A session key authentication system for smart homes

Jeremy Varghese K and A Vinnarasi
1M-Tech-Embedded System Technologies, 2Assistant Professor
Department of Electronics &Communication Engineering, SRM Institute of Science &Technology, Kattankulathur, Kanchipuram, Tamil Nadu, India
E-mail: jvkjeremy@gmail.com

Abstract. The Internet of Things (IoT) is a rapidly growing emerging topic of social, economic and technical significance. The spread of today's wireless information and communication technologies has changed people's lifestyle and social interactions - the next frontier is the smart home environments. A smart home consists of low capacity devices (sensors, for example) and wireless networks. These devices are networked together and they need to all work together as a secure system. The proposed scheme aims to offer a secure session key establishment scheme which can then be implemented in smart home networks. Every node (sensor and control unit) uses a small authentication token, along with a random number to establish a session key which is both secure and dynamic. A few popular attacks like denial-of-service and eavesdropping are prevented by the proposed scheme.

1. Introduction
The Internet of Things (IoT) is a rapidly growing emerging topic of social, economic and technical significance. Internet of Things refers to the network of physical objects embedded with electronics that are interconnected to exchange and gather data over the internet. These devices are usually low capacity devices with limited computational power and a way to connect to the internet. Over time, the definition of Internet of Things have been modified due to the introduction of new technologies such as a multitude of sensors, machine learning, and analytics of data in real time. An IoT system generally comprises of four components. Sensors, which collect data from their environment. For e.g. accelerometer, temperature sensor, GPS etc. The cloud, to which the data is sent to through an internet connection. This connection can be Wi-Fi, cellular, satellite, Li-Fi or even directly via ethernet. The data processing, which happens at the cloud. For e.g., it could process how much the temperature of a system is increasing or decreasing throughout the month. Finally, the user interface, which lets this information be passed down to the user. This can be in the form of a website, an app, an email or even an SMS. An often-sought-after application of IoT is smart home systems (SHS). In a smart home system, sensors, home appliances and smart devices are connected to the internet to remotely monitor and control a residential environment. These various devices converge to form an ecosystem which can then be controlled by your smartphone or even voice activated assistants. This ecosystem could, for example, help in saving energy by ensuring that devices power down or turn off automatically. Smart homes also assist elderly people, a single smart device can be used to control everything in their house. Individual disabilities can be accounted for in this case. For example, users with sight and mobility impairments can make use of assistive technology like voice control. There is no doubt that smart home systems bring great convenience to the lives of people using them, but the security of
these systems have always been a critical issue. The limited resources of the devices, coupled with the fact that users are using a public channel like Internet to communicate with them make it easy for a malicious adversary to intercept and alter the transmitted messages. Confidentiality, availability and integrity are some requirements that need to be satisfied by the exchanged information. Smart devices can range from lights to microwaves to washing machines to HVAC’s, and there is a lack of security standards for most smart devices. Furthermore, the resource constrained nature of smart devices like low memory, low processing power and low link bandwidth make it challenging to implement traditional public-key cryptosystems, such as Diffie–Hellman and Rivest–Shamir–Adleman (RSA). Thus, this project presents a secure key establishment scheme which also allows each entity to perform a mutual authentication, and only then participate in the home network. In recent years, there has been a significant amount of systems proposed for smart home security. Furthermore, many of these schemes only consider eavesdropping attackers and not about what happens when a smart device is compromised. In [1], Ling et al. demonstrates how the smart devices available in the market, in this case a smart plug, uses insecure communication protocols and lacks device authentication. Their analysis show these devices are not protected against popular attacks like brute force and device spoofing. Using these, an attacker is able to control the smart devices completely. This paper proposes to satisfy a reasonable level of security by using a lightweight session key establishment scheme. This scheme, while not resource heavy, allows two devices to undergo mutual authentication before participating in the home network while also establishing a time dependent session key in a secure manner. This scheme allows new devices to be added in easily and securely later. The rest of this paper is organized as follows. Section II conducts a literature survey on relevant existing works. Section III handles system design and the various security properties. Section IV introduces the proposed scheme and Section V analyses the proposed scheme in terms of security and Section VI analyzes the performance of the scheme. Finally, the paper is concluded in Section VII.

2. Literature Survey

Security features in smart home applications have been proposed in a number of ways, each with their own merits and demerits. Some of those literatures proposed smart home appliances have been described here. Gomez et al. [2] discusses Wireless Home Automation Networks (WHANs), comparing and contrasting multiple architecture for home automation such as ZigBee, Wavenis, Z-Wave, Insteon and IP-Based Technologies. It takes a deep dive into the features and security obstacles of the above architectures and concludes on which are designed for general purposes and which are designed for specific applications. Kumar et al. [3] introduced a lightweight and secure session key establishment scheme in smart home environments. It makes use of a short authentication token to establish a session key and authenticating the smart devices. The message integrity is maintained along with its freshness and the lightweight protocol ensured that it could be run on resource constrained devices. It was found to resist popular attacks like replay attack, denial of service attack, known key attack. Vaidya et al. [4] introduced an algorithm that is designed to provide a robust and efficient authentication mechanism. It uses an HMAC-based one-time password (HOTP) and a hash chaining technique along with smart card. If a smart card alone is used, although secure, it could be stolen by an adversary. This algorithm overcomes that disadvantage by bringing one-time passwords into the setup. Baruah et al. [5] proposed a smart home model based on If This Then That (IFTTT) system. In this system, remote users have to log on to an IFTTT server to remotely control the devices and also configure recipes. This smart home framework incorporates a captcha based One Time Password (OTP) along with a lightweight PUF function to ensure that only legitimate users can log in and manipulate the IFTTT recipe. Ashibani et al. [6] proposed the usage of a signcryption scheme based on Identity-Based Cryptography that authenticates without access to a trusted third party and provides confidentiality as well as integrity. The communication is protected against various possible attacks. It uses bilinear pairing along with Elliptic-curve cryptography (ECC) to implement this scheme. However, it glosses over the feasibility of the algorithm in resource-constrained smart devices. Hoang et al. [7] analyzed various attack methods usually implemented against smart devices. The attacks
were explored in depth and the obtained data used to suggest a completely new approach to communication between devices. Using the anonymous communication of the Onion Router, also known as TOR, to secure smart home appliances was suggested. However, the authors themselves admit that it is only a temporary solution and that this scheme cannot be used in resource constrained devices. Kumar et al. [12] also uses the concept of a session key. An anonymous secure framework was proposed which was meant to be deployed in smart homes. Using lightweight operations, the proposed framework not only provides integrity and authentication to the system, but also achieves anonymity and unlinkability for the smart devices. The session key used was dynamic as well.

3. System design and security properties

3.1. System Design
In the system being proposed, the smart home comprises of multiple heterogeneous smart devices (such as smart light, temperature sensor, and a home gateway). These devices form a Home Area Network which is being used by the resource constrained smart devices to communicate to the home gateway. The home gateway is connected to the internet and enables the user to monitor and operate the smart devices through it. The proposed system is visualized in Fig. 1. There are three main units involved in this smart home environment. They are as follows

1. Smart Devices: These can forward data to the home gateway. If needed, the home gateway can also perform queries to the same.

2. Home Gateway: It is a special node that all the smart devices are connected to. It is also connected to the internet and with it, the outside world. The smart devices send the sensor data to the home gateway and it is responsible for aggregating and, if need be, analyzing the data.

3. Security Service Provider: A trusted server which is responsible for generating the device identities and also keying material such as generated token to smart home entities.

3.2. Security Properties
[8-9] go into issues regarding security and privacy in modern Internet of Things (IoT) systems. They identify major security properties that should be possessed by any smart home. These properties are as follows

1. Mutual Authentication: A malicious entity may pretend to be an authentic entity within the system with the purpose of acquiring sensitive data collected by the smart devices with regards to the services of the smart home. Here, unauthorized network access from adversaries can be prevented if the smart devices perform mutual authentication and verify that the entities involved are legal.

2. Session Key establishment: After performing mutual authentication and verifying themselves, the legal entities should agree on a session key thereby ensuring security for further communication.

3. Message confidentiality: Smart devices all over the smart home collect data and send it to the gateway wirelessly and on open channels. This data could contain sensitive information and an adversary can eavesdrop on this communication by monitoring the wireless channels. Thus, the protocol messages are vulnerable to eavesdropping attacks and by extension, information leakage.

4. Message integrity: The users of a smart home may depend on the data from the smart devices. Keeping this data confidential does not necessarily mean that it is safe from external
modifications. Data could be tampered by a malicious entity. Message integrity would guarantee that the message the receiver received is the same as the one sent by the transmitter. In other words, it ensures that the data was not altered during transit (by an attacker).

5. Message freshness: It is not enough to guarantee only message confidentiality and authentication, but as suggested in [10], for a security protocol to be adequate, it should ensure that each received message is fresh.

6. Lightweight: Smart homes and IoT systems in general are implemented using resource-constrained chips. Thus, this scheme to establish session key should be as lightweight as possible, taking into consideration these resource-constrained smart devices.

7. Safe from popular attacks: The scheme must be able to resist many popular attacks like message replay, denial-of-service message forgery attacks.

4. System design and security properties
This section presents a scheme that satisfies the above security property requirements. The proposed system can be implemented in many scenarios including lighting, Heating, ventilation, and air conditioning (HVAC), appliance control systems, activities of daily living (ADL), security systems, smart energy monitoring and consumption reduction systems. Table 1 defines the symbols used
Table 1. Symbols and Descriptions.

| Symbols | Descriptions |
|---------|--------------|
| SP      | Security service provider |
| idₐ     | Identity of smart device A |
| Eₖ[M]   | Message ‘M’ is encrypted using secret key, ‘K’ |
| Dₖ[M]   | Message ‘M’ is decrypted using secret key, ‘K’ |
| tokenₐ  | Unique authentication token for device A |
| HMAC    | Hashed Message Authentication Code |
| h(s)    | One-way hashed function of string ‘s’ |
| ||      | Concatenation operation |
| ⊕       | Ex-or operation |
| <<      | Left-shift operator |

Our assumptions are as follows.

- A typical smart home use-case is considered where the heterogeneous smart devices contain various sensors which send data to the home gateway regularly or when sudden substantial spikes or drops are registered. For eg. Sudden spike in CO₂ would be relayed immediately.
- The security provider and gateway are trusted and tamper-proof entities. Moreover these two components share a secure connection.
- The gateway and smart devices have identical cryptographic systems. These are, for the purposes of the paper, assumed to be secure.
- The clocks of the smart devices are synchronized using Network Time Protocol (NTP) servers. Furthermore, a mutually agreed upon transmission delay (ΔT) is used to avoid replay attacks.

Three phases make up the proposed scheme: Setup, Authentication and key establishment, and ease of addition of new device.

4.1. Setup
To begin with, the security provider should register each smart device off-line by assigning identity of device and security parameters. For each smart device A, the security provider assigns an identity(idₐ), a secret key(Kₐ) and a key identifier(Kₐid). A short authentication token(tokenₐ) is also generated by the security provider. This authentication token is hashed to get Qₐ = h(tokenₐ). Security provider stores identity of device, authentication token, key and key identifier (idₐ, tokenₐ, Kₐ, Kₐid respectively) into the memory of device A. Next, it stores identity of device, key, key identifier, and Qₐ(idₐ, Kₐ, Kₐid, Qₐ respectively) in the home gateway. The security provider stores a database with all this information of each smart device. It is assumed that each assigned key has a lifetime. This lifetime is set by the security provider.

4.2. Authentication and key establishment
A mechanism to perform authentication and key establishment is presented in Fig 2. It is then explained in words below.
6

Step 1. Client sends a message requesting for the gateway to connect. This message 
\{"Connect","id_{A}","0"\} to the home gateway.

Step 2. Home gateway obtains current time, T1 and uses it to calculate 
\(C = \text{HMAC}[Q_{A}, id_{A} \| T1]\) and sends a message 
\{C, T1, id_{A}, '1'\} to device A.

Step 3. As soon as device A receives the message, it obtains current time T2. It then checks for the 
time difference ie. \((T2-T1) \leq \Delta T\) with \(\Delta T\) being an agreed upon acceptable time difference. 
If yes, proceed. Compute \(Q_{A}^{*}\) by hashing \(token_{A}\). Use the newly obtained \(Q_{A}^{*}\) to calculate 
\(C^{*} = \text{HMAC}[Q_{A}^{*}, id_{A} \| T1]\). Newly calculated \(C^{*}\) is compared with received C and if \(C^{*} \neq C\), a 
false message is generated and system is terminated. If \(C^{*} = C\), scheme proceeds. Device A 
generates a random number s. and computes 
\(N_{A} = E_{K_{A}}[id_{A}, s \oplus T1, T2]\) along with 
\(tag = \text{HMAC}[Q_{A}, id_{A} \| T1 \| T2]\) and a message containing 
\{K_{A}id_{A}, N_{A}, tag, T2, id_{A}, '2'\} is sent back 
to the home gateway.

Step 4. Home gateway obtains current time T3 and verifies that \((T3-T2) \leq \Delta T\). It uses \(K_{A}id_{A}\) to fetch the key \((K_{A})\) for the corresponding device. \(N_{A}\) is decrypted using above key and 
\(id_{A}^{*}, s^{*}, T1^{*}, T2^{*}\) are obtained. Verify \(id_{A}^{*} = id_{A}, T2^{*} = T2\). If not, halt, else continue. Calculate 
tag^{*} = \text{HMAC}[Q_{A}, id_{A} \| T1 \| T2] and verify tag^{*} = tag. Now comes the generation of the session 
key \(\sigma\). Calculate \(\sigma = h(id_{A} \| T3 \| s \| T2 \| Q_{A}^{*})\). Next compute 
\(N_{HG} = E_{K_{A}}[(s << 2) \oplus T3]\). Send 
message \{\(N_{HG}, T3, id_{A}, '3'\)\} to device A.

Step 5. Device A obtains current time T4 and verifies \((T4-T3) \leq \Delta T\). It is true, decrypt \(N_{HG}\) to 
obtain \(s^{*}\) and \(T3^{*}\). Verify \(s = s^{*}\). Compute \(\sigma = h(id_{A} \| T3 \| s \| T2 \| Q_{A}^{*})\).

4.3. Ease of addition of new device

In a smart home, new devices will constantly be added. To add a new device X, execute subsection A. 
Security provider will assign identity and required parameters (\(id_{X}, token_{X}, K_{X}, K_{Xid}\)) to the new device. 
It then passes the information of device X (\(id_{X}, K_{X}, K_{Xid}, Q_{X} (= h(token_{X}))\)) to the home gateway.
securely. Now the device is considered as one of the network. The gateway and device will execute subsection B.

5. Performance analysis

For a prototype implementation, we have used python programming language and is implemented on a raspberry pi. The symmetric key algorithm AES (Advanced Encryption Standard) is used for the encryption due to its ease of implementation. For the verification of message authentication, we have used HMAC (Hashed Message Verification Code) utilizing SHA-1. The time used by the smart devices is read from NTP servers. The various parameters used in the key establishment scheme are sized as IDs being 3-digit integers, random numbers being 5-digit integers and AES key size being 16 bytes.

![Execution Time vs Processes used in different systems](image)

**Figure 3.** Execution time vs Processes used in different systems.

Fig 3 shows the execution time taken for different operations on different schemes. These operations are 128-bit Advanced Encryption System (AES), Hashed Message Authentication Code (HMAC-SHA1) and Cipher Block Chaining Message Authentication system (CBC_MAC). AES is a symmetric-key algorithm and here we are using the 128-bit version which takes a 128-bit cypher key. Message Authentication Codes are a way for the receiver to ensure that the message is coming from the real sender and not an actor or a malicious entity. Two different types of MAC’s are shown in this diagram, CBC-MAC and HMAC. In CBC-MAC, the message is encrypted with a block cypher algorithm in CBC mode to create a chain of blocks. In this way, the proper encryption of the previous block is depended on by the present block. The interdependence of each block means that even a change of a single bit in the plaintext will cause the final MAC to change in a way that cannot be obtained without knowing the key to the cypher. We can also observe that CBC-MAC is not used by [12] or the proposed system. For the other authentication code, HMAC SHA-1 was chosen. The principle behind this operation is that a hash function is applied to the message. Hash functions are
one-way functions which mean that it is extremely difficult to obtain the plaintext message from the output of the function. In the message authentication code application for hashed functions, the key is used as a seed so that that value can only be obtained when the key is used along with the hashing operation. Here we can see that our setup takes less time for each operation. In the proposed system, only one type of message authentication code, HMAC, was used.

Table 2. Execution time comparison.

|                      | [2]   | [12]  | Proposed |
|----------------------|-------|-------|----------|
| Total Executing      | ≈ 0.17 | ≈ 0.123 | ≈ 0.10   |
| time (in seconds)    |       |       |          |

Table 2 shows us the total execution time in seconds of the proposed scheme with the existing schemes. Here we can observe that the new scheme requires less time than existing schemes for overall operation. Observing the table, we can deduce that compared to the closest previous time, the new time is 18.7% faster.

Table 3. Computation Cost Comparison.

|                      | [2]   | [12]  | Proposed |
|----------------------|-------|-------|----------|
| Hash Operation       | 2H    | 2H    | 2H       |
| CBC-MAC              | 1MAC  | -     | -        |
| HMAC                 | 1HMAC | -     | 2HMAC    |
| Cryptosystem         | 1E+1D | 1E+1D | 1E+1D    |
| XOR                  | -     | 3XOR  | 1XOR     |
| Shifting Operations  | -     | -     | 1        |

Table 3 shows computation cost comparisons of the proposed scheme with various other schemes. The proposed scheme requires 2 hashing operation, 1 encryption and 1 decryption using AES, 2 Hashed message authentication codes, 1 XOR operation and 1 shifting operation. These are lightweight operations that are excellent to use in resource-constrained devices.

6. Security analysis

This section discusses the resilience of the proposed system against popular attacks (message-forgery, message replay, known-key, message forgery, denial-of-service). The security properties mention in Section III-B (mutual authentication, session-key establishment, message confidentiality, integrity and freshness) are also checked with respect to the proposed scheme.

To analyze the scheme, the adversary is assumed to be able to overhear, intercept and synthesize any message. This threat model is referred to as the Dolev-Yao model [11]. In this model, Basically, the only limit of the malicious entity are the constraints of the cryptographic methods used. Let us assume an intruder, Bob, wants to access the network. The rest of this section shows how the proposed scheme is resilient to the following attacks and how the security properties are satisfied.

6.1. Masquerade Attack

When an attacker gains access to a system using a forged identity, it is known as masquerade attack. Here an attacker, Bob can intercept the first message (refer Section IV-B) and pretend to be a legal gateway by sending the message \( f_{C_{Bob}}T_{Bob}, id_A, '1' \) to device A. Here, \( C_{Bob} \) will not be verified due to Bob not knowing \( Q_A \) and hence using some other key. Bob will be detected as not a legal entity and the session will be terminated by device A. Furthermore, as long as Bob does not know the parameters of
the key (token), which is allotted by the security provider and known only to the legal entities, he cannot recreate Q. Thus, masquerading as a home gateway is not possible for Bob and device A remains safe. Bob can also intercept message 2 \{K_{ab}, N_a, tag, T_2, id_a, ‘2’\} to and pretend to be a smart device messaging the gateway. But he cannot capture the contents of N_A since it is encrypted and the key is known only to legal entities.

6.2. Message Forgery Attack

Sending of a message to deceive the receiver as to who the real sender is called message forgery attack. Bob is able to intercept and record previous messages between the real smart device and home gateway. He can then try to forge the required hidden parameters (C from message1 and tag from message 2). Bob sends a message \{C_{Bob}, T_{1Bob}, id_a, ‘1’\} to device A. This will be rejected at device A since C_{Bob} = \{Q_{ABob}, id_{ABob} || T_{1Bob}\} and Q_{ABob} is different from the real Q_A and the resulting HMAC is different. The process is terminated.

Bob can also attempt to send an artificial message 2 \{K_{ab}, N_a, tag, T_2, id_a, ‘2’\} to the home gateway. Similar to the above situation, tag_{Bob} = HMAC \{Q_{ABob}, id_{ABob} || T_{1Bob}\} and Q_{ABob} is different from the real Q_A and the resulting HMAC is different. The process is terminated.

6.3. Replay Attack

Replay attack happens when a malicious entity records and replays legal messages between legal entities. Bob can intercept message 1,2 and 3 and attempt to send them back to the respective recipients. Consider message 1. At time T_{Bob}, Bob sends a message \{C_{Bob}, T_{1Bob}, id_d, ‘1’\}. Now, device A checks for time difference between T_{1Bob} and T_2 which may be greater than ΔT which is the mutually agreed upon time difference. Even if message is sent within the time difference, the process stops when the C_{*} = HMAC[Q_{ABob}, id_A || T_{1Bob}] calculated from T_{1Bob} does not match with the C = HMAC[Q_A, id_A || T_1] sent over because that C uses a different time, T_1. The process is terminated. Consider message 2. At time T_{Bob}, Bob sends a message \{K_{ab}, N_a, tag, T_2, id_a, ‘2’\}. Now, the home gateway checks for time difference between T_{2Bob} and T_3 which may be greater than ΔT. Even if message is sent within the time difference, the process stops when N_A is decrypted and the encrypted T_2 is found to be not equal to T_{2Bob}. The process is terminated. Similarly, a replay of message 3 fails as well.

6.4. Known-key Attack

Assume Bob is able to eavesdrop on many wireless messages over a long period of time and studied some session keys. Assume that Bob somehow manages to get a past session key, σ (= h(id_A || T_3 || T_2 || Q_*)). He cannot reuse the same key because every session key is novel in that it incorporates timestamps. Due to the usage of one-way hash functions, he cannot extract the Q_A required to create a new session key. Thus, this scheme is resilient to known-key attacks.

6.5. Denial-of-Service Attack

Bob can initiate a Denial-of-Service attack by replaying old messages. This proposed scheme makes use of timestamps to mitigate such attacks to an extent. This is why the first thing done in steps 2,3 and 4 (in Section IV-B) is checking whether the time difference is less than the accepted time difference. In this way. The proposed scheme is able to resist Denial-of-Service attacks.

6.6. Security Properties

The properties discussed in Section III-B are fulfilled by the proposed scheme.

(1) Mutual Authentication: Unauthorized access to the smart home network is prohibited with the help of a proper mutual authentication. It means that both the parties involved, not just one, are verified before any sensitive data is exchanged. This is done through the in the first message by C (=HMAC[Q_A, id_A || T_1]) in the first message and N_A (=E_{KA}[id_A, T_1, T_2]) in the second
message. If Bob tries to fabricate message 1 and send \{C_{Bob}, T1_{Bob}, id_A, '1'\} to device A, it fails because the C* calculated by device 1 will differ. Similarly, home gateway can verify a message is coming from the real device A (id_A* = id_A) when it is able to decrypt N_A by using the key K_A since K_A is known only to the legal devices.

(2) Session key establishment: After each authentication a session key is established. The session keys make use of two different timestamps thereby ensuring that each session key is different.

(3) Message Confidentiality: Adequate confidentiality is provided to the messages in these proposed schemes. For example, \(C = \text{HMAC}[Q_A, id_A || T1]\), \(N_A = \text{E}_{K_A}[id_A, T1, T2]\), \(\text{tag} = \text{HMAC} [Q_A, id_A || T1 || T2]\), \(N_{HG} = \text{E}_{K_A}[^1, T3]\). Furthermore, a unique key, K_A exists for each device A and this is known only by two entities, the gateway and the device A. The gateway uses the identifier, K_{Aid} sent by the device to find out what key is being used. If a key different from K_A is used (for e.g. By an intruder), the gateway cannot decrypt the message to get anything meaningful.

(4) Message Integrity: Both the gateway and device A compute an HMAC. These can be computed only by the legitimate devices on the network. If an intruder, Bob, tries to replicate C in message 1 by calculating \(C = \text{HMAC} [Q_{A, Bob}, id_A || T1]\), it fails because device A calculates a different C using the real Q_A and the validation fails. For the same reason, trying to replicate tag by using \(\text{tag} = \text{HMAC} [Q_{A, Bob}, id_A || T1 || T2]\), it fails due to the same reason.

(5) Message freshness: Each message makes use of timestamps. This ensures that the messages \{C, T1, id_A, '1'\}, \{K_{Aid}, N_A, tag, T2, id_A, '2'\}, \{N_{HG}, T3, id_A, '3'\} are fresh each time they are sent.

7. Conclusion
In this paper we propose A Session Key Authentication system for smart homes that can be utilized by low resource embedded systems. The proposed scheme is shown to meet the security goals of being resistant to the following attacks: Masquerade attack, Message Forgery attack, Replay attack, Known-key attack, Denial-of-Service attack. The security properties of mutual authentication, session key establishment, message confidentiality, message integrity and message freshness have also been shown to meet the algorithm. Simultaneously, the scheme has been shown to reduce the amount of time required to run the same. A proof-of-concept implementation has also been undertaken by using the raspberry pi.

8. References
[1] Ling Z, Luo J, Xu Y, Gao C, Wu K and Fu X 2017 Security vulnerabilities of internet of things: A case study of the smart plug system IEEE Internet of Things Journal 4 1899-909
[2] Gomez C and Paradells J 2010 Wireless home automation networks: A survey of architectures and technologies IEEE Communications Magazine 48 92-101
[3] Kumar P, Gurtov A, Inatti J, Ylianttila M and Sain M 2015 Lightweight and secure session-key establishment scheme in smart home environments IEEE Sensors Journal 16 254-64
[4] Vaidya B, Park JH, Yeo SS and Rodrigues JJ 2011 Robust one-time password authentication scheme using smart card for home network environment Computer Communications 34 326-36
[5] Baruah B and Dhal S 2018 A two-factor authentication scheme against FDM attack in IFTTT based Smart Home System Computers & Security 77 21-35
[6] Ashibani Y and Mahmoud QH 2017 An efficient and secure scheme for smart home communication using identity-based signcryption In2017 IEEE 36th International Performance Computing and Communications Conference 1-7 IEEE
[7] Hoang NP and Pishva D 2015 A TOR-based anonymous communication approach to secure smart home appliances. In 2015 17th International Conference on Advanced Communication Technology 517-525 IEEE

[8] Islam K, Shen W and Wang X 2012 Security and privacy considerations for wireless sensor networks in smart home environments In Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design 626-633 IEEE

[9] Porambage P, Ylianttila M, Schmitt C, Kumar P, Gurtov A and Vasilakos AV 2016 The quest for privacy in the internet of things IEEE Cloud Computing 3 36-45

[10] Perrig A, Szewczyk R, Tygar JD, Wen V and Culler DE 2002 SPINS Security protocols for sensor networks Wireless networks 8 521-34

[11] Dolev D and Yao A 1983 On the security of public key protocols IEEE Transactions on Information theory 29 198-208

[12] Kumar P, Braeken A, Gurtov A, Innati J and Ha PH 2017 Anonymous secure framework in connected smart home environments IEEE Transactions on Information Forensics and Security 12 968-79