



**University
of Windsor**

**Master Of Applied Computing
Networking And Data Security
Project Report**

**Submitted To
Dr. Shaoquan Jiang**

Submitted By

Section 1, Group 5 (Friday)

Venkata Rahul Nittala 110094777

Venkata Naveen Varma Vegesna 110090483

Section 2, Group 9 (Tuesday)

Adeel Ahmed 110091296

Ishmeet Singh 110093296

1. Introduction

The research paper discusses a new technology that is growing enormously in the area of wireless communication which is known as Internet of Things (IoT) as it completes different tasks by connecting between the things which we use in daily lives. We are able to check and control different types of devices in the network by gathering different data like temperature, pressure, etc using these devices. This method is applied in several domains of smart building, environmental monitoring, smart city, healthcare monitoring system, and Smart parking in the context of these IoT benefits. In various countries today, IoT devices are emerging, and the number of IoT devices is fast expanding.

Because of the vast range of applications, elements in IoT communicate by broadcasting messages, which efficiently create the messages' broadcasting. This form of IoT network is vulnerable to attacks, with attackers attempting to disrupt networks. It will be easier for attackers to disrupt, fabricate, or even steal data in networks that may include great-secret or private confidential information. Personal and industrial information will suffer significant loss as a result of an IoT network assault. According to the research paper [1], the new Internet of Things structure revealed many network threats such as selective forwarding, DoS attacks, transmission of fraudulent routing-based data, and spying. DOS attacks are considered to be the most common types of network attacks as mentioned in [2]. Because of the many risks of network attacks, studying security issues in IoT organizations is critical.

As a result, the primary source of concern surrounding these hazards is the need for safety. One of the most serious issues is the disclosure of private information, which demonstrates the dangers of confidentiality. When data and identity are stolen, integrity problems arise. Attacks on integrity have the potential to limit sensing and control information. Another excessive target for the attacks is availability. This research paper mostly focuses on denial-of-service (DoS) attacks, which have a high probability of occurring when it comes to availability. DoS attacks occur when a required system or service cannot be retrieved. As a result, research is being performed into safeguarding broadcast communication protocols. This study presents an IoT security architecture based on a lightweight cryptographic method known as the Tiny Encryption method.

2. DoS Attacks

The author refers to a research paper [3] to support the definition of DoS attack which is: A DoS attack can be used to disrupt a network connection, rendering it unreachable to its users. It is used to flood the objective with a constant rise in traffic or to deliver information that causes the crash.

The different types of targets for these DoS attacks are servers of big organizations like banking sector, e-commerce sectors. Hence these attacks are widely spread over networks and focus on security of the sensor networks. This research paper mentions different types of DoS attacks like SYN Flood, Internet Control Message Protocol Flooding, Teardrop attacks, Peer-to-Peer attacks, Low rate DoS attack.

In SYN flood, an attacker sends a large number of SYN requests to the system to be targeted, attempting to utilize vast amounts of server resources to render the system impervious to legitimate traffic. This attack targets the three-way handshake process of TCP to overwhelm a target server's resources.

In Internet Control Message Protocol Flooding, we transmit an excessive number of ICMP packets to the target in an attempt to process each incoming ICMP request, resulting in a denial-of-service scenario. It is used in a connectionless protocol for IP diagnostics, faults, and operations.

In Teardrop attack, a hacker is involved who sends huge IP pieces that are unfinished, unorganized, and broken, as well as overlapping payloads, to the target's workstation. This will cause the servers to fail due to a problem in the reassembly of communicating channel fragmentation.

In a Peer-to-Peer attack, each device is known as a peer. This device has the ability to act as both client and server which shares resources and information directly with other devices without a need for a central server.

In a Low rate DoS attack, the focus is to exploit vulnerabilities in the target system to cause disruption. Hence it aims to disrupt the availability of the target system by sending a relatively low volume of malicious packets unlike the traditional DoS which sends a massive amount of traffic.

3. IoT Security

The devices used in IoT often have limited processing power, memory and energy resources. Hence by implementing the traditional cryptography algorithms it becomes very difficult to communicate between devices and becomes impractical. Also the computation becomes very heavy which results in load imbalancing. Therefore preferring the lightweight cryptography algorithms is the optimal solution to provide security that is efficient and suitable for these resource-limited environments. These lightweight algorithms focus on the key requirements like Low Computation Overhead, Low Memory Usage, Energy Efficiency, Fast Execution.

At the moment, there are a few selected basic cryptographic algorithms that do not always manipulate security and its efficiency interchanges. Among hash functions, block-based ciphers, and stream-based ciphers, block-based ciphers perform much better. For most encryption needs, modern block ciphers are highly efficient and secure. Stream ciphers find their niche in real time communication scenarios, while hash functions are critical for data integrity verifications.

In a research paper [4], mCrypton as a block cipher is suggested which supports the key sizes of 8 bytes, 12 bytes and 16 bytes. In research paper [5], Crypton is trailing this method design through functionality minimization to improve its performance for the hardware. In research paper [6], an algorithm named Hummingbird-2(HB-2) is presented as a heir to hummingbird-1 which is mentioned in research paper [7], in which 8 bytes of initialization vector and 12 bytes of key are tested to remain unaffected by all previous discovered attacks. In research paper [8], the cryptanalysis of HB-2 also highlights the algorithm's flaws, and the initial key can be updated as a result.

The author mentioned that in the research papers [9][10][11], the investigation was done for energy usage of different algorithms including RC4, RC5, IDEA. Even the cost of these algorithms in various platforms were considered. But the TEA algorithm was not mentioned anywhere and was ignored throughout the analysis.

This related study provides very good information on what was implemented earlier and why only the TEA algorithm is the main focus in this paper which supports the idea behind the work.

4. IoT Security Architecture of DoS

IoT Sensor Board

It is a sensor network node that is capable of performing some processing. This node's job is to collect sensor information and connect with other sensor nodes in the sensor network. Sensors on IoT sensor nodes attempt to collect data from their surroundings. Sensor nodes are hardware devices that produce measurable changes in physical conditions such as temperature, pressure, and so on. By detecting and monitoring physical data, the sensor node acquired certain features such as accuracy, sensitivity, and so on.

IoT Board

It is a board with low-power CPUs that support a variety of programming techniques. The sensor's data is transported to the cloud server, where it is collected by various technologies such as Wi-Fi, Ethernet, and so on. There are numerous IoT capable hardware prototype alternatives available, including the Raspberry Pi, Beagle Board, Arduino Uno, Photon, and more.

Network Gateway

The network gateway connects the cloud server to the IoT devices. It might be a physical item or a software program. All data going to and from the cloud passes through the gateway. The data generated by IoT devices is preprocessed locally at network devices by the gateway before being sent to the cloud server. This device manages information traveling in both directions, and it can prevent data leaking when data is sent to the cloud and IoT devices from being compromised by hostile assaults through change detection and encryption.

Cloud Server

It is similar to a virtual server that users access via an internet based cloud computing platform. I have real abilities and can grasp and represent itself as a real server. It can be obtained via cloud service provider. It also provides flexibility, scalability and cost efficiency compared to traditional physical servers.

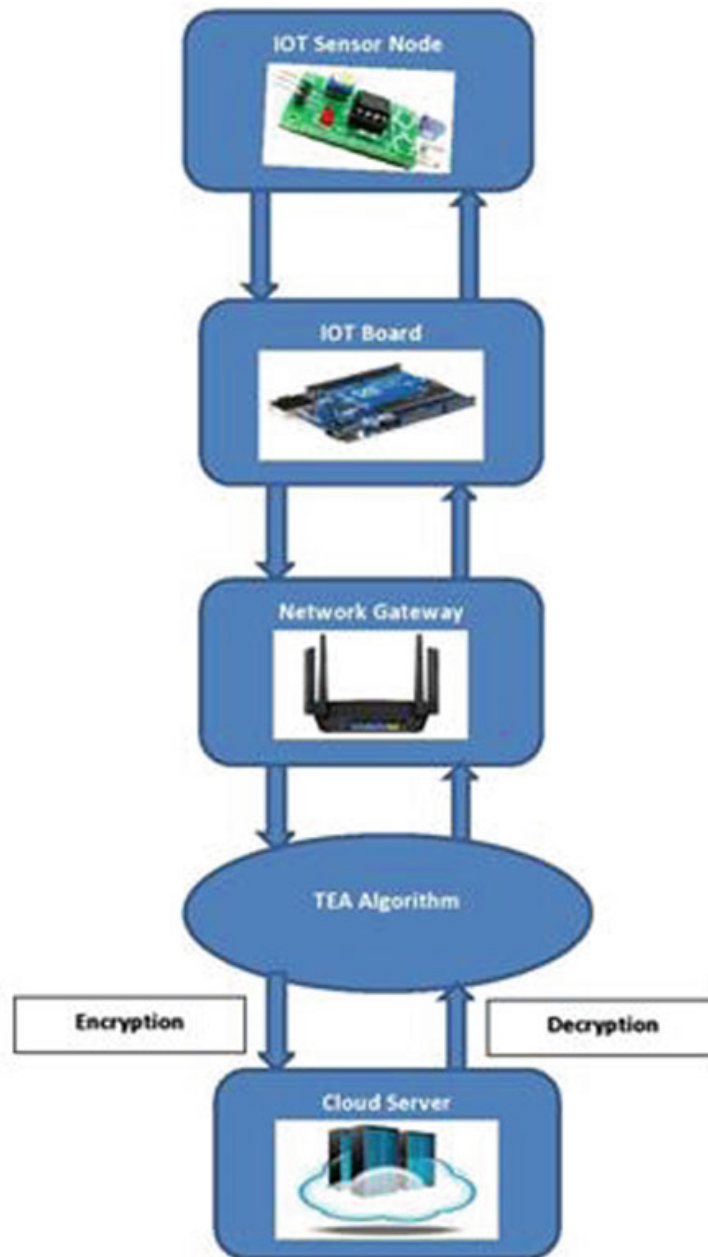


Fig-1 TEA implementation on IoT '[12]'

As shown in Fig-1, To implement the TEA algorithm above mentioned are the basic required parts which are needed for complete understanding of the functionality of the algorithm and mentions how the proposed model of this research paper works theoretically. These give clear understanding on how the proposed model works and can be easily understood how it can be implemented.

5. TEA - Tiny Encryption Algorithm

David Wheeler and Roger Needham created the block cipher known as the Tiny Encryption Algorithm (TEA). It uses a 128-bit key and two 32-bit unsigned integers as its data. The algorithm has 64 cycles and employs a Feistel structure. The author gives a detailed explanation of the methodology of algorithm on how it works and how it has been implemented and how to encrypt and decrypt the messages which gives us good insight of the implementation and makes our work easy for our practical approach.

5.1 TEA Methodology

A straightforward and effective technique is used to carry out TEA encryption. The 128-bit key is divided into four 32-bit blocks, designated as E[0], E[1], E[2], and E[3] by the key scheduling algorithm. It differs from traditional Feistel ciphers in that the encryption method uses bitwise operations and addition modulo 232 to combine the data and key.

Key Scheduling Algorithm

- Four 32-bit blocks, designated as E[0], E[1], E[2], and E[3], make up the 128-bit key.
- The golden number ratio $(5-1)*2^{31}$ is used to calculate the delta constant, which is set to 0x9e3779b9.
- E[0] and E[1] are utilized during odd cycles of encryption, while E[2] and E[3] are utilized during even cycles.
- The subkeys are used in reverse order during decryption.

Encryption Algorithm

```
void Encrypt(long* data, long* key) {  
    unsigned long delta = 0x9e3779b9; // Key schedule constant  
    unsigned long sum = 0;  
    unsigned long left = data[0];  
    unsigned long right = data[1];  
  
    for (int i = 0; i < 32; i++) {  
        sum += delta;
```

```

    left += ((right << 4) + key[0]) ^ (right + sum) ^ ((right >> 5) +
key[1]);
    right += ((left << 4) + key[2]) ^ (left + sum) ^ ((left >> 5) +
key[3]);
}

data[0] = left;
data[1] = right;
}

```

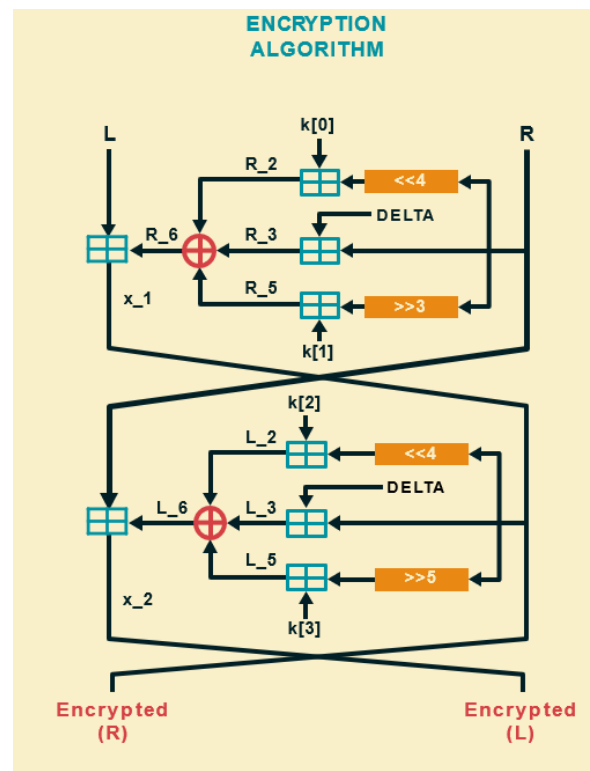


Fig-2 TEA Encryption Flowchart

From Fig-2, 'left' and 'right' are the two 32-bit halves that make up the data block.

The following actions are included in each encryption cycle:

The left half is added after adding the key value "key[0]" to the right half, followed by a bitwise XOR operation with the sum and the right half moved by 5 bits.

Add the key value "key[2]" to the left half, then do a bitwise XOR operation using the sum and the leftmost 4 bits of the result, and then add the outcome to the right half.

Before moving on to the following cycle, increase the sum by the delta constant.

Decryption Algorithm

```
void Decrypt(long* data, long* key) {  
    unsigned long delta = 0x9e3779b9; // Key schedule constant  
    unsigned long sum = delta << 5;  
    unsigned long left = data[0];  
    unsigned long right = data[1];  
  
    for (int i = 0; i < 32; i++) {  
        right -= ((left << 4) + key[2]) ^ (left + sum) ^ ((left >> 5) +  
key[3]);  
        left -= ((right << 4) + key[0]) ^ (right + sum) ^ ((right >> 5) +  
key[1]);  
        sum -= delta;  
    }  
  
    data[0] = left;  
    data[1] = right;  
}
```

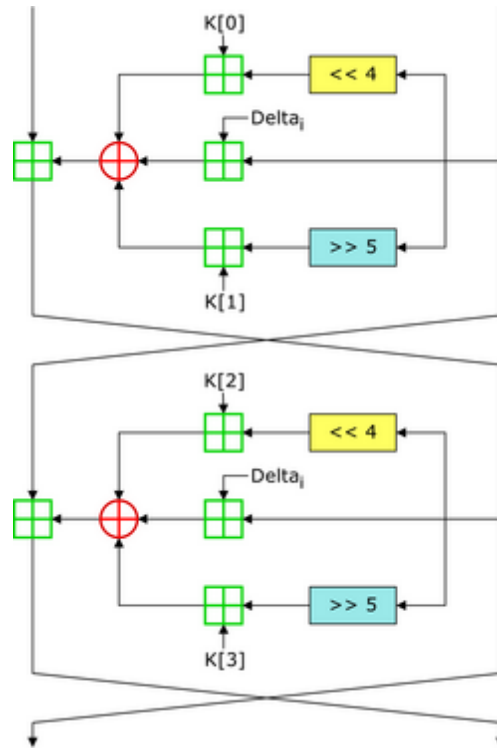


Fig-3 TEA Decryption Flowchart

Although the subkeys are used in reverse order during decryption, the technique is similar to encryption.

Decryption cycles work similarly to encryption cycles, except they reverse the processes to recover the original data.

TEA offers a quick and effective encryption approach that can be used in a variety of situations where just light cryptography is needed. To maintain the TEA encryption's security, it is crucial to make sure that the key is kept secure and difficult to predict.

5.2 Implementation

The working mechanism of the proposed solution involves using a Raspberry Pi Pico microcontroller with a temperature and humidity sensor to collect data. The collected data is then encrypted using the TEA algorithm, which is implemented in the microcontroller itself. Once encrypted, the data is sent to the web server through a HTTP webhook. The webhook ensures that the encrypted data is stored on a cloud sheet with a timestamp, making it easier to track and monitor.

On the web server side, the latest record from the cloud sheet is read and decrypted using the same TEA algorithm, which ensures that the data remains secure throughout the transmission process. This same mechanism works vice versa as well, where the web server can send encrypted data to the Raspberry Pi Pico through the HTTP webhook, and the microcontroller will decrypt it using the TEA algorithm.

The project successfully implemented an E2E encrypted communication channel for IoT devices and web servers using the TEA algorithm. The performance evaluation of the TEA algorithm showed that it has a fast encryption and decryption time and low resource consumption. The project's comparison of the TEA algorithm with other lightweight encryption algorithms used in IoT-based applications showed that the TEA algorithm outperforms other algorithms in terms of encryption and decryption time and resource consumption.

5.3 Flowchart:

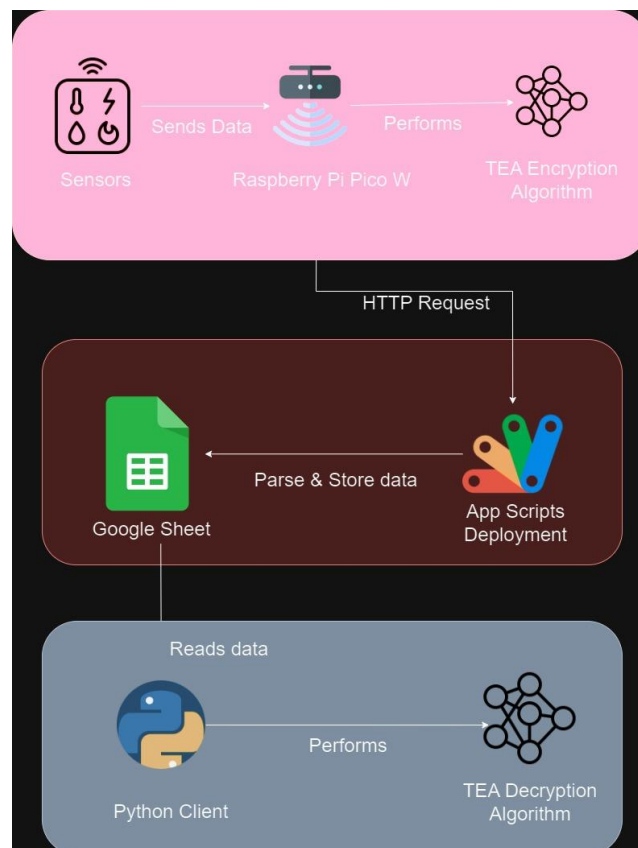


Fig-4 Implementation WorkFlow

5.4 Output:

```
adeel@Adeel:~/pprograms/pns-project/e2e-iot-encryption-system$ python e2e_encryption_system.py

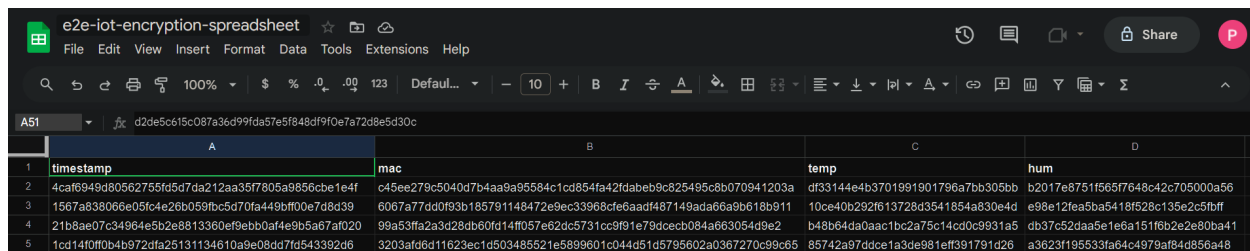
Encrypted data fetched from cloud: ['d2de5c615c087a36d99fda57e5f848df9f0e7a72d8e5d30c', 'd6d6dd30edc189885a629d71afe19679199396157c063fdbbd95fc366af11f7f', '1a6cd5f794f78825896beaef44f84a63', '112a3ad7bd397129e00cda4b883c04e']
Decrypted data: ['1691447479', '28:cd:c1:07:13:82', '24.3', '60.3']
```

Fig-5 Console Output of Python Client

```
Shell
>>> %Run -c $EDITOR_CONTENT
Connected to Wi-Fi! IP: 172.20.10.11
Plain payload: {'macAddress': '28:cd:c1:07:13:82', 'temperature': '25.0', 'humidity': '63.9', 'timestamp': '1691624470'}

Encrypted payload: {'macAddress': '2facf2c58e9647753d2cab90f9dd23cbbaabafcd09aad27c9695b502e48214', 'temperature': '6619b0675132a4e1cb8370bad5557e7c', 'humidity': 'fba6c24d01c7a020be71ba9e664ff61e', 'timestamp': '57c90c102c0e01ff80e43dccc3339de6215143175aabf'}
Response status code: 200
```

Fig-6 Console Output of Raspberry Pi Pico W



	A	B	C	D
1	timestamp	mac	temp	hum
2	4caf6949d80562755fd5d7da212aa35f7805a9856cbe1e4f	c45ee279c5040d7b4aa9a95584c1cd854fa42fdabeb8c825495c8b070941203a	d33144e4b3701991901796a7bb305bb	b2017e8751f565f7648c42c705000a56
3	1567a83806e05fc4e26b059fbc5d70fa449b7f00e7d8d39	6067a77dd0f93b185791148472e9ec33968cfe6aadf487149ada66a9b618b911	10ce40b292f613728d3541854a830e4d	e98e12fa5ba5418f528c135e2c5fbff
4	21b8ae07c34964e5b2e8813360ef9ebb0af4e9b5a67af020	99a53ffa2a3d28db0fd14f057e62dc5731cc9f91e79dcecb084a663054d9e2	b48b64da0aac1bc2a75c14cd0c9931a5	db37c52daa5e1e6a151f6b2e2e80ba41
5	1cd14f0f0b4b5972dfa25131134610a9e08dd7fd543392d6	3203afd6d11623ec1d503485521e5899601c044d51d5795602a0367270c99c65	85742a97ddce1a3de981eff391791d26	a3623f195533fa64c4979af84d856a48

Fig-7 Encrypted Data Storage

6. Conclusion

The Internet of Things will be a crucial component of our daily lives in the approaching era. Various sensors of IoT devices are constantly talking with one another while using limited energy. The security of the communication cannot be bargained for. As a result of this security motivation, this work proposes a lightweight algorithm TEA with a simple and integrated countermeasure against DoS in IoT. Security processes and controls ensure the confidentiality, integrity, and availability of the information processed. However, several DoS attack approaches exist, such as attack tools, application-layer floods, service degradation, and service denial.

7. References

1. Alanazi S, Al-Muhtadi J, Derhab A, Saleem K (2015) On resilience of Wireless Mesh routing protocol against DoS attacks in IoT-based ambient assisted living applications. In: ICENAS
2. Alanazi S, Al-Muhtadi J, Derhab A, Saleem K (2015) On resilience of Wireless Mesh routing protocol against DoS attacks in IoT-based ambient assisted living applications. In: ICENAS
3. Atzori L, Iera A, Morabito G (2010) The internet of things: a survey. *Comp Networks* 54: 2787– 2805
4. Lim CH, Korkishko T (2005) mcrypton—a lightweight block cipher for security of low-cost RFID tags and sensors. *Info Sec App Springer*, pp 243–258
5. Lim CH (1998) Crypton: a new 128-bit block cipher. In: NIST AEs Prop.
6. Engels D, Fan X, Gong G, Hu H, Smith EM (2009) Ultralight weight cryptography for low-cost RFID tags: Hummingbird algorithm and protocol. *CACR Tech Rpts* 29
7. Engels D, Saarinen MJO, Schweitzer P, Smith EM (2011) The hummingbird-2 lightweight authenticated encryption algorithm. *RFID Sec. & Privacy*. Springer, pp 19–31
8. Zhang K, Ding L, Guan J (2012) Cryptanalysis of hummingbird-2. *IACR Crypt. ePrint Arc.*, vol 2012, p 207
9. Ganesan P, Venugopalan R, Peddabachagari P, Dean A, Mueller F, Sichitiu M (2003) Analyzing and modeling encryption overhead for sensor network nodes. In: *Proceedings of the 2nd ICWSNA*. ACM, pp 151–159
10. Schneier B (2007) *Applied cryptography: protocols, algorithms, and source code in C*. Wiley
11. Lai X (1992) On the design and security of block ciphers. Ph.D. diss., Diss. Techn. Wiss ETH Zürich, Nr. 9752, 1992. Ref.: Massey JL, Korref, uhlmann HB
12. Sharma, V., & Sharma, A. (2020, August 19). IoT Security Architecture with TEA for DoS Attacks Prevention. *Advances in Information Communication Technology and Computing*, 215–226.