by converting these systems from the legacy platforms, such as Modbus or other serial communication protocols, to an IP-enabled infrastructure.

# 2.3 IoT Reference Model

IoT Reference Model’s purpose is to provide clear definitions and descriptions that can be applied accurately to elements and functions of IoT systems and applications. This reference model according to CISCO:

**Simplifies**: It helps break down complex systems so that each part is more understandable.

**Clarifies**: It provides additional information to precisely identify levels of the IoT and to establish common terminology.

**Identifies:** It identifies where specific types of processing is optimized across different parts of the system.

**Standardizes:** It provides a first step in enabling vendors to create IoT products that work with each other.

**Organizes:** It makes the IoT real and approachable, instead of simply conceptual.

In an IoT system, data is generated by multiple kinds of devices, processed in different ways, transmitted to different locations, and acted upon by applications. The IoT reference model according to CISCO is comprised of seven levels. Each level is defined with terminology that can be standardized to create a globally accepted frame of reference.

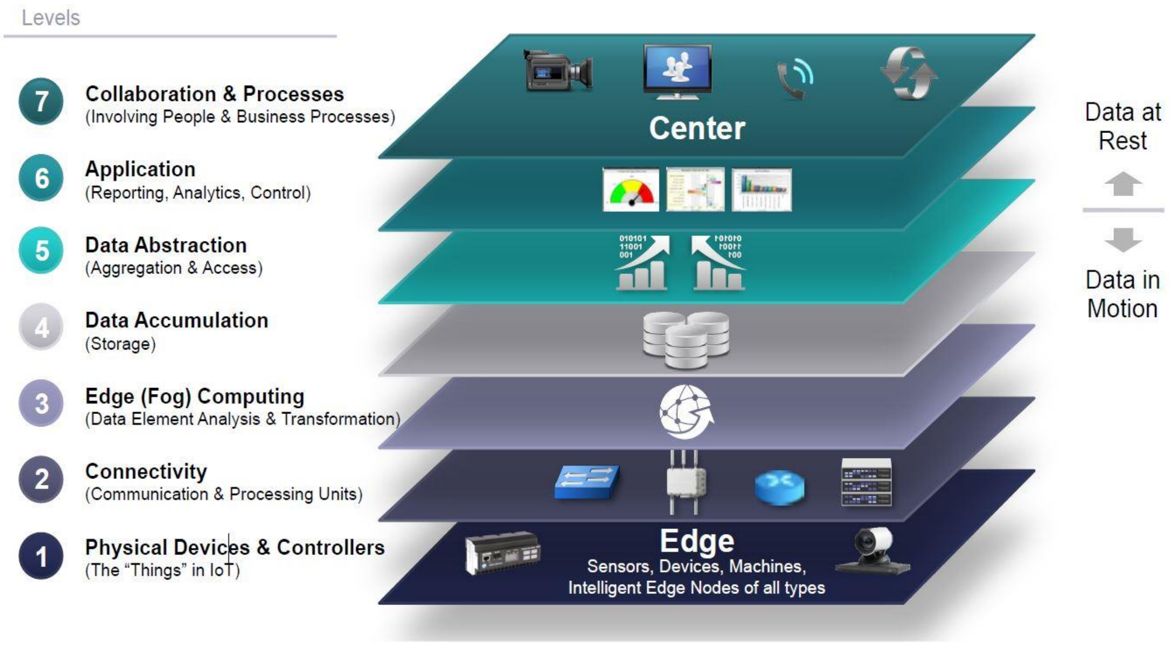

Levels
7 Collaboration & Processes
(Involving People & Business Processes)
6 Application
(Reporting, Analytics, Control)
5 Data Abstraction
(Aggregation & Access)
4 Data Accumulation
(Storage)
3 Edge (Fog) Computing
(Data Element Analysis & Transformation)
2 Connectivity
(Communication & Processing Units)
1 Physical Devices & Controllers
(The "Things" in IoT)
Center
Edge
Sensors, Devices, Machines,
Intelligent Edge Nodes of all types
Data at Rest
Data in Motion

**Figure 4: IoT Reference Model**

**Level 1, Physical Devices and Controllers:** The IoT Reference Model starts with Level 1 physical devices and controllers that might control multiple devices. These are the "things" in the IoT, and they include a wide range of endpoint devices that send and receive information. Today, the list of devices is already extensive. It will become almost unlimited as more equipment is added to the IoT over time.

**Level 2: Connectivity:** Communications and connectivity are concentrated in one level—Level 2. The most important function of Level 2 is reliable, timely information transmission. This includes transmissions:

i) Between devices (Level 1) and the network

ii) Across networks (east-west)

iii) Between the network (Level 2) and low-level information processing occurring at Level 3

**Level 3: Edge (Fog) Computing:** The functions of Level 3 are driven by the need to convert network data flows into information that is suitable for storage and higher level processing at Level 4 (data accumulation). This means that Level 3 activities focus on high-volume data analysis and transformation.

**Level 4: Data Accumulation:** Networking systems are built to reliably move data. The data is "in motion." Prior to Level 4, data is moving through the network at the rate and organization determined by the devices generating the data. The model is event driven.

**Level 5: Data Abstraction:** IoT systems will need to scale to a corporate—or even global—level and will require multiple storage systems to accommodate IoT device data and data from traditional enterprise ERP, HRMS, CRM, and other systems. The data abstraction functions of Level 5 are focused on rendering data and its storage in ways that enable developing simpler, performance-enhanced applications.

**Level 6: Application:** Level 6 is the application level, where information interpretation occurs. Software at this level interacts with Level 5 and data at rest, so it does not have to operate at network speeds. The IoT Reference Model does not strictly define an application. Applications vary based on vertical markets, the nature of device data, and business needs.

**Level 7: Collaboration and Processes:** One of the main distinctions between the Internet of Things (IoT) and IoT is that IoT includes people and processes. This difference becomes particularly clear at Level 7: Collaboration and Processes. The IoT system, and the information it creates, is of little value unless it yields action, which often requires people and processes.

# 2.4 IoT Security Threats

**Botnets:** A botnet is a network that combines various systems together to remotely take control over a victim's system and distribute malware. Cybercriminals control botnets using Command-and-Control-Servers to steal confidential data, acquire online-banking data, and execute cyber-attacks like DDoS and phishing. Cybercriminals can utilize botnets to attack IoT devices that are connected to several other devices such as laptops, desktops, and smartphones.

**Denial of service:** A denial-of-service (DoS) attack deliberately tries to cause a capacity overload in the target system by sending multiple requests. Unlike phishing and brute-force attacks, attackers who implement denial-of-service don't aim to steal critical data. However, DoS can be used to slow down or disable a service to hurt the reputation of a business. For instance, an airline that is attacked using denial-of-service will be unable to process requests for booking a new ticket, checking flight status, and canceling a ticket. In such instances, customers may switch to other airlines for air travel. Similarly, IoT security threats such as denial-of-service attacks can ruin the reputation of businesses and affect their revenue.

**Man-in-the-Middle:** In a Man-in-the-Middle (MiTM) attack, a hacker breaches the communication channel between two individual systems in an attempt to intercept messages among them. Attackers gain control over their communication and send illegitimate messages to participating systems. Such attacks can be used to hack IoT devices such as smart refrigerators and autonomous vehicles.

**Identity and data theft:** Attackers can also exploit vulnerabilities in IoT devices that are connected to other devices and enterprise systems. For instance, hackers can attack a vulnerable IoT sensor in an organization and gain access to their business network. In this manner, attackers can infiltrate multiple enterprise systems and obtain sensitive business data. Hence, IoT security threats can give rise to data breaches in multiple businesses.

**Social engineering:** Hackers use social engineering to manipulate people into giving up their sensitive information such as passwords and bank details. Alternatively, cybercriminals may use social engineering to access a system for installing malicious software secretly. Usually, social engineering attacks are executed using phishing emails, where an attacker has to develop convincing emails to manipulate people. However, social engineering attacks can be simpler to execute in case of IoT devices.

**Advanced persistent threats:** Advanced persistent threats (APTs) are a major security concern for various organizations. An advanced persistent threat is a targeted cyber-attack, where an intruder gains illegal access to a network and stays undetected for a prolonged period of time. Attackers aim to monitor network activity and steal crucial data using advanced persistent threats. Such cyber-attacks are difficult to prevent, detect, or mitigate.

**Ransomware:** Ransomware attacks have become one of the most notorious cyber threats. In this attack, a hacker uses malware to encrypt data that may be required for business operations. An attacker will decrypt critical data only after receiving a ransom. Researchers have demonstrated the impact of ransomware using smart thermostats. With this approach, researchers have shown that hackers can turn up the temperature and refuse to go back to the normal temperature until they receive a ransom.

**Remote recording:** Documents released by WikiLeaks have shown that intelligence agencies know about the existence of zero-day exploits in IoT devices, smartphones, and laptops. These documents imply that security agencies were planning to record public conversations secretly. These zero-day exploits can also be used by cybercriminals to record conversations of IoT users. For instance, a hacker can attack a smart camera in an organization and record video footage of everyday business activities.

# 2.5 IoT and its security history:

**January 2010, Stuxnet malware attack:** Stuxnet was a virus that exploited a zero-day vulnerability in the Windows operating system. In early 2010, the virus was detected in computers that hosted programmable logic controllers (PLCs) connected to nuclear centrifuges in Iran. Before it could be contained, the virus had sabotaged hundreds of centrifuges.

**Lesson learned:** The stuxnet incident served as an early example regarding the vulnerability of industrial systems connected to the internet.

**July 2015, Jeep hacked by researchers:** Researchers revealed that they had found various ways of exploiting connected Jeep vehicle systems. Researchers found ways to exploit the onboard entertainment system’s Wi-Fi by brute forcing all possible combinations in its weak password generation system. This exploit eventually led them to gain remote access to critical systems such as steering and braking. The event led Fiat Chrysler to recall more than 1.4 million vehicles.

**Lesson learned:** Wi-Fi passwords must be complex and able to withstand brute force attacks. Best practice is to use long phrases that are difficult to guess yet easy to remember (e.g. Weate8$5applepies). This hack also shows the importance of separating IoT systems from those that host critical systems.

**October 2016, Mirai botnet exploits IoT devices:** The Mirai botnet used hundreds of thousands of malware-infected IoT devices such as security cameras, routers, and smart thermostats to launch massive DDoS attacks that took down major websites such as GitHub, Netflix, and Spotify.

The malware took advantage of out-of-date firmware and scanned the internet for devices with open ports using default—and widely known—credentials. The scheme has inspired numerous copycat IoT botnets, many of them using the Mirai source code which was leaked to the internet.

**Lesson learned:** IoT devices often ship with default credentials that are widely known to those interested in exploiting them. Bad actors use automated programs to scan the internet for vulnerable devices and infect them with malware to form botnets. These zombified armies of IoT devices can then be used to deliver DDoS attacks capable of taking down major websites. Make sure your device isn’t using one of the username and password combos coded into Mirai and botnets bases on its source code.

**April 2017, BrickerBot malware:** Like the Mirai botnet, BrickerBot malware infected thousands of devices with default credentials. However, instead of using devices to launch attacks, BrickerBot destroys the device—or "bricks" it—by corrupting its memory, disrupting connectivity, and blocking all ports needed to update its firmware.

**Lesson learned:** Firmware must be updated as soon as new versions are made available. Unfortunately, updates for IoT devices aren't typically well publicized, meaning consumers and businesses must proactively stay up-to-date on IoT firmware availability.

**May 2018, VPNFilter router attack:** VPNFilter malware infected more than half a million devices—mostly consumer-grade routers—throughout the world. The malware monitored data transmitted through devices, stole passwords, and disabled devices. Some devices were salvageable after a reboot, but others were left inoperable.

**Lesson learned:** The VPNFilter episode serves as a warning to small businesses that use off-the-shelf consumer-grade network gear; these routers are far less secure than their commercial counterparts. Also, because malware is often stored in an IoT device's RAM, it can sometimes be removed with a simple reboot: Thus, it's good practice to switch them on and off every now and then.

**June 2018, Ships found vulnerable to hacking:** Researchers found that the navigation systems of many ships, from the smallest to the largest, are susceptible to attacks that could alter their GPS coordinates and knock them off course. They also found it possible to disable the navigation systems completely by remotely altering the firmware.

**Lesson learned:** IoT security is in its infancy, and some industry-specific applications might be more vulnerable than others. Buyers must engage IoT vendors and demand more secure IoT solutions.

# 2.6 Physically Unclonable Function

**Physically Unclonable Function** i.e. **PUF** is a physically-defined "digital fingerprint" that serves as a unique identity for a semiconductor device such as a microprocessor. PUF provides quantification of variations in hardware designed to be identical. It can’t be forged or predicted. Its error correction ensures stability.

In biometrics, every human has a fingerprint, the fingerprint of each individual person remains unique ; similarly, different instance of the same PUF design, fed with the same inputs – known as challenges – results with different outputs – known as responses – this is the basic principle of a PUF that can be exploited in different applications within security and random number generation. Once a PUF is fabricated, its set of challenges and responses – called the challenge response pairs (CRPs) – is fixed and unique to that instance. When a challenge is fed into a PUF, the same response results with every power-up, this means that the set of CRPs are practically stored in the physical structure of a PUF, that is in the dead silicon, and this feature can be advantageously used to replace non-volatile memories to store security keys in. This is not only more efficient in design area and power consumption, but also more secure as no continuous power-up is required to maintain any attack preventing circuitry

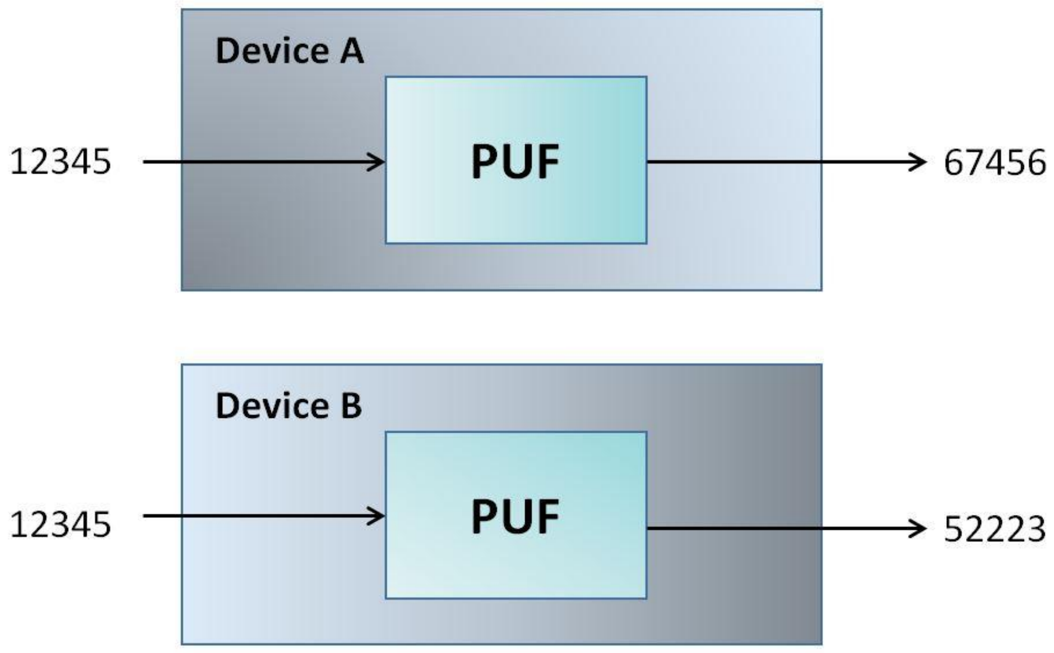


**Figure 5: Physically Unclonable Function**

PUF follows the following properties:

- $PUF_d$ : Challenge and Response pairs ($C\rightarrow R$) are such that

  -Ordered pairs ($c_i$ , $r_j$) defined by hardware variation of device d

- Ideal PUF Assumption:

  -For $PUF_d$ : $C\rightarrow R$

  - $H(c_i , r_j)$ distributed normally from 1 to n [ H is the Hamming Distance]

- Given $H(c_i , c_j)= 1$, $H(r_i , r_j) \approx n/2$

Theoretically, PUFs are unclonable as the name goes, but practically, there have been compromises using different methodologies to clone a PUF device and obtain its CRPs, which goes to show that this literature requires further research to improvise on the current meta.

**Emulation/Modeling Attacks**: The size of a CRP LUT of a memory-based PUF is limited to the number of memory cells it contains, which enables an adversary to read the finite responses and emulate the device. A suggestion against this is to use the configurable PUFs such as the TSRAM PUF; this would provide a larger CRP despite it being a memory-based PUF to help weight down the adversary from obtaining the entire CRP LUT in a feasible amount of time.

**Side-Channel Attacks:** Since the behavior of a PUF is dependent on the data processed, it can leak information, this is where an adversary can observe the behavior of the PUF's power consumption, electromagnetic radiation, or timing behavior to estimate the responses of the device. This type of attack usually targets a sub-component in the PUF, known as the fuzzy extractor, that is responsible for clarifying the response from any noise before allowing out of the device.

**Fault-Injection Attacks:** An adversary can exploit the limits of a PUF's robustness and unpredictability properties of the PUF under extreme physical conditions where the device is operated either under extreme environmental conditions or by varying its power supply, this injects a fault into the device, and if this is combined with crypto analysis, which is the process of deciphering coded messages without knowing the challenge, the response can be found.

**Invasive Attacks:** The idea behind an Invasive Attack is to either physically micro-probe the PUF, or to use reverse engineering techniques deduced from the circumvention of the device, both of which are methods of hardware analysis. This is the most expensive attacking methodology but is the most powerful despite the fact that many PUFs have been assumed to be tamper sensitive, meaning that invasive attacks would alter the PUF's response behavior permanently and notably.

# 3. Proposed Method:

## 3.1 Skeleton of Proposed Method

Our proposed solution is to integrate authentication in the protocol using Physically Unclonable Function Challenge-Response pair. It can be easily expressed as:

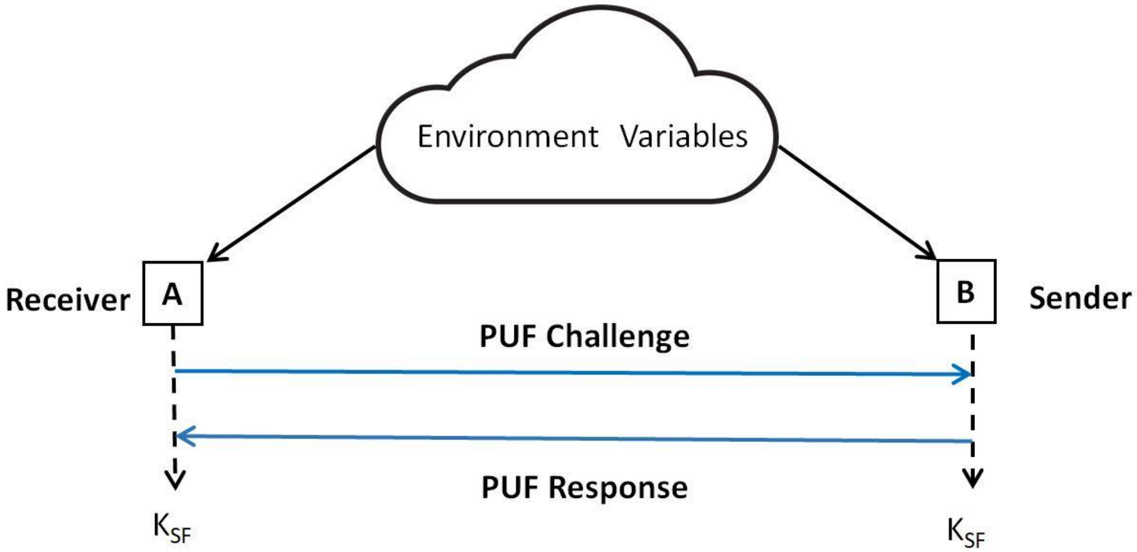


**Figure 6: Integration of PUF**

Here we can see that, receiver or server who is being contacted by sender throws a PUF challenge. In response, Sender replies with a PUF Response value. The server then matches the PUF Response value to its stored PUF Challenge-Response pairs. If the values match then the device is authenticated. Thus, authentication is done.

# 3.2 Proposed Model

In this section, we elaborately discuss about our proposed model. The model on which we want to integrate our extension step is a model which has attained Perfect Forward Secrecy where the session keys are generated from the common environmental variables of both sender and receiver. A brief representation of model is shown below:

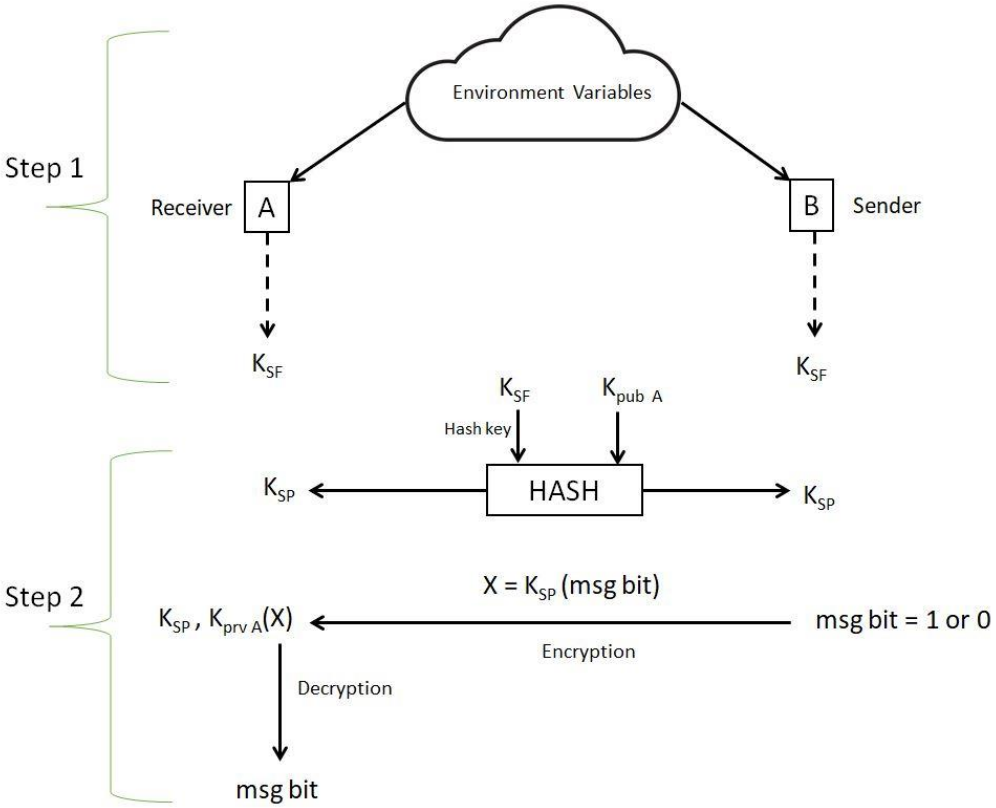


**Figure 7: 2 steps of the existing model**

In the existing model we can see that there are two steps. In 1st step, both sender and receiver generate a Session Final Key, $K_{SF}$. In 2nd step, from the created

Session Final Key and with available Public key it creates a Session Public Key. This Session Public Key $K_{SP}$ is used to encrypt the message bit and decryption of the message bit is done also by the $K_{SP}$ and the Private key.

## 1st Step:

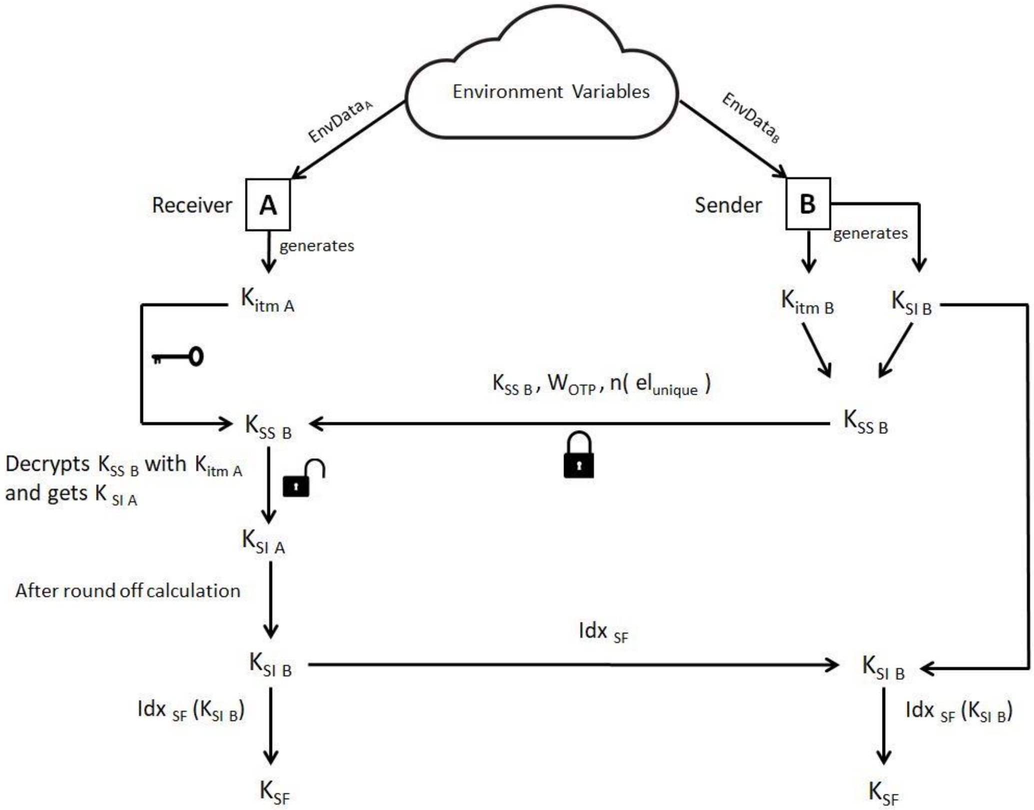


**Figure 8: First Step of the existing model**

## 2nd Step:

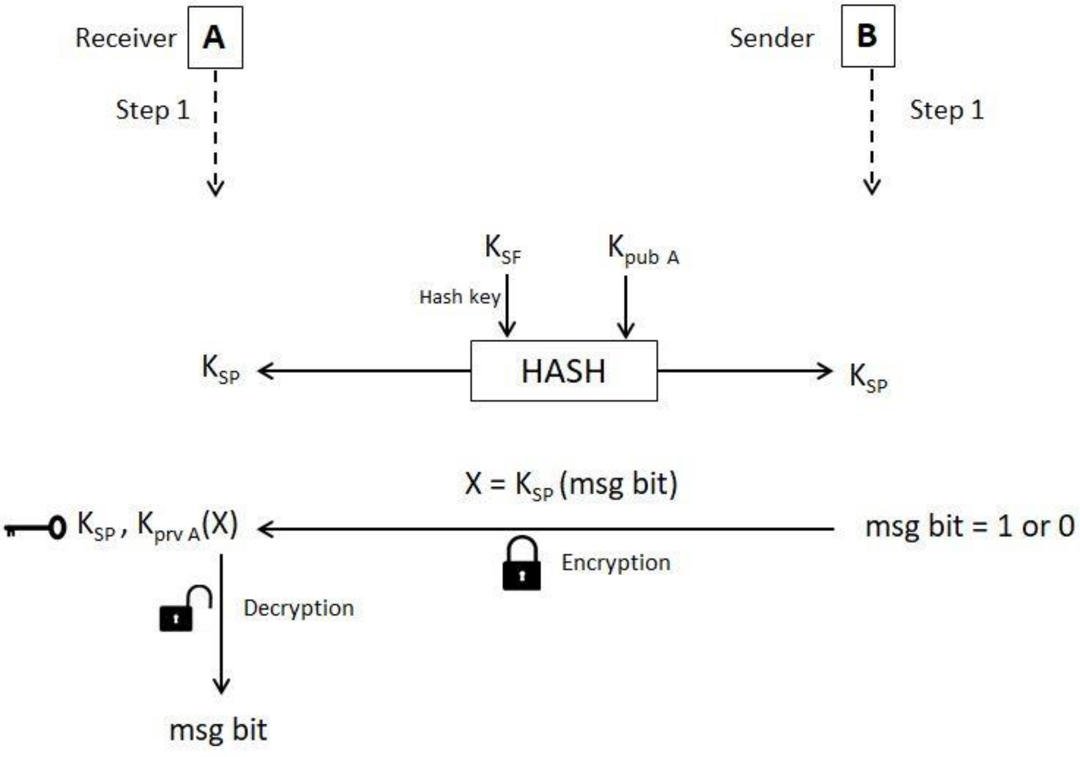


**Figure 9: Second Step of the existing model**

**Our Solution:** The aforementioned protocol and its steps already exist. Our solution is to integrate the Authentication procedure in this protocol. The way we integrate it is shown in the following figure:

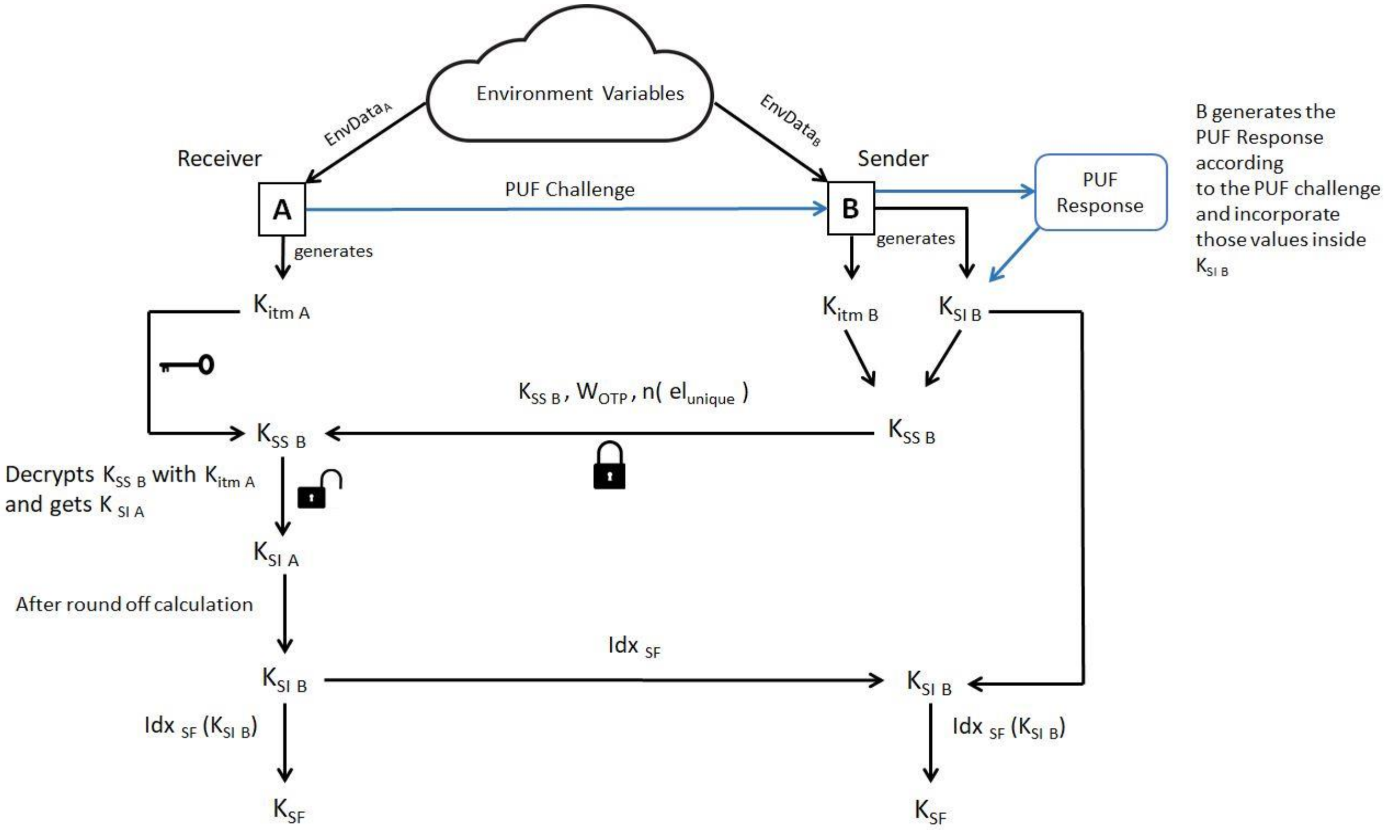


**Figure 10: Authentication integration on the existing model**

# 4. Result & Discussion

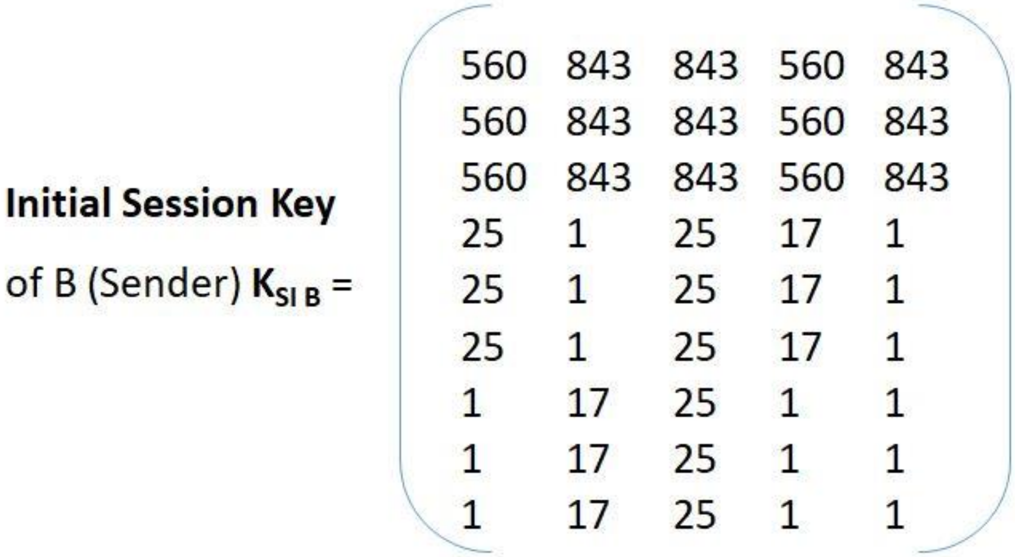
We conducted an experiment where we calculate the Session Final Keys using the existing protocol and our authentication extension. The results were as follows:

**Sender's Calculation:**

**Intermediate Key** of B (Sender), $\mathbf{K}_{\mathbf{itmB}} =$

$$\begin{pmatrix} 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 3 & 6 & 5 \\ 4 & 2 & 3 & 5 & 6 \\ 1 & 4 & 5 & 3 & 4 \\ 6 & 2 & 1 & 5 & 6 \\ 2 & 1 & 5 & 6 & 3 \\ 3 & 2 & 4 & 3 & 1 \\ 1 & 2 & 3 & 4 & 3 \\ 2 & 4 & 1 & 3 & 2 \end{pmatrix}$$

**Figure 11: Sender's Intermediate Key**



$$\textbf{Initial Session Key of B (Sender) } \mathbf{K}_{\mathbf{SI\,B}} = \begin{pmatrix} 560 & 843 & 843 & 560 & 843 \\ 560 & 843 & 843 & 560 & 843 \\ 560 & 843 & 843 & 560 & 843 \\ 25 & 1 & 25 & 17 & 1 \\ 25 & 1 & 25 & 17 & 1 \\ 25 & 1 & 25 & 17 & 1 \\ 1 & 17 & 25 & 1 & 1 \\ 1 & 17 & 25 & 1 & 1 \\ 1 & 17 & 25 & 1 & 1 \end{pmatrix}$$

**Figure 12: Sender's Initial Session Key**

Here, PUF challenges sent by Sender are 64, 83 and the response values are 560, 843 and $el_{unique} = \{25, 1, 17\}$

$\therefore n(el_{unique}) = 3$ &$W_{OTP} = 3$

Now Session Key Seed of B is generated from the above mentioned 2 matrices.

Generating **Session Key Seed** of B (Sender), $\mathbf{K_{SS\,B}}$ =

$$\begin{pmatrix}
(2*560) & (3*843) & (4*843) & (5*560) & (6*843) \\
(2*560) & (4*843) & (3*843) & (6*560) & (5*843) \\
(4*560) & (2*843) & (3*843) & (5*560) & (6*843) \\
(1*25) & (4*1) & (5*25) & (3*17) & (4*1) \\
(6*25) & (2*1) & (1*25) & (5*17) & (6*1) \\
(2*25) & (1*1) & (5*25) & (6*17) & (3*1) \\
(3*1) & (2*17) & (4*25) & (3*1) & (1*1) \\
(1*1) & (2*17) & (3*25) & (4*1) & (3*1) \\
(2*1) & (4*17) & (1*25) & (3*1) & (2*1)
\end{pmatrix}$$

Here, $\mathbf{K_{itm\,B}}$ is used as an **One Time Pad (OTP)** Key to encrypt the $\mathbf{K_{SI\,B}}$ Matrix**.** The encrypted value becomes $\mathbf{K_{SS\,B}}$

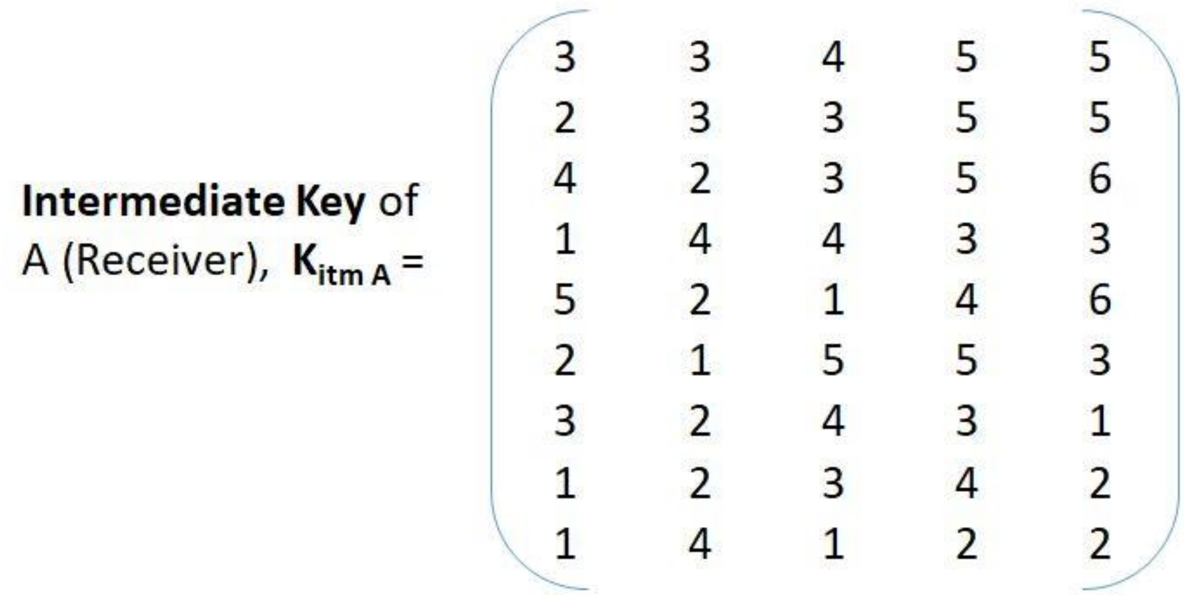
$$\Downarrow$$

$\therefore$ $\mathbf{K_{SS\,B}}$ =

$$\begin{pmatrix}
1120 & 2529 & 3372 & 2800 & 5058 \\
1120 & 3372 & 2529 & 3360 & 4215 \\
2240 & 1686 & 2529 & 2800 & 5058 \\
25 & 4 & 125 & 51 & 4 \\
150 & 2 & 25 & 85 & 6 \\
50 & 1 & 125 & 102 & 3 \\
3 & 34 & 100 & 3 & 1 \\
1 & 34 & 75 & 4 & 3 \\
2 & 68 & 25 & 3 & 2
\end{pmatrix}$$

**Figure 13: Generating Sender's Session Key Seed**

## Receiver’s Calculation:



$$\textbf{Intermediate Key} \text{ of A (Receiver), } \mathbf{K}_{\mathbf{itm\ A}} = \begin{pmatrix} 3 & 3 & 4 & 5 & 5 \\ 2 & 3 & 3 & 5 & 5 \\ 4 & 2 & 3 & 5 & 6 \\ 1 & 4 & 4 & 3 & 3 \\ 5 & 2 & 1 & 4 & 6 \\ 2 & 1 & 5 & 5 & 3 \\ 3 & 2 & 4 & 3 & 1 \\ 1 & 2 & 3 & 4 & 2 \\ 1 & 4 & 1 & 2 & 2 \end{pmatrix}$$

**Figure 14: Receiver's Intermediate Key**

Now, Receiver generates $\mathbf{K}_{\mathbf{SI\ A}}$ by decrypting $\mathbf{K}_{\mathbf{SS\ B}}$ using $\mathbf{K}_{\mathbf{itm\ A}}$ .

$\therefore\ \mathbf{K}_{\mathbf{itm\ A}}(\mathbf{K}_{\mathbf{SS\ B}}\ ) = \mathbf{K}_{\mathbf{SI\ A}}$

$$\text{Generating } \textbf{Initial Session Key} \text{ of A (Receiver), } \mathbf{K_{SI\,A}} = \begin{pmatrix} (1120/3) & (2529/3) & (3372/4) & (2800/5) & (5058/5) \\ (1120/2) & (3372/3) & (2529/3) & (3360/5) & (4215/5) \\ (2240/4) & (1686/2) & (2529/3) & (2800/5) & (5058/6) \\ (25/1) & (4/4) & (125/4) & (51/3) & (4/3) \\ (150/5) & (2/2) & (25/1) & (85/4) & (6/6) \\ (50/2) & (1/1) & (125/5) & (102/5) & (3/3) \\ (3/3) & (34/2) & (100/4) & (3/3) & (1/1) \\ (1/1) & (34/2) & (75/3) & (4/4) & (3/2) \\ (2/1) & (68/4) & (25/1) & (3/2) & (2/2) \end{pmatrix}$$

**Figure 15: Generating Receiver's Initial Session Key**

$$\therefore \mathbf{K_{SI\,A}} = \begin{pmatrix} 373 & 843 & 843 & 560 & 1011 \\ 560 & 1124 & 843 & 672 & 843 \\ 560 & 843 & 843 & 560 & 843 \\ 25 & 1 & 31 & 17 & 1 \\ 30 & 1 & 25 & 21 & 1 \\ 25 & 1 & 25 & 20 & 1 \\ 1 & 17 & 25 & 1 & 1 \\ 1 & 17 & 25 & 1 & 1 \\ 2 & 17 & 25 & 1 & 1 \end{pmatrix}$$

**Figure 16: Receiver's Initial Session Key**

Now receiver creates a frequency count of the elements of $K_{SI\,A}$. Receiver creates a frequency chart of the elements of $\mathbf{K_{SI\,A}}$ and takes the elements with highest frequencies.

Number of elements with highest frequencies it will choose = **n(el$_{unique}$)** + **n(response values)** = 3+2 = 5

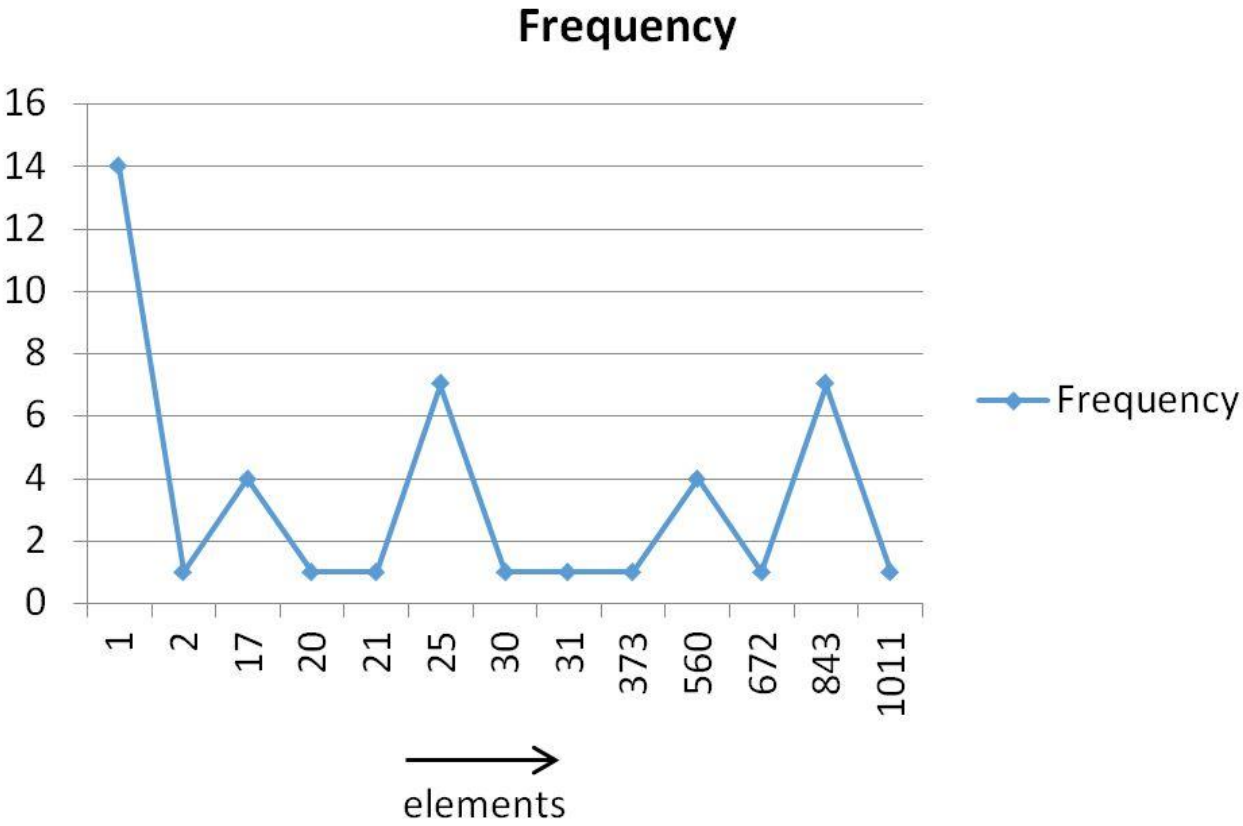


**Figure 17: Frequency of the elements**

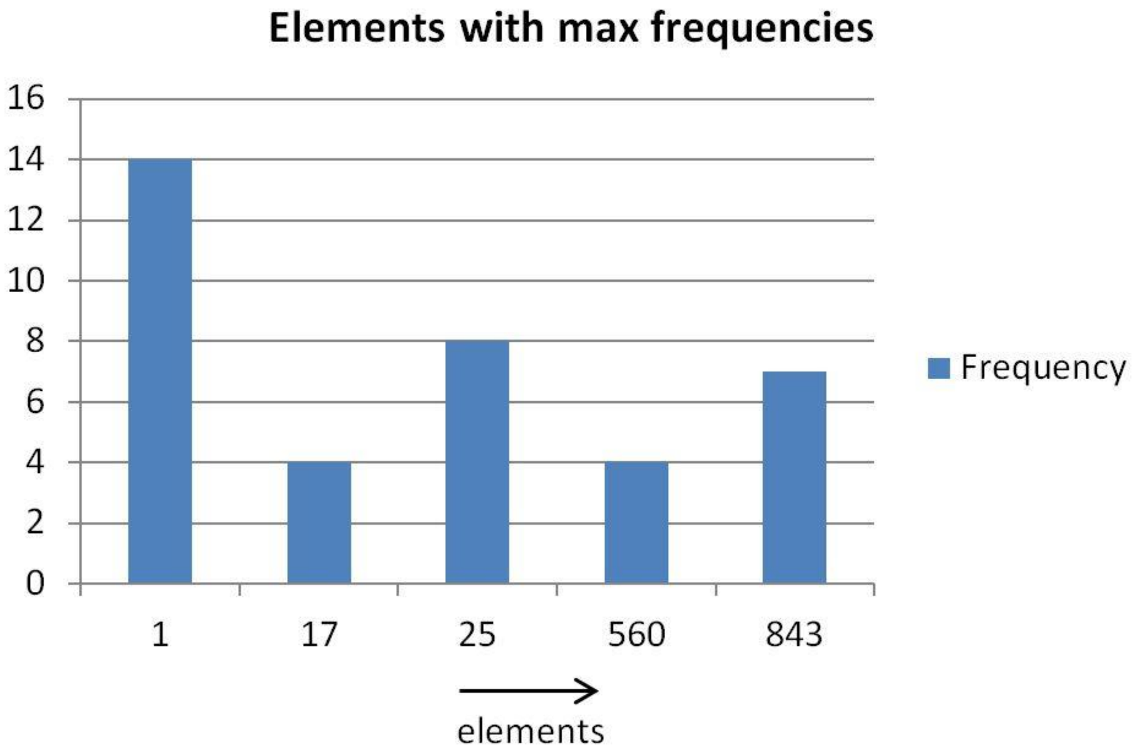
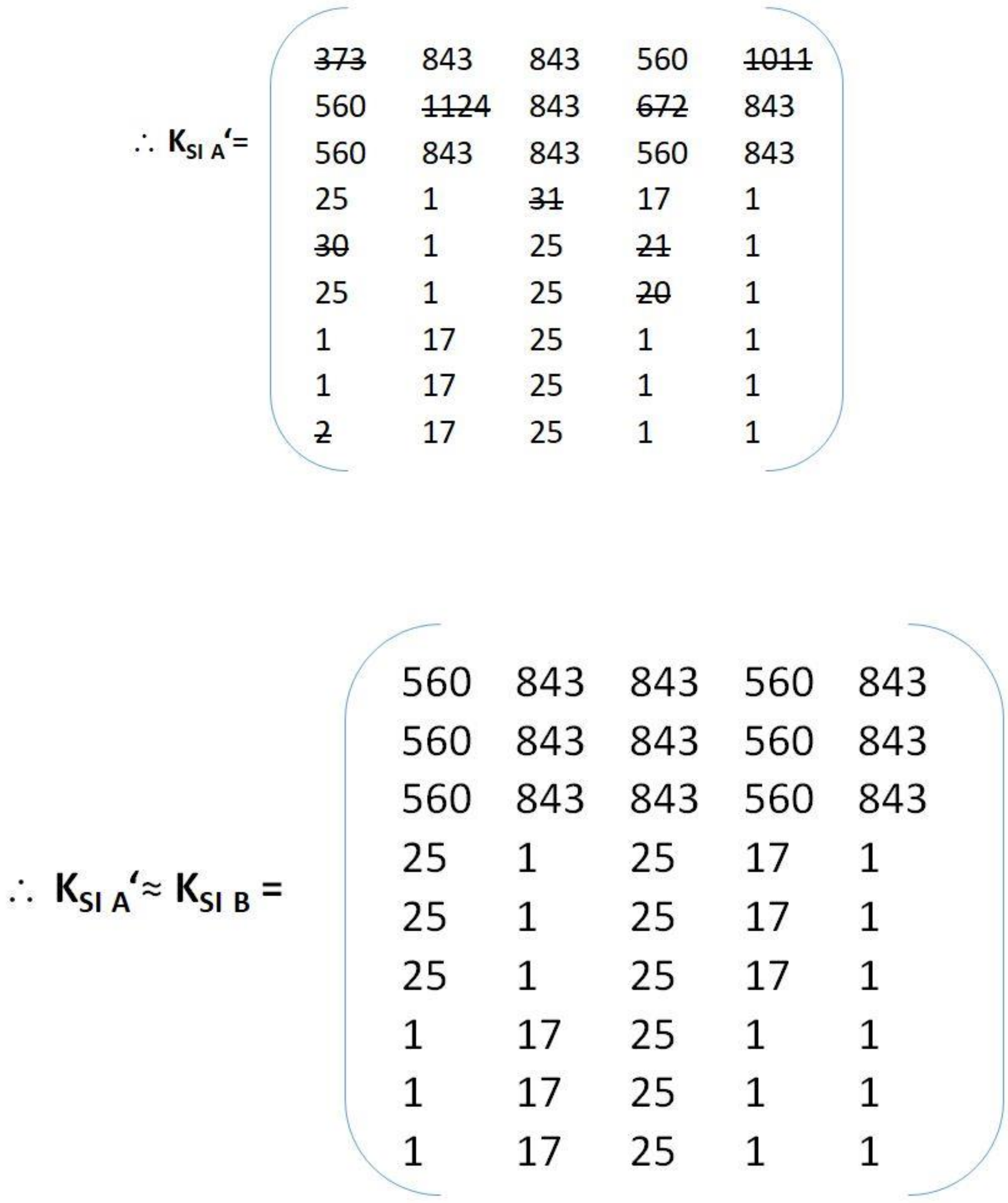


**Figure 18: Elements with max frequencies**

Now we replace the other elements with the selected elements. Thus our $\mathbf{K}_{\mathbf{SI\ A}}$ becomes almost as close to $\mathbf{K}_{\mathbf{SI\ B}}$. So after replacing with the values, the new matrix that we get is:



$$\therefore K_{SI\ A}' = \begin{pmatrix} \cancel{373} & 843 & 843 & 560 & \cancel{1011} \\ 560 & \cancel{1124} & 843 & \cancel{672} & 843 \\ 560 & 843 & 843 & 560 & 843 \\ 25 & 1 & \cancel{31} & 17 & 1 \\ \cancel{30} & 1 & 25 & \cancel{21} & 1 \\ 25 & 1 & 25 & \cancel{20} & 1 \\ 1 & 17 & 25 & 1 & 1 \\ 1 & 17 & 25 & 1 & 1 \\ \cancel{2} & 17 & 25 & 1 & 1 \end{pmatrix}$$

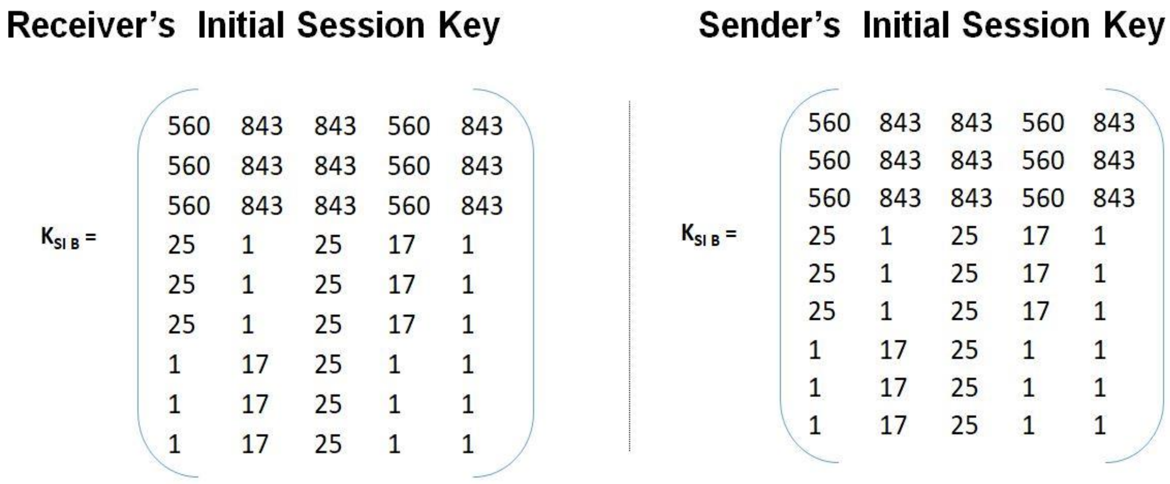
$$\therefore K_{SI\ A}' \approx K_{SI\ B} = \begin{pmatrix} 560 & 843 & 843 & 560 & 843 \\ 560 & 843 & 843 & 560 & 843 \\ 560 & 843 & 843 & 560 & 843 \\ 25 & 1 & 25 & 17 & 1 \\ 25 & 1 & 25 & 17 & 1 \\ 25 & 1 & 25 & 17 & 1 \\ 1 & 17 & 25 & 1 & 1 \\ 1 & 17 & 25 & 1 & 1 \\ 1 & 17 & 25 & 1 & 1 \end{pmatrix}$$

**Figure 19: Inital Session Key**

Here, we replaced the other elements with the selected elements and the **selected elements (max freq.)** are **1, 17, 25, 560, 843**.

And we can also notice that here 560 and 843 are PUF response values as they are present in the first row.

Finally we can see that both the Initial Session keys are same.



Receiver's Initial Session Key

$$K_{SI\ B} = \begin{pmatrix} 560 & 843 & 843 & 560 & 843 \\ 560 & 843 & 843 & 560 & 843 \\ 560 & 843 & 843 & 560 & 843 \\ 25 & 1 & 25 & 17 & 1 \\ 25 & 1 & 25 & 17 & 1 \\ 25 & 1 & 25 & 17 & 1 \\ 1 & 17 & 25 & 1 & 1 \\ 1 & 17 & 25 & 1 & 1 \\ 1 & 17 & 25 & 1 & 1 \end{pmatrix}$$

Sender's Initial Session Key

$$K_{SI\ B} = \begin{pmatrix} 560 & 843 & 843 & 560 & 843 \\ 560 & 843 & 843 & 560 & 843 \\ 560 & 843 & 843 & 560 & 843 \\ 25 & 1 & 25 & 17 & 1 \\ 25 & 1 & 25 & 17 & 1 \\ 25 & 1 & 25 & 17 & 1 \\ 1 & 17 & 25 & 1 & 1 \\ 1 & 17 & 25 & 1 & 1 \\ 1 & 17 & 25 & 1 & 1 \end{pmatrix}$$

**Figure 20: Similar Initial Session Keys**

∴ Both sender and receiver have similar matrix and the traversal of matrix data were completely encrypted.

Therefore, **560, 843** are the corresponding **PUF Response values** which server/ admin asked for. Now, server matches the values with its stored response values and if it matches then the device is validated otherwise not.

# 5. Conclusion & Future work

In the aforementioned protocol, generation of session key from the environmental variables is both resource and time costly. Our future goal is to create a protocol that ensures both Perfect Forward Secrecy and Authentication and produces Session Key from less number of steps. Thus, it can be implemented in real life as a cost effective solution too.

# References

[1] Abdullah-Al-Tariq et al. "Forward-Secrecy and Group Membership Verification using Behavioral Patterns of Heterogeneous IoT Devices.", Masters Thesis work, Department of Computer Science & Engineering, Islamic University of Technology

[2] A. Mosenia and N. K. Jha, “A comprehensive study of security of internet-of-things,” IEEE Transactions on Emerging Topics in Computing, vol. 5, pp. 586–602, Oct.-Dec. 2017.

[3] CISCO, The Internet of Things reference model, 2014. Available at http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf.

[4] S. H. J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, pp. 1735–1780, Nov. 1997.

[5] A. Al-Tariq, A. R. M. Kamal, et al., “A scalable framework for protecting user identity and access pattern in untrusted web server using forward secrecy, public key encryption and bloom filter,” Concurrency and Computation: Practice and Experience, vol. 29, no. 23, 2017.

[6] D. M. Mendez, I. Papapanagiotou, and B. Yang, “Internet of things: Survey on security and privacy,” arXiv preprint arXiv: 1707.01879, 2017.

[7] A. Brandt and J. Buron, Home automation routing requirements in low-power and lossy networks. [Online] Available: https://tools.ietf.org/html/rfc5826

[8] P. Dusart and S. Traor´e, “Lightweight authentication protocol for low-cost rfid tags,” in IFIP International Workshop on Information Security Theory and Practices, pp. 129–144, Springer, 2013.

[9] S. Satpathy, S. Mathew, V. Suresh and R. Krishnamurthy, "Ultra-low energy security circuits for IoT applications," 2016 IEEE 34th International Conference on Computer Design (ICCD), Scottsdale, AZ, 2016, pp. 682-685.

[10] M. Ye, N. Jiang, H. Yang and Q. Yan, "Security analysis of Internet-of-Things: A case study of august smart lock," 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Atlanta, GA, 2017, pp. 499-504.

[11] Maes R. (2013), “Physically Unclonable Functions: Properties”, Springer, Berlin, Heidelberg.

[12] Roel Maes, “PUF-Based Key Generation”. Chapter 6, Physically Unclonable Functions: Constructions, Properties and Applications, 2012.

[13] M. Al-Haidary and Q. Nasir, "Physically Unclonable Functions (PUFs): A Systematic Literature Review," 2019 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, United Arab Emirates, 2019, pp. 1-6.

[14] Maes R., Van Herrewege A., Verbauwhede I. (2012) PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator. In: Prouff E., Schaumont P. (eds) Cryptographic Hardware and Embedded Systems – CHES 2012. CHES 2012. Lecture Notes in Computer Science, vol 7428. Springer, Berlin, Heidelberg

[15] Paral, Zdenek & Devadas, Srinivas. (2011). Reliable and efficient PUF-based key generation using pattern matching. 2011 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2011. 128-133. 10.1109/HST.2011.5955010.

[16] J. Delvaux, D. Gu, D. Schellekens and I. Verbauwhede, "Helper Data Algorithms for PUF-Based Key Generation: Overview and Analysis," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 6, pp. 889-902, June 2015.

[17] C. Labrado and H. Thapliyal, "Design of a Piezoelectric-Based Physically Unclonable Function for IoT Security," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2770-2777, April 2019.

[18] A. R. Korenda, F. Afghah, B. Cambou and C. Philabaum, "A Proof of Concept SRAM-based Physically Unclonable Function (PUF) Key Generation Mechanism

for IoT Devices," 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Boston, MA, USA, 2019, pp. 1-8.

[19] A. Mohsen Nia, S. Sur-Kolay, A. Raghunathan and N. K. Jha, "Physiological Information Leakage: A New Frontier in Health Information Security," in IEEE Transactions on Emerging Topics in Computing, vol. 4, no. 3, pp. 321-334, July-Sept. 2016.

[20] D. Thatmann, S. Zickau, A. Förster and A. Küpper, "Applying Attribute-Based Encryption on Publish Subscribe Messaging Patterns for the Internet of Things," 2015 IEEE International Conference on Data Science and Data Intensive Systems, Sydney, NSW, 2015, pp. 556-563.

[21] U. Guin, P. Cui and A. Skjellum, "Ensuring Proof-of-Authenticity of IoT Edge Devices Using Blockchain Technology," 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 2018, pp. 1042-1049.

[22] Kirkpatrick, Michael S., Elisa Bertino and Sam Kerr. “PUF ROKs: generating read-once keys from physically unclonable functions.” CSIIRW (2010).

[23] M. Dammak, O. R. M. Boudia, M. A. Messous, S. M. Senouci and C. Gransart, "Token-Based Lightweight Authentication to Secure IoT Networks," 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 2019, pp. 1-4.

[24] G. E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," 2007 44th ACM/IEEE Design Automation Conference, San Diego, CA, 2007, pp. 9-14.

[25] M. Barbareschi, P. Bagnasco and A. Mazzeo, "Authenticating IoT Devices with Physically Unclonable Functions Models," 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), Krakow, 2015, pp. 563-567.