Journal of Cybersecurity Education, Research and Practice

Volume 2022 | Number 2 Article 8

January 2023

Lightweight Pairwise Key Distribution Scheme for IoTs

Kanwalinderjit Kaur
California State University, Bakersfield, kgagnej@csub.edu

Lightweight Pairwise Key Distribution Scheme for IoTs

Abstract

Embedding a pairwise key distribution approach in IoT systems is challenging as IoT devices have limited resources, such as memory, processing power, and battery life. This paper presents a secure and lightweight approach that is applied to IoT devices that are divided into Voronoi clusters. This proposed algorithm comprises XOR and concatenation operations for interactive authentication between the server and the IoT devices. Predominantly, the authentication is carried out by the server. It is observed that the algorithm is resilient against man-in-the-middle attacks, forward secrecy, Denial of Service (DoS) attacks, and offers mutual authentication. It is also observed that the given scheme has low communication and computing overheads compared to some existing methods.

Keywords

IoTs, attacks, intruders, forward secrecy, DoS

Cover Page Footnote

We want to thank GRaSP for funding this research with project number: PJ0039

Lightweight Pairwise Key Distribution Scheme for IoTs

Kanwalinderjit Kaur
Department of Computer and Electrical Engineering and Computer Science
California State University
Bakersfield, USA
kgagnej@csub.edu

***Abstract*— Embedding a pairwise key distribution approach in IoT systems is challenging as IoT devices have limited resources, such as memory, processing power, and battery life. This paper presents a secure and lightweight approach for IoT devices that are divided into Voronoi clusters. This proposed algorithm comprises XOR and concatenation operations for interactive authentication between the server and the IoT devices. Predominantly, the authentication is carried out by the server. It is observed that the algorithm is resilient against man-in-the-middle attacks, forward secrecy, Denial of Service (DoS) attacks, and offers mutual authentication. It is also observed that the given scheme has low communication and computing overheads compared to some existing methods.**

Keywords—IoTs, attacks, intruders, forward secrecy, DoS

I. INTRODUCTION

Nowadays, Internet of Things (IoT) devices are used by many to collect data for various purposes such as health, environment, industrial control, weather, home appliances, thermostats, etc. IoT is the fastest growing area, where the number of IoT devices has already surpassed the number of human beings on this earth. In terms of their availability and cost, most of the population can access them and use them in their day-to-day life. Such an advancement and a range of such IoT devices introduce various challenges associated with security.

IoT devices are used for a wide range of applications. Such applications may also be considering other variables such as energy efficiency, data analysis of data gathered by IoTs, security, availability, privacy, and interoperability with the given application [1], [21]. The integration of IoT devices in various systems provides numerous opportunities for interdisciplinary areas of researchers to work on the challenges that such integration provides. The distributed nature of these integrated systems also presents a huge, vulnerable surface for intruders. Hence, it raises various security issues due to a variety of attributes of IoTs'. Additionally, IoT devices are bound to generate voluminous data, so securely analyzing and transmitting it is another challenge.

It is understandable that systems using IoT devices are convoluted and require integrating multiple tools, devices, networking arrangements, transmitters, etc. Moreover, IoT devices usually operate in an unattended atmosphere. As a result, an attacker may possibly gain physical access to the devices or even gather data sent by these devices over communication channels. Furthermore, IoT devices have limited resources, such as memory, energy, and processing power [2], which calls for greater security requirements. The solutions to this require a holistic approach to meet the security requirements. Obviously, the IoT security structure is intricate not only because of limited resources it also involves trustworthy interaction with the cyber-physical system. This gives rise to another domain where IoT devices should adapt to the changing needs as and when they arise.

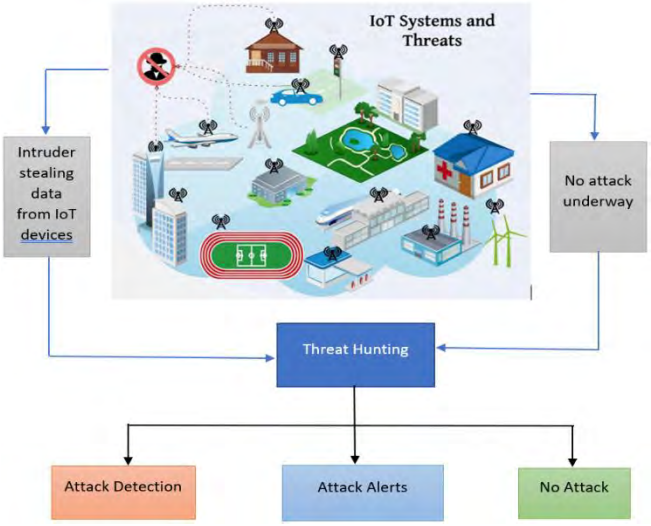

Figure 1: Threat Hunting in action in an IoT system

Basically, IoT devices are accessed universally, and some of the devices may have known vulnerabilities. When there are multiple devices connected to form a complete system, then this system could be secured by installing access control and authentication, along with computational encryption, and by applying network and application security at various levels of the system. However, when there are vulnerabilities in the connected devices, it becomes easier for the attackers to compromise the system. Recently, 'Mirai' botnets triggered Distributed Denial of Service attacks because of known vulnerabilities in the IoT devices [3], [4]. Since there are multiple types of IoT devices, their applications, and various scenarios in which they are used, instead of just adding layered security, the IoT devices should also be secured to save the complete system from being compromised.

Figure 1 presents a framework where threat hunting monitors the given IoT system and predicts any attack that

may occur, and can also come up with a solution to any fresh or zero-day exploits. Threat hunting is an approach to identify if some exploit could happen or if the operations are normal. The data is collected from every component of the IoT system to compute any possible threats. This helps in detecting mischievous acts at the initial moments.

II. IOTs Structural Design

Before applying security to IoT devices, it is better to understand their structural design. There are numerous Internet of Thing devices connected to each other in various ways, such as device-to-device, person-to-device, or person-to-person [5], [8]. The architecture of the IoTs is a collection of physical devices that are incorporated into a computational network of protocols to provide services to the end-users. The IoT architecture is a collection of various heterogeneous devices using various transmission approaches. There are three layers to IoT architecture, application layer, network layer, and perception layer [7]. The application layer directly deals with IoT devices and uses them to fulfill organizational goals with the partnership of other organizations and systems. The network layer deals with communication protocols, middleware and application programming interface, and threat hunting. The lowest layer is the perception layer. It deals with how IoT devices should be connected using various protocols and standards. Its structural design is represented in Figure 2.

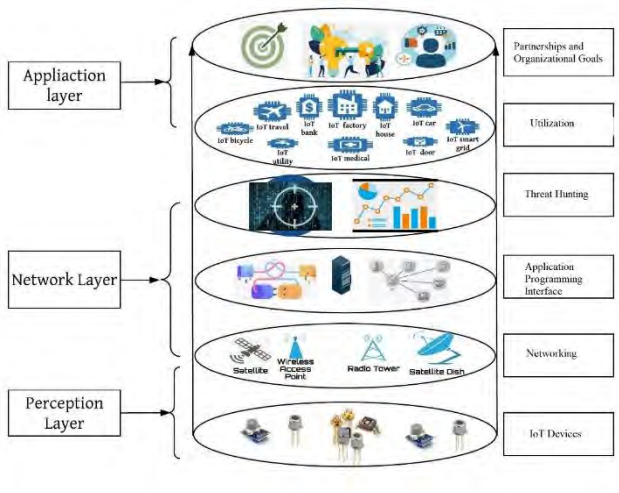

Figure 2 IoT Structural Design

Various components of this structural design are explained as follows:

A. IoT Devices

The lowest level of the perception layer includes physical sensors. The sensors collect data by sensing and processing the data to deliver information. The sensors are generally heterogeneous and could read the temperature, motion, humidity, etc. [8], [10], [11]. Such IoT devices are usually resource-constrained as they have limited battery power, memory, and computation capacity.

B. Networking

The IoT devices are connected using communication networks. Each device should be provided with a unique IP address. Since these devices are small as to require low power communication for transferring data. Another connectivity issue with such devices is an efficient routing algorithm, as these devices could be mobile and are usually memory and battery constrained. So, the reliable communication protocols used for IoT are NFC (Near Field Communication), Wi-Fi, Bluetooth, 6LoWPAN, IEEE 802.15.4, etc. [8].

C. Application Programming Interface

An application programming interface (API) works among the operating system, the applications, and the network protocols for the coordinated functioning of various IoTs [12], [14], [15]. So, API coordinates the interaction among various IoT devices with different communication standards, memory requirements, and processing needs. The interface handles the scalability associated with changing needs. It also provides security to the transmitted data. Additionally, it supports context-aware computing for sensors to be aware of other devices' contexts.

D. Threat Hunting

Threat hunting is considered a proactive technique for examining various threats that are posed to an organization's internal network, whereas threat hunters consider that malicious actors are already in your environment, and they try to find the source of malicious activities that signifies that there is some threat. In any organization, the IoT devices generate vast amounts of data, which should be processed immediately for useful information and that information could be beneficial to the adversary. So, it is extremely important to set up a lightweight pairwise key distribution scheme for IoTs.

E. Utilization

There are a number of places where IoT devices are in use, such as healthcare systems, smart homes, intelligent transportation systems, smart cities, and smart grids. It is quite obvious that the IoT device is more vulnerable to exploits for the following reasons:

- IoT systems are complex in structure, and they work differently for different applications. The successful security approach applied for one application may not be appropriate for another application.
- The communication features used in IoT devices are not standardized. Lack of standardization is the main obstruction in the development of a functional security approach.
- Mostly, IoT devices are controlled by apps or other devices. Consequently, compromising them is easier.
- The IoT devices produce a lot of data for the related application. Because of the lack of end-to-end security, this data ought to be breached.
- Natural disasters, such as floods, earthquakes, wars, etc., can cause physical damage to the devices.

Therefore, the threat hunting approach should be applied to predict the unassertive and dynamic threats to significantly improve the security of IoT devices. Aman *et al.*, in their paper "Mutual authentication in IoT systems using physical unclonable functions," proposed an authentication scheme for IoTs when they set up a connection with the server using physically unclonable functions (PUF) [16]. Chatterjee *et al.*,

in their paper "A PUF- based secure communication protocol for IoT," proposed physically unclonable functions used by IoTs for authentication and key exchange [19]. Braeken, in his paper "PUF based authentication protocol for IoT," proposed an algorithm that establishes trust amongst the users of IoT devices that are unfamiliar with each other [20].

III. Threats to IOTs

It is important to learn what types of threats are posed to IoTs so an effective and efficient security algorithm could be designed. When vulnerabilities of a system are exploited to gather information for financial gains is known as a threat. As shown in fig. 3 there could be unassertive threats or dynamic threats to the IoT systems [5]. Unassertive threats usually are an attack on confidentiality, where an intruder can install a keylogger or can capture packets in an IoT system. Dynamic threats are attacks that threaten Confidentiality, Integrity, and Availability (CIA) tirade. It also threatens authorization, availability, and non-repudiation. An example of an attack on integrity is identity theft and information extortion. Examples of attacks on authentication are credential stuffing and passwords not hashed properly. There are destroying and manipulation of data attacks related to authorization. Denial of service and buffer overflow attacks are a threat to availability. After a careful review of threats to IoT systems and their limited resources, it's important to set up a lightweight pairwise key distribution scheme for IoTs.

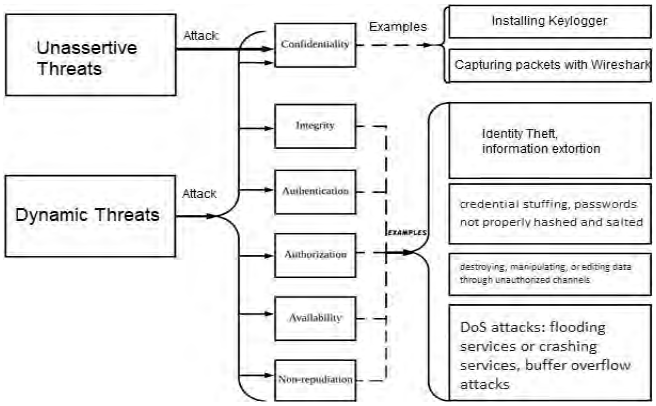

Figure 3: Possible threats with respect to Confidentiality, Integrity, and Availability (CIA) tirades against IoTs

IV. Pairwise Key Distribution Algorithm

It is known that in any IoT system, there are a variety of IoT devices and these IoT devices have limited memory, battery life, and processing power [6], [9], [13]. As discussed in section III, IoT devices are vulnerable to various attacks [17], [18]. We propose a pairwise key distribution scheme for heterogeneous IoT devices in a network to securely transmit the data. The IoT devices are initially partitioned into Voronoi clusters. The Voronoi partitioning makes use of Euclidean distance to make clusters of IoT devices in a 2D plane. The following fig. 4 presents Voronoi clusters in a 2D plane. The clusters made using the Voronoi algorithm are constructed by a set of vertical bisectors among the pairs of various cluster heads.

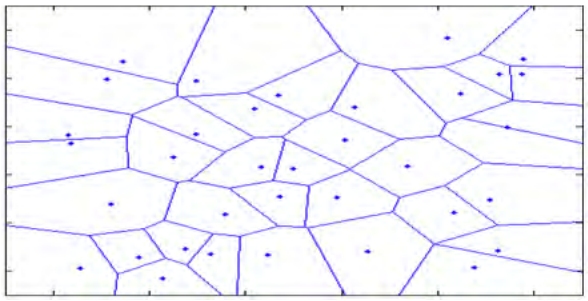

Figure 4: Voronoi clusters in a 2D plane

We worked with the acceptability model to ensure that the area under consideration is completely covered. The following presents the optimality of the given design:

T ={1,…,n}, the set of IoTs

C = {1,…,.m}, the set of clusters

i= 1..n indexes for the IoTs

j= 1..p indexes for the clusters

X_i are the IoTs in the given network.

C_i are the clusters in the given network

E_n is the used energy at any time by the given network

E_r is the remaining energy at any time for the given network

t_{ni} is the total energy of the given network=log $(1+e_n / e_r)$

t_{ri} is the remaining energy

Minimize:

$$\sum_{i=1}^{m} Ci.\log\left(1+\frac{Eni}{Eri}\right)+\sum_{i=1}^{n} Xi.\log\left(1+\frac{Pxi}{Eri}\right) \quad \text{eq(1)}$$

Subject to:

$$\sum_{i=1}^{n}\left(\sum_{j=1,m<n}^{m}(Ci \le Xi)\right) \quad \text{eq(2)}$$

$$\sum_{j=1}^{m}(Xij \le 1)) \quad \text{eq(3)}$$

Where constraint 1 (eq(2)) ensures that the number of clusters is fewer than IoTs. And constraint 2 (eq(3))ensures that every IoT device (X_i) is appropriately covered in each cluster.

This paper presents a pairwise key distribution algorithm that is secure and lightweight for IoT systems. This proposed algorithm comprises XOR operations and concatenation for interactive authentication between the server and the IoT devices. Predominantly, the authentication is carried out by the server. It is observed that the algorithm is resilient against man-in-the-middle attacks, impersonation, and forward secrecy attacks.

Once the IoT devices are partitioned into Voronoi clusters, then the keys are established. The IoT devices use their address plus the cluster number as their keys. The IoTs find out their shared keys through the distributed scheme as each IoT device knows the address of all other IoT devices in one cluster and the cluster number assigned to their cluster by

the server. Hence, the shared key $A_{c,p,q}$ is shared by two IoT devices *p, q,* and *c* is the cluster number assigned by the server.

In many security schemes, IoT devices keep many keys to pick from. This process requires memory, processing power, and battery energy. Since IoT devices are memory, energy, and computation constrained, the security scheme should not consume much power, memory, and energy.

When the shared keys are set up, the two IoT devices, *p,* and *q* carry out a 2-way handshake, where *p* may send a nonce to *q*: $\{p\}A_{c,q,p} + MAC(A_{c,q,p}, *)$, with $MAC(A_{c,q,p}, *)$ as a Message Authentication Code produced by network layer with the key $A_{c,q,p}$. Upon receiving the key *q* responds to *p*: $\{B_{q,p}, q\}\ A_{c,q,p} + MAC(B_{q,p}, *)$. $B_{q,p}$ is a shared key produced by *q* to use for future data communication between *p* and *q*.

Each packet has two parts, header, and data. The header part is comprised of unique packet identification (*up-id*), associated MAC ($MAC(A_{c,q,p}, *)$), event time (*et*), the type of the packet (*typep*), and the size of the packet (*sizep*). The *up-id* is used to track the packet for its route. The MAC part keeps track of any modification done to the packet from the time when it was generated. The following table I presents the list of parameters used.

Table I Parameter List

α	IoT devices requesting a shared key from the server
W_t	Time spent waiting
D_{ID}	Address of IoT device
R_r	Radio range of any IoT device
et	Event time
$Cluster_{ID}$	Cluster Identification number

When the system runs for the first time, then each IoT device configures by executing SetUpIoTs function. When the IoTs are within the radio range of each other, then they listen to the server for any communication for a randomly selected time (lines 2-4). When an IoT receives a nonce, then it generates a shared key (lines 5- 8). The IoT verified if the key received is a shared key or not (line 12). If there is a match, keys are shared and appended to the existing all-keys (lines 13-14).

```
1: function SetUpIoTs(α, Wt, Rt)
2:  time=random(0<Wt) //random function is used to compute waiting time
3:  do
4:    listen to the server
5:    if Talk= nonce(DID, Rr) at et
6:    Shared-Key= DID⊕ClusterID
7:    Request (Shared-Key) //request for a key
8:    end if
9:    end while(time> 0)&&(R ≤ Rr)
10:  Reply(nonce(DID, Rr)) //reply to the sending IoT device
11:  do
12:    if (KeyReceived) ==Shared-Key
13:          Key-is-Shared // the key is shared
14:          all-keys = all-keys ∪ Key-is-Shared
15:          send(DID, MAC, Key-is-Shared)
16:          end if
17:  end while ((α-1)> all-keys)
18: end function
```

This scheme has the following concerns. Initially, there could be leakage of data, and it's not able to detect compromised nodes. The IoTs are not temper resilient, so a DoS attack could take place and could reveal the keys.

V. Evaluation

This section presents various security features that are supported by the proposed algorithm to mitigate various attacks. The given algorithm is compared with the already existing protocols physically unclonable functions (PUF) [16] proposed by Aman et al., PUF-based secure communication protocol for IoT [19] proposed by Chatterjee et al., and PUF-based authentication protocol for IoT [20] proposed by Braeken.

A. Feature Comparison

The following features are compared with the existing above-mentioned techniques:

1) Man In The Middle Attack

In the man-in-the-middle attack, an adversary may attempt to fool the IoT device or the server or both of them by inducing his own communication after hearing what they both are talking about. The presented scheme can avoid the man-in-the-middle attack since it requires both participating parties to mutually authenticate using their unique verification codes. One can argue that an intruder can spoof an admissible entry if he knows the shared keys of the device. However, meddling physically with such IoT devices to get the keys will be useless.

2) Denial of Service (DoS) Attack

During the data exchange, the IoT device and the server both verify the event time (*et*). The intruder may replay the previous data. The IoT device and the server would discard this data since the transmitted data is encoded with the current event time and other integrity checks. Therefore, the proposed approach is capable of identifying fake data to avoid DoS attacks by disconnecting from unauthorized users.

3) Forward Secrecy

The proposed scheme maintains the forward secrecy by making sure the session keys for the previous sessions are not compromised. Now consider the private key of the IoT device gets compromised for some reason. However, the forward secrecy is maintained by the previously computed event time and random waiting time.

4) Mutual Authentication

The given scheme validates mutual authentication for the IoT device and server since it requires a 2-way handshake and intermediate authentication keys to secure the transmission. Such keys can be computed by authorized devices providing legitimate data and shared keys.

Table II: Feature comparison

Features	Chatterjee et al. [16]	Aman et al. [19]	Braeken [20]	Given
Man in the middle attack	No	Yes	Yes	Yes
DoS attack	No	No	Yes	Yes

Forward Secrecy	_	_	No	Yes
Mutual Authentication	Yes	Yes	Yes	Yes

Table II demonstrates the feature comparison of the given approach with the existing approaches. The feature supported by Chatterjee's approach is only mutual authentication. Noting Aman's protocol features can only provide protection for mutual authentication and man-in-the-middle attacks. At the same time, Braeken's approach maintains high levels of security. However, it does not provide for forward secrecy. Though, the proposed algorithm provides security for all the listed features.

B. Overhead Analysis

1) Communication Overhead

To calculate the communication overhead, one should know how many bytes of data are sent and received by the IoT device for various phases of the algorithm. Table III presents the data in bytes for various approaches in consideration. It can be inferred from the data that the given approach has the lowest communication overheads.

Table III: Communication Overheads

Approach	Chatterjee et al. [16]	Aman et al. [19]	Braeken [20]	Given
Data sent in bytes	131	209	102	110
Data received in bytes	112	122	189	118
Total bytes	243	331	291	228

2) Computing Overheads

The computing overheads are calculated based on the codes of various approaches implemented on a PC. The time required to run initial authentication, MAC, and encryption and decryption is 1.986, 0.142, and 12.293 ms. The following Table IV presents the computing overhead for Chatterjees' IoT device and server are approximately 15 and 56ms. The computing overhead for Amans' IoT device and server are around 22 and 31ms. The computing overhead for Braekens' IoT device and the server is approximately 2 and 3ms. The computing overhead for a given IoT device and server is about 3 and 6ms.

Table IV: Computing Overheads

Approach	Chatterjee et al. [16]	Aman et al. [19]	Braeken [20]	Given
IoT device	15 ms	22 ms	2 ms	3 ms
Server	56 ms	31 ms	3 ms	6 ms

VI. CONCLUSIONS

The IoT devices are resource crunched as they have limited memory, battery life, and processing power. However, securing data communication over IoT devices is important as there are various attacks on such devices. Initially, the IoT system is divided into Voronoi clusters. Then a pairwise key distribution approach is applied to IoTs, which is secure and lightweight. This proposed algorithm comprises XOR and concatenation operations for interactive authentication between the server and the IoT devices. Mainly, the authentication is carried out by the server. After careful comparison with the existing approaches, it is noted that the proposed approach is better prepared to fight against man-in-the-middle and Denial of Service (DoS) attacks. The Server and IoT devices work with each other through mutual authentication, so it also offers forward secrecy. The comparison with existing schemes also presents the fact that the given method has low communication and computing overheads.

REFERENCES

[1] Ray S., Jin Y., and Raychowdhury A., "The changing computing paradigm with internet of things: A tutorial introduction," IEEE Design & Test, vol. 33, no. 2, pp. 76-96, 2016.

[2] Abomhara M., "Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks," Journal of Cyber Security and Mobility, vol. 4, no. 1, pp. 65-88, 2015.

[3] Bertino E. and Islam N., "Botnets and internet of things security," Computer, vol. 50, no. 2, pp. 76-79, 2017.

[4] Kolias C., Kambourakis G., Stavrou A., and Voas J., "DDoS in the IoT: Mirai and other botnets," Computer, vol. 50, no. 7, pp. 80-84, 2017.

[5] Alaba F. A., Othman M., Hashem I. A. T, and Alotaibi F., "Internet of Things security: A survey," Journal of Network and Computer Applications, vol. 88, pp. 10-28, 2017.

[6] Gagneja K.K. and Nygard K., "Heuristic Clustering with Secured Routing in Heterogeneous Sensor Networks", IEEE SECON, New Orleans, USA, pages 51-58, June 24-26, 2013.

[7] Zhao K. and Ge L., "A survey on the internet of things security," in Computational Intelligence and Security (CIS), IEEE 9th International Conference on, 2013, pp. 663-667.

[8] Al-Fuqaha A., Guizani M., Mohammadi M., Aledhari M., and Ayyash M., "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347-2376, 2015.

[9] Morteza Safaei Pour, Antonio Mangino, Kurt Friday, Matthias Rathbun, Elias Bou-Harb, Farkhund Iqbal, Sagar Samtani, Jorge Crichigno, Nasir Ghani, On data-driven curation, learning, and analysis for inferring evolving internet-of-Things (IoT) botnets in the wild, Computers & Security, Volume 91, 2020.

[10] Yang Z., Yue Y., Yang Y., Peng Y., Wang X., and Liu W., "Study and application on the architecture and key technologies for IoT," in Multimedia Technology (ICMT), IEEE International Conference on, 2011, pp. 747-751.

[11] Wu M., Lu T. J., Ling F. U., Sun J., and Du H. Y., "Research on the architecture of Internet of things," in Advanced Computer Theory and Engineering (ICACTE), IEEE 3rd International Conference on, 2010, vol. 5, pp. V5- 484-V5-487.

[12] Sethi P. and Sarangi S. R., "Internet of things: architectures, protocols, and applications," Journal of Electrical and Computer Engineering, vol. 2017, 2017.

[13] Kim-Kwang Raymond Choo, Keke Gai, Luca Chiaraviglio, Qing Yang, A multidisciplinary approach to Internet of Things (IoT) cybersecurity and risk management, Computers & Security, Volume 102, 2021.

[14] Razzaque M. A., Milojevic-Jevric M., Palade M., and Clarke S., "Middleware for internet of things: a survey," IEEE Internet of Things Journal, vol. 3, no. 1, pp. 70-95, 2016.

[15] Bandyopadhyay B., Sengupta M., Maiti S., and Dutta S., "Role of middleware for internet of things: A study."

[16] Aman M.N., Chua K. C., and Sikdar B., "Mutual authentication in IoT systems using physical unclonable functions," IEEE Internet Things J., vol. 4, no. 5, pp. 1327–1340, Oct. 2017.

[17] C. Riggs, J. Patel, and K. Gagneja, "IoT Device Discovery for Incidence Response," 2019 Fifth Conf. on Mobile and Secure Services (MobiSecServ), Miami Beach, FL, USA, 2019, pp. 1-8.

[18] S. Godwin, B. Glendenning and K. Gagneja, "Future Security of Smart Speaker and IoT Smart Home Devices," 2019 Fifth Conf. on Mobile

and Secure Services (MobiSecServ), Miami Beach, FL, USA, 2019, pp. 1-6.

[19] Chatterjee U., Chakraborty R.S., and Mukhopadhyay, "A PUF- based secure communication protocol for IoT," ACM Trans. Embedded Comput. Syst., vol. 16, no. 3, p. 67, 2017.

[20] Braeken A., "PUF based authentication protocol for IoT," Symmetry, vol. 10, no. 8, p. 352, 2018.

[21] N. M. Karie, N. M. Sahri, W. Yang, C. Valli and V. R. Kebande, "A Review of Security Standards and Frameworks for IoT-Based Smart Environments," in IEEE Access, vol. 9, pp. 121975-121995, 2021.