A Lightweight Pairing Protocol for IoT Devices in Smart Homes

Ali Tufail

Faculty of Computer and Information Systems, Islamic University of Madinah, Saudi Arabia

Summary

The concept of Internet of Things (IoT) and smart homes is now a reality. The stake of both academia and industry has increased in the devices related to IoT and smart homes. A lot of work has been done in this domain in the last decade or so. However, the architectures and frameworks related to this domain are still evolving. IoT enabled devices and smart homes are promising revolutionary changes in the normal day to day activities for the consumers by providing easy access and control, cheaper products with less energy consumption and a considerable difference in their monthly bills. However, at the same time a lot of challenges related to security and privacy of consumers data are emerging. These devices contain a considerable amount of information related to consumers like name, phone number etc. The security and privacy issues are posing a big threat to the legitimacy of these devices. However, due to the inherent limited memory and processing capabilities for most of the IoT devices, the traditional security mechanisms cannot be applied. This paper proposes a lightweight pairing protocol that not only works towards making IoT devices communication secure but also contribute towards making the smart homes less vulnerable to all the wireless attacks. Also, consumers information will stay safe and adversaries will not be able to steal any data stored in those devices. A unique signature is extracted depending upon various factors of IoT devices like location. This signature is used in the mutual verification and to prove the legitimacy of the IoT devices. Moreover, a unique concept of trust has been introduced. The trust builds over time and once it gets over the certain threshold the devices start communications. Without building trust devices will not communicate to each other. Moreover, security and complexity analysis of the proposed protocol has been performed that indicates that the proposed protocol is lightweight and has the ability to protect IoT devices from various threats and attacks.

Key words:

Smart Homes, IoT, Security, Key Establishment, Privacy

1. Introduction

The concept of Internet of Things (IoT) is gaining momentum and it has been predicted that IoT enabled devices will be an essential part of smart homes [6]. Both industry and academia are showing a great interest in the research and development of a lot of concepts related to IoTs. The main problem which is arising out of the development of these devices is of security and privacy. Most of the data that consumers store on these devices can be easily hacked or adversary can launch several attacks on these devices. Also, bot network can be formed that can be used to launch DDOS attacks. The architectures and frameworks related to this domain are still evolving and a lot of challenges related to security and privacy of consumers data are still persistent. The IoT devices contain a considerable amount of information related to consumers like name, phone number etc. The security and privacy issues are posing a big threat to the legitimacy of these devices. However, due to the inherent limited memory and processing capabilities for most of the IoT devices the traditional security mechanisms cannot be applied. In order to establish a secure environment for the communication amongst these IoT devices, two important factors are very crucial i.e. authentication and key exchange. After performing successful authentication devices establish legitimacy of each other and the next step to perform secure communication is of key exchange. Various approaches have been developed for the conventional networks but for IoT enabled networks inherent constraints usually serve as a bottleneck. Public key cryptography (PKC) and symmetric key cryptography have been utilized typically to secure the communication. Nevertheless, both techniques have drawbacks and benefits over each other.

As shown in related work section, some amount of research is being done by researcher community in order to find a way to enhance the security and privacy of these devices. This paper aims at solving this problem by proposing a novel lightweight paring protocol for IoT enabled networks. This protocol has been specifically designed to work efficiently and effectively in a smart home-based environment. SKC and hashing has mainly been deployed for most of the communication for resource constrained devices. However, for the resource rich devices like gateway and sink public key has also be used. Nevertheless, typically, in a smart home environment even for the IoT devices resources are better and less constrained especially power and computing capability. Signature based mutual authentication has been proposed. Moreover, a unique concept of trust has also been introduced. The trust builds over time and once it gets over the certain threshold the devices start communications. Without building trust devices will not communicate to each other. Additionally, location is being used to identity the boundary of local (within a single home) IoT devices. It will help to pin point the illegitimate users.

The rest of the paper has been organized as follows. Section 2 talks about the related, section 3 introduces the proposed protocol, section 4 discusses results and analysis and finally section 5 concludes the paper.

Manuscript received April 5, 2019
Manuscript revised April 20, 2019

2. Related Work

The area of Wireless Sensor Networks has been the focus of both academia and industry. However, with the advancement and materialization of the concepts related to smart homes these sensors based IoT devices are now in the limelight. A lot of products are being launched that contain any of those sensors like movement, voice, heat sensors. This area is going to have a significance contribution towards modern and contemporary lifestyle of human beings. A lot of products are being deployed; however, these products contain some sensitive information of consumers like their name, address, telephone number or any other ID related information. If any adversary gets hold of those devices by coming into the communication range, it will have a huge breach of privacy. This issue can hinder the widescale deployment of these devices. Therefore, like in [1], [2], [3], [4], researchers are now investigating various ways and means through which they can not only make sure that the consumer information stays protected but also, they are trying to make sure to protect those devices from any possible attacks from adversaries.

Authors in [1] propose a security framework for IoT devices. Their framework is based on the MobilityFirst future Internet architecture. Their main aim is to integrate the local IoT systems with the global internet while making sure that the usability, interoperability and security is preserved.

Authors in [2] talk about proposing a security framework for cloud-centric IoT solution. They suggest the process to integrate various objects of smart homes with their proposed architecture. Furthermore, they highlight major issues and challenges faced by many communication protocols designed to work in smart homes.

[3] surveys various security and privacy issues pertaining to IoTs. The authors divide their survey into four useful segments. The first segments surveys limitations and solutions for IoT devices. The second segment focuses on highlighting various security attacks on IoT devices, the third segment talks about authentication process and lastly the fourth segment talks about security issues in regards to different layers of OSI model.

Authors in [4] talk about different security challenges that IoT devices have to face. They propose a new context-based pairing mechanism, which they call Perceptio. Their proposed mechanism makes use of the context to authenticate other IoT devices. They claim that the implementation of their mechanism proved that devices became more secure and the attacks were reduced since they were authenticating the IoT devices.

In [11] authors summarize various issues related to the security of the IoT network and devices. They talk about both the opportunities and threats related to IoT. The major focus of the paper is Computer Aided Design (CAD). After providing the survey of IoT and its techniques the paper suggests the guidelines to develop IoT CAD security techniques.

Authors in [12] discuss the possibilities of coming up with the new approach for the design and deployment of security techniques for IoT based networks. They claim that the traditional security approaches cannot work effectively for the new diverse IoT network. Therefore, they lay emphasis on the need of creating new approaches for securing these networks.

[13] presents a critical analysis of the present security techniques of the IoT networks. However, they claim that the existing approaches are uncertain and do not provide adequate and much needed security for the IoT devices and network. They propose a new confidentiality-based approach that they claim provides the needed security and privacy to the users.

Authors in [14] talk about security concerns in the IoT network. They claim that due to constrained nature of the IoT networks the provided security is not enough and is not providing the needed protection against various attacks. They provide a lightweight security scheme which they claim can provide needed security to the IoT devices and the networks.

3. Proposed Protocol

This section introduces the proposed protocol for the IoT devices deployed in a smart home. Following are the salient characteristics of the proposed protocol:

1. The proposed protocol focuses on the IoT devices used in a smart home environment.
2. The protocol focuses on the security and privacy issues of IoT communication.
3. The protocol suggests security for Infrastructure based IoT networks.
4. A unique feature of signature extraction has been proposed that will be used for enhanced security and mutual authentication of devices.
5. The protocol proposes the use of symmetric key so that the efficiency of IoT devices is not affected.
6. The protocol suggests the use of a distinctive feature of trust which is built over time. If the trust factor is less than a given acceptable value no communication is made amongst devices.
7. The wireless nature of communication has been considered therefore the key establishment is performed using double factors i.e. signature and trust.

3.1 Network Architecture and Assumptions

Our proposed protocol assumes a heterogeneous IoT network where a variety of devices have the capability of communicating with each other preferably in a smart home environment (refer to the below figure). It is assumed that the IoT network will have the capability to work in infrastructure mode where IoT devices communicate with the help of gateway node. Gateway is typically connected to the central node where information sharing and information analysis will be performed. CN plays the role of third party in the key exchange and key authentication phase. It is further assumed that one event can be reported by multiple IoT devices and the position of the IoT devices are fixed in the smart home. Adversary will be outside of the boundary of the home and cannot penetrate inside physically.

Figure 1 gives the idea about the assumed smart home environment. It shows typically various IoT devices are installed inside the boundary of the home. These devices might include temperature sensor, security camera, motion detector etc. As per our assumed network, a gateway and command node are also part of the network. These nodes exist in the following hierarchy (consult figure 2): Command Node, Gateway, sensors/actuators. Command node and gateways are more resource rich in comparison to sensors/actuators (mainly referred as IoT device (s))

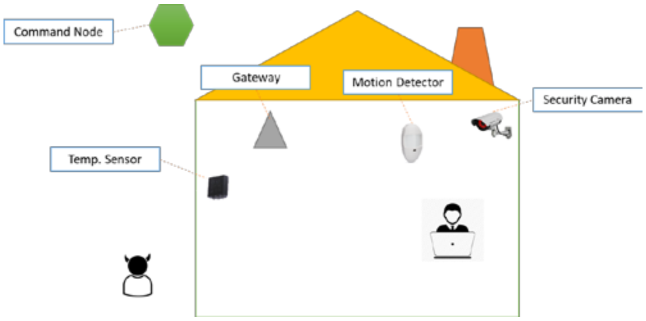


Fig. 1 Smart Home Environment

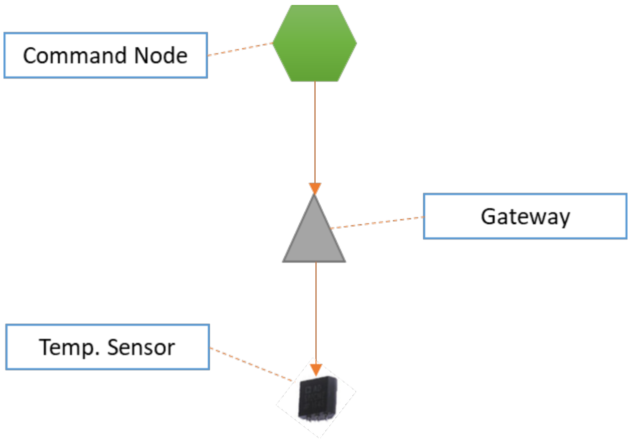


Fig. 2 Communication Hierarchy of the IoT Based Network

3.2 Threat Model

As shown in the previous figure the adversary will stay outside of the boundary of the smart home. However, due to the wireless nature of the communication amongst IoT devices the adversary can take advantage of the loopholes and the weaknesses of the network and might try to attack the network with the following attacks scenarios:

a. Man-in-the-Middle Attack
Adversary might get hold of the messages being transferred and then launch this attack. Also, he might be able to read and alter the information

b. Shamming Attack
Attacker even sitting outside of the boundary (but still within the transmission range of the IoT device) of the home might still be able to convince any of the IoT device to connect by fooling them and by pretending to be a legitimate IoT device

c. Replay Attack
Attacker might get hold of the legitimate traffic and then later replay the same messages by pretending to be a legitimate device

d. Spoofing
Adversary can also launch a spoofing attack and might succeed in fooling the authentication process

e. Sniffing
Due to wireless nature of the communication, adversary might be able to get hold of the network traffic and might use that to infer or steal private information like identity, password etc.

Table 1: Important notations and their description

Notation	Explanation	Notation	Explanation
G	Gateway	RTC	Request to connect
CN	Command Node	OTC	OK to connect
Ki	Common Initial Key	UID	ID for IoT devices
GID	ID for gateway	RTC	Request to connect
Nonce	One-time random number	Timestamp	Time used for the process of authentication
Ks	Session key used after authentication for the rest of the communication	Ka	Temporary key generated by the gateway to do mutual authentication
PKc and PRc	Public and private keys for the command node	CONCT	This is a function that concatenates all the arguments and return the result
Encrypt	Function that is utilized to encrypt the given message (first argument) with the given key (second argument).	Decrypt	Function that is utilized to decrypt the given message (first argument) with the given key (second argument).

3.3 Main Phases of the Pairing Protocol

Following are the main phases of the proposed protocol:

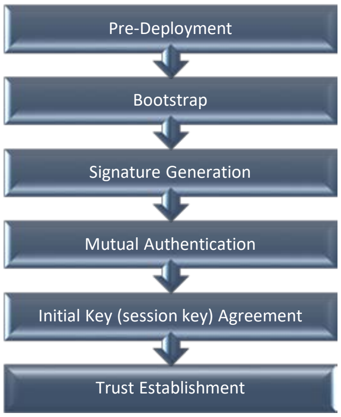


Fig. 3 Main Phases of the Security Protocol

The figure above lists down the main phases of the proposed security protocol. Each phase has been defined subsequently

Pre-Deployment Phase

In this phase the keying material and the keys are deployed in both IoT devices and the gateway. Each device is deployed with the following:

a. Common Initial Key (Ki)
b. Unique Identity (GID for gateway and UID for IoT devices)
c. Public Key (PKc) for communication with the CN
d. Keying material like PKC (just for the gateway) algorithm and SKC algorithm (for both gateway and IoT Devices)
e. Hashing algorithm for both IoT devices and gateway
f. Location information for the IoT devices. This location information is pre-determined and is given as per the location of a particular device. As mentioned before, it has been assumed that the location of the IoT devices has been fixed in the setup of a smart home.

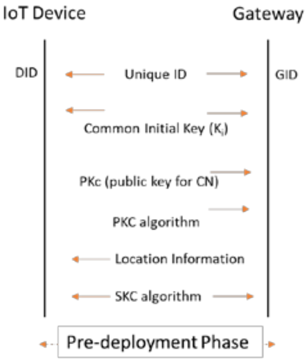


Fig. 4 Pre-deployment Phase

Bootstrap Phase

In this phase the initial messages are exchanged and the IoT devices or the gateway show their willingness to connect to each other. A broadcast message (encrypted with the common initial key) is sent by the gateway. It has been assumed that this broadcast message can be sent only by the gateways.

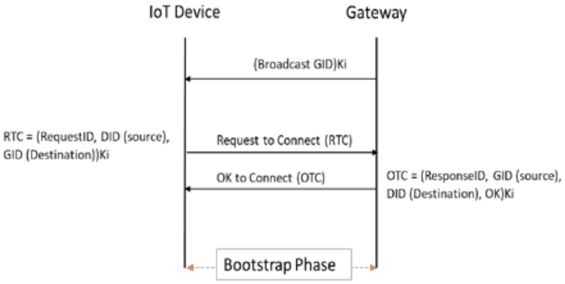
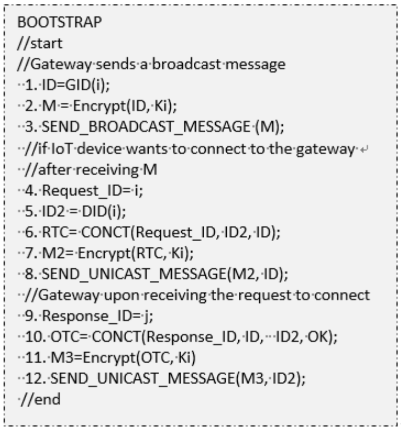

IoT Device
Gateway
{Broadcast GID}Ki
RTC = {RequestID, DID (source), GID (Destination)}Ki
Request to Connect (RTC)
OK to Connect (OTC)
OTC = {ResponseID, GID (source), DID (Destination), OK}Ki
Bootstrap Phase

Fig. 5 Bootstrap Phase

However, on the receipt of this broadcast message the IoT device can initiate the connection request. This broadcast message by the gateway is like a beacon message and is repeated after fixed intervals. If IoT device is willing to connect it will send a request to connect (RTC) message and upon receiving OK to connect (OTC) message from the gateway the IoT device will initiate the authentication and key establishment phase. This process has been shown in the figure 5.
Following is the algorithm explaining the overall bootstrap process in more detail.



```
BOOTSTRAP
//start
//Gateway sends a broadcast message
  ID=GID(i);
  M = Encrypt(ID, Ki);
  SEND_BROADCAST_MESSAGE (M);
  //if IoT device wants to connect to the gateway
  //after receiving M
  Request_ID= i;
  ID2 = DID(i);
  RTC= CONCT(Request_ID, ID2, ID);
  M2= Encrypt(RTC, Ki);
  SEND_UNICAST_MESSAGE(M2, ID);
  //Gateway upon receiving the request to connect
  Response_ID= j;
  OTC= CONCT(Response_ID, ID,   ID2, OK);
  M3=Encrypt(OTC, Ki)
  SEND_UNICAST_MESSAGE(M3, ID2);
 //end
```

Fig. 6 Algorithm for Bootstrap Phase

Signature Generation

This phase is an important phase of the suggested protocol. A unique signature is produced by the IoT device and then using the same set of parameters a signature is produced by the gateway. This signature is later utilized in the mutual authentication phase.

On receiving OK to connect message from the gateway the IoT device first generates the signature and then send it to the gateway. The IoT device generates a random one time use nonce, timestamp and the location (initially stored at the pre-deployment phase). The use of nonce and timestamp protects from various attacks like replay attack. The IoT device sends all the previously stated parameters along with Response ID, DID and GID to a one-way hash function. The hash value along with the nonce and the timestamp is encrypted with the common session key and sent to the gateway.

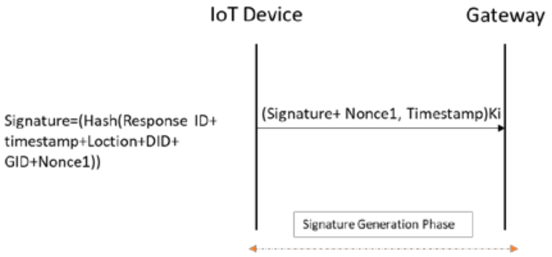
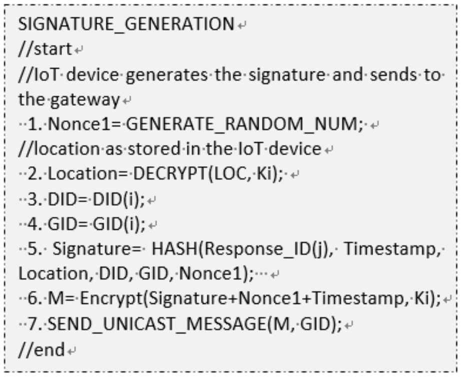

IoT Device
Gateway
Signature=(Hash(Response ID+ timestamp+Loction+DID+ GID+Nonce1))
(Signature+ Nonce1, Timestamp)Ki
Signature Generation Phase

Fig. 7 Signature Generation Phase

The gateway on receiving this encrypted message will decrypt it and then will trigger the next phase of mutual authentication. The above figure shows the overall process of signature generation while the below figure explains the algorithm and the process in more detail.



```
SIGNATURE_GENERATION
//start
//IoT device generates the signature and sends to the gateway
  1. Nonce1= GENERATE_RANDOM_NUM;
//location as stored in the IoT device
  2. Location= DECRYPT(LOC, Ki);
  3. DID= DID(i);
  4. GID= GID(i);
  5. Signature= HASH(Response_ID(j), Timestamp, Location, DID, GID, Nonce1);
  6. M= Encrypt(Signature+Nonce1+Timestamp, Ki);
  7. SEND_UNICAST_MESSAGE(M, GID);
//end
```

Fig. 8 Algorithm Signature Generation Phase

Mutual Authentication

This phase involves three level of devices i.e. IoT device, gateway and command node. This phase is triggered once the gateway receives the encrypted signature from the IoT device. The gateway authenticates the IoT device and if the signature matches it would mean that the IoT device is a legitimate device. The IoT device authenticates the gateway with the help of the CN.

On receiving the signature message, the gateway decrypts it using the common key i.e. Ki. It then gets all the required information like signature, nonce, timestamp. In order to get the accurate location of the IoT device the gateway sends a message to the CN to get the location of that IoT device. Upon receiving the location from the CN it will use it along with the other parameters to compute the signature using hash function. If this hash matches with the hash sent by the IoT device the gateway will be sure that the IoT devices is the legitimate device and not a malicious node. Only if the location is the same the same signature will be produced. By now it has authenticated the IoT device so it will increase the trust by certain factor (let's say by 1) and store information in the table. CN also has a table that contains the location with the ID of the IoT device or the gateway and other information (as shown in below table).

After authentication, the gateway generates a temporary key Ka for the purpose of authenticating itself with the IoT device. It then encrypts it with the public key of the CN and send it to the CN. Gateway also encrypts it with the common key Ki and sends this Ka to the IoT device. CN on receiving the Ka from the gateway will encrypt it along with the GID using the temporary key Ka and will send it to the IoT device.

The IoT device will first receive the key from the gateway and will decrypt it using Ki. Upon receiving the GID and the same key from CN the IoT device will authenticate the legitimacy of the gateway. It will also increase the trust for the gateway by certain factor (let's say by 1) and store information in its table. At this stage the mutual authentication process has finished where both the IoT device and the gateway have authenticated each other.

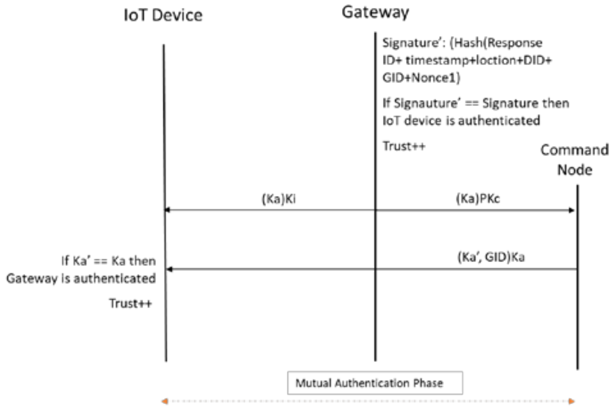
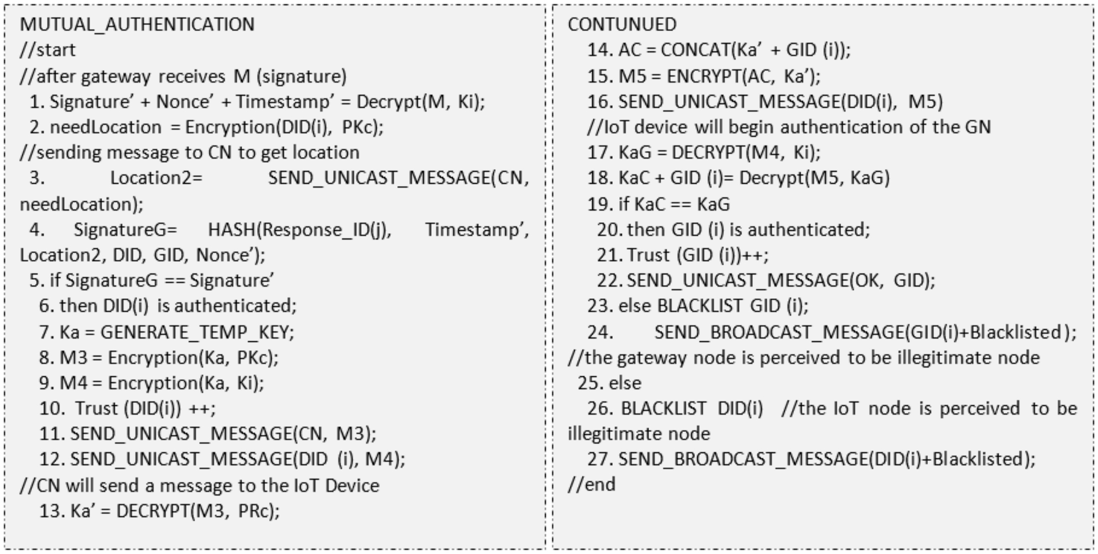

IoT Device
Gateway
Signature': (Hash(Response ID+ timestamp+loction+DID+ GID+Nonce1)
If Signauture' == Signature then IoT device is authenticated
Trust++
Command Node
(Ka)Ki
(Ka)PKc
If Ka' == Ka then Gateway is authenticated
(Ka', GID)Ka
Trust++
Mutual Authentication Phase

Fig. 9 Mutual Authentication Phase

Table 2: Location and other information at the Command Node

----- --	Node ID	Node Type	Node Location	Node Status	----- --

The above figure 9 shows the overall process of mutual authentication phase while the below figure 10 explains the algorithm and the process in more detail.



```
MUTUAL_AUTHENTICATION
//start
//after gateway receives M (signature)
  1. Signature' + Nonce' + Timestamp' = Decrypt(M, Ki);
  2. needLocation = Encryption(DID(i), PKc);
//sending message to CN to get location
  3. Location2= SEND_UNICAST_MESSAGE(CN, needLocation);
  4. SignatureG= HASH(Response_ID(j), Timestamp', Location2, DID, GID, Nonce');
  5. if SignatureG == Signature'
    6. then DID(i) is authenticated;
    7. Ka = GENERATE_TEMP_KEY;
    8. M3 = Encryption(Ka, PKc);
    9. M4 = Encryption(Ka, Ki);
    10. Trust (DID(i)) ++;
    11. SEND_UNICAST_MESSAGE(CN, M3);
    12. SEND_UNICAST_MESSAGE(DID (i), M4);
//CN will send a message to the IoT Device
    13. Ka' = DECRYPT(M3, PRc);

CONTUNUED
    14. AC = CONCAT(Ka' + GID (i));
    15. M5 = ENCRYPT(AC, Ka');
    16. SEND_UNICAST_MESSAGE(DID(i), M5)
    //IoT device will begin authentication of the GN
    17. KaG = DECRYPT(M4, Ki);
    18. KaC + GID (i)= Decrypt(M5, KaG)
    19. if KaC == KaG
      20. then GID (i) is authenticated;
      21. Trust (GID (i))++;
      22. SEND_UNICAST_MESSAGE(OK, GID);
    23. else BLACKLIST GID (i);
    24. SEND_BROADCAST_MESSAGE(GID(i)+Blacklisted);
//the gateway node is perceived to be illegitimate node
  25. else
    26. BLACKLIST DID(i) //the IoT node is perceived to be illegitimate node
    27. SEND_BROADCAST_MESSAGE(DID(i)+Blacklisted);
//end
```

Fig. 10 Algorithm Mutual Authentication Phase

Session Key Agreement

After the mutual authentication has been successful the session key agreement phase will start. A session key is produced by the gateway and is then shared with the IoT device. The gateway generates another nonce and a key length value (a number). With help of the hash function, the gateway then produces the session key Ks by using the previous nonce shared by IoT device along with this new nonce value and the key length. It encrypts this key Ks along with the new nonce and key length by using the initial key Ks. The IoT device receives this message and after decrypting it produces the same key and compares it with the key Ks. If it is the same IoT device becomes sure of the legitimacy of the message and the key. By this stage session key has been generated and agreed upon between both IoT device and the gateway.

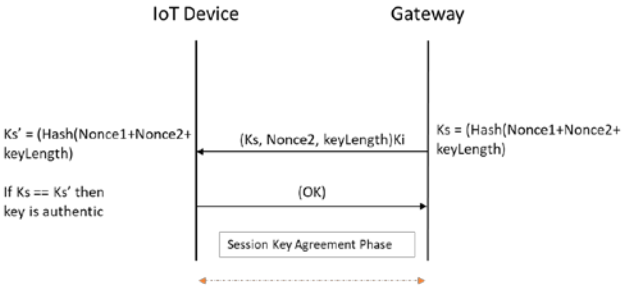
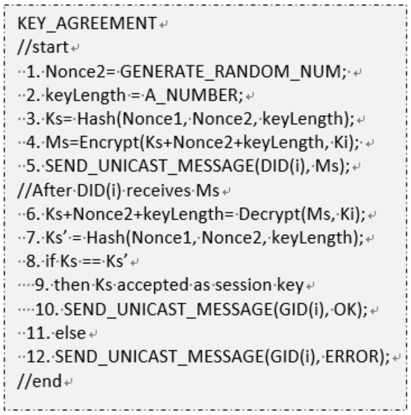

IoT Device
Gateway
Ks' = (Hash(Nonce1+Nonce2+ keyLength)
(Ks, Nonce2, keyLength)Ki
Ks = (Hash(Nonce1+Nonce2+ keyLength)
If Ks == Ks' then key is authentic
(OK)
Session Key Agreement Phase

Fig. 11 Session Key Agreement Phase

The above figure shows the overall process of session key agreement phase while the below figure explains the algorithm and the process in more detail.



```
KEY_AGREEMENT
//start
  1. Nonce2= GENERATE_RANDOM_NUM;
  2. keyLength = A_NUMBER;
  3. Ks= Hash(Nonce1, Nonce2, keyLength);
  4. Ms=Encrypt(Ks+Nonce2+keyLength, Ki);
  5. SEND_UNICAST_MESSAGE(DID(i), Ms);
//After DID(i) receives Ms
  6. Ks+Nonce2+keyLength= Decrypt(Ms, Ki);
  7. Ks' = Hash(Nonce1, Nonce2, keyLength);
  8. if Ks == Ks'
    9. then Ks accepted as session key
    10. SEND_UNICAST_MESSAGE(GID(i), OK);
  11. else
  12. SEND_UNICAST_MESSAGE(GID(i), ERROR);
//end
```

Fig. 12 Algorithm Session Key Agreement Phase

Trust Establishment

This phase is an important and crucial part of the proposed protocol. Before the start of the communication both IoT device and the gateway establish and check the trust. If the trust is below certain level (a threshold) then the communication is not started unless the trust reaches that level. In this case the overall process will be repeated from phase one onwards. Trust is built over time and also depends upon the previous communication experiences. The threshold value has to be set with care in order to make a balance in trusting an illegitimate device and delaying or creating problems in communicating with the legitimate device. The following figure explains the overall trust establishment phase.

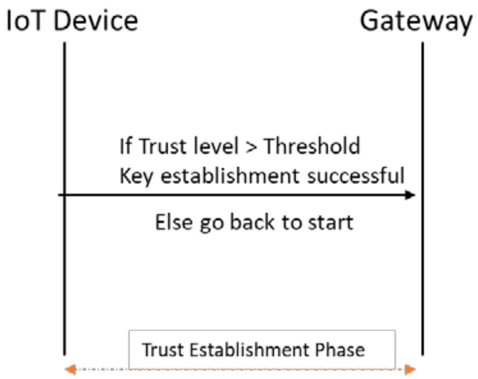

IoT Device
Gateway
If Trust level > Threshold
Key establishment successful
Else go back to start
Trust Establishment Phase

Fig. 13 Trust Establishment Phase

The tables below show the kind of information that will be stored with the gateway and the IoT devices. It also stores the important trust factor for each device they communicate with. Signal strength is sometimes utilized to guess the location of the device. It can be an additional factor in order to make sure that the communication is not taking place with illegitimate or unauthenticated devices

Table 3: Location and other information at the gateway for IoT devices

DID	Location 1 (longitude and latitude)	Signal Strength	Current Session Key	Trust

Table 4: Location and other information at the IoT device for the gateway

DID	Location 1 (longitude and latitude)	Signal Strength	Current Session Key	Trust

4. Results and Discussion

This section talks about the results and analysis related to the proposed protocol. The first part discusses the security analysis and the second part discusses the complexity analysis related to the project.

4.1 Security Analysis

This section performs the security analysis of the proposed protocols. Various attacks have been discussed in regards to protection the proposed protocol provides against those attacks.

Following are the main security properties that are provided by the proposed protocol:

Confidentiality and Privacy: All communication between IoT Device, gateway and command node use encryption techniques hence ensuring complete confidentiality and privacy.

Authentication: Before communication, gateway and IoT device perform mutual authentication and later on they use the shared session key. This process ensures the authenticity of communication.

Non-repudiation: Unique signature-based authentication provides non-repudiation property. Both, IoT device and the gateway cannot deny performing any communication.

Integrity: The use of session key to encrypt communication make sure that the data has not been modified in any manner. Also, hashing techniques are deployed in various phases of the proposed protocols which helps in identifying if the data has been modified by illegitimate device.

The following discussion proves that the proposed protocol has the capacity to guard the IoT based communication in a smart home setup against different attacks. Various attacks and the relevant scenarios have been discussed. It is worth noting here that with the help of encryption and authentication the chances of launching a successful attack reduce immensely.

Man-in-the-middle-attack: This attack will be extremely difficult to launch because of various functionalities provided by the protocol. Since the malicious node will be outside of the premises of the smart home the location could be identified. Also, with the help of signal strength IoT device or gateway will be in a position to know the approximate location of the malicious node. In order to successfully launch man in the middle attach the malicious node will have to 1)compromise the session key 2) compromise the common shared key 3)Compromise or spoof the ID of the legitimate node 4)Prove its location to be inside the smart home 5)Show the trust factor above the set threshold value 6) Most importantly fool powerful gateway and command node. Therefore, chances of this attack being successful are very rare.

Shamming attack: It will be very difficult for the attacker to fool the legitimate nodes and launch this attack. First of all, the location of attacker will speak of its illegitimacy and secondly the mutual authentication and trust factors will minimize the chances of this attack to be launched.

Replay attack: With the use of timestamp it will be difficult to launch this attack.

Spoofing: This attack is another possibility to enter the legitimate network. However, with the help of the proposed protocol the chances that the attacker will be able to spoof or fool a legitimate node are minimum. The mutual authentication process is required before the start of the any communication. With the help of the CN it will be very easy to verify the identity of any node. Moreover, establishing trust and location of nodes are some other factors that are required as part of the proposed protocol. Any node that is trying to launch this attack can be blacklisted and any further communication with it will be halted.

Sniffing: This kind of attack will be next to impossible since the proposed protocol is using encryption for all of its communication. No ID or information is being passed in raw format. Therefore, in order to sniff the network traffic, the adversary will have to compromise the network and the nodes which will be very difficult.

There are certain attacks that could be launched from inside the network. Although, as per our assumption the attacker will be outside of the boundary of the smart home, however, even if we believe that somehow the attacker successfully broke into the house and has the capacity to launch attacks like **selective forwarding attack** the proposed protocol provides protection against those kinds of attacks. One event is usually reported by multiple IoT devices so if a malicious node is not reporting that event with the help of gateway and command node that malicious node can easily be spotted and blacklisted.

Other attacks like **sinkhole and wormhole attacks** are also very powerful attacks and are even more difficult to detect. However, in our proposed protocol all the communication goes through the gateway and then the command node and here these two nodes are aware of the network topology and the location of the IoT devices/nodes. It will not be difficult for these two nodes to identify a malicious node. If an attacker introduces a lot of illegitimate traffic to the network the gateway and the command node can detect this anomaly and can blacklist the node to avoid any further escalation of the attack.

4.2 Complexity Analysis

The following section performs the complexity analysis of the proposed protocol. The main point of focus for this analysis is the number of required messages for the overall process of authentication and key establishment with varying number of IoT devices.

Number of messages required to perform authentication and key establishment are:

1. Broadcast Message from the gateway to the IoT devices
2. IoT device sends a unicast message to join the gateway
3. Gateway confirms the connection request with a unicast message
4. IoT device sends signature in a unicast message
5. Gateway sends two unicast messages: one for the CN and other for the IoT device
6. CN sends a unicast message to the IoT device
7. Gateway sends the session key in a unicast message to the IoT device
8. IoT device sends a unicast message confirming the receipt of the session key

In total number of messages that have been exchanged for whole of this process are:

- 1 broadcast + 7 unicast messages

These are not a lot of messages and will not add a burden on the IoT devices especially in a smart home setup. Let's say there are "N" number of IoT devices in the smart home so a total of 7N messages will be required for all the IoT devices to authenticate and connect to the gateway. The following graph shows this observation with varying number of IoT devices:

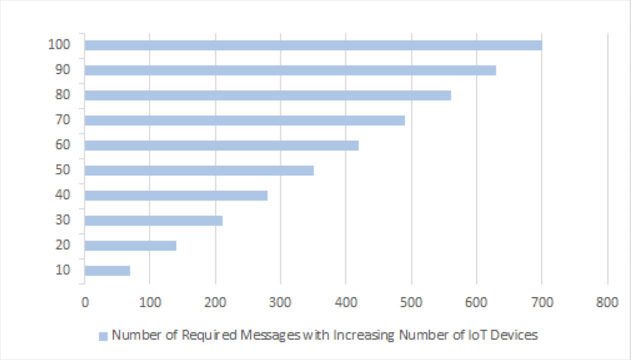


Fig. 14 Number of Required Messages with varying number IoT Devices

5. Conclusion

This paper presents a distinctive lightweight pairing protocol for Internet of Things (IoT) enabled devices that is tailored for the environment of smart homes. The protocol utilizes encryption techniques to provide authentication, confidentiality and privacy. Protocol introduces two important unique features of signature and trust. Signature is extracted based on various parameters and is used for mutual authentication. Trust is also necessary in order to have secure communication between IoT devices and the gateway. No communication takes place if the level of trust is below certain threshold value. Another important parameter i.e. location is used in order to make sure that the IoT devices are within the boundary of the smart home and also to differentiate between legitimate and illegitimate IoT devices. The proposed protocol is analyzed and assessed against several threats and attacks that could be launched on the IoT enabled devices and networks. Security analysis and complexity analysis has also been performed that show that the proposed protocol is not only lightweight but also provides an acceptable level of security, privacy, confidentiality, network survivability and resilience to various attacks.

Acknowledgement

This research work was funded by the Deanship of Research, Islamic University of Madinah, Saudi Arabia.

References

[1] Liu, X.; Zhao, M.; Li, S.; Zhang, F.; Trappe, W , A Security Framework for the Internet of Things in the Future Internet Architecture. Future Internet 2017, 9, 27.

[2] Biljana L. Risteska Stojkoska, Kire V. Trivodaliev, A review of Internet of Things for smart home: Challenges and

solutions, Journal of Cleaner Production, Volume 140, Part 3, 2017, Pages 1454-1464

[3] Y. Yang, L. Wu, G. Yin, L. Li and H. Zhao, "A Survey on Security and Privacy Issues in Internet-of-Things," in IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1250-1258, Oct. 2017.

[4] J. Han et al., "Do You Feel What I Hear? Enabling Autonomous IoT Device Pairing using Different Sensor Types," 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, US, , pp. 678-694.

[5] http://ci.emse.fr/teaching/majinfo/iot/2017/IotApplications.pdf

[6] https://www.edureka.co/blog/iot-applications/

[7] https://aioti.eu/wpcontent/uploads/2017/03/AIOTIWG01Report2015-Applications.pdf

[8] Yousuf, Tasneem & Mahmoud, Rwan & Aloul, Fadi & Zualkernan, Imran. (2015). Internet of Things (IoT) Security: Current Status, Challenges and Countermeasures. International Journal for Information Security Research. 5. 608-616. 10.20533/ijisr.2042.4639.2015.0070.

[9] http://www.internet-of-things-research.eu/pdf/

[10] https://bwn.ece.gatech.edu/presentations/IoT%20Trends%202017-04.pdf

[11] Xu, Teng & Wendt, James & Potkonjak, Miodrag. (2014). Security of IoT Systems: Design Challenges and Opportunities. IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD. 2015. 10.1109/ICCAD.2014.7001385.

[12] Riahi Sfar, Arbia & Challal, Yacine & Natalizio, Enrico & Chtourou, Zied & Bouabdallah, Abdelmadjid. (2013). A Systemic Approach for IoT Security. Proceedings - IEEE International Conference on Distributed Computing in Sensor Systems, DCoSS 2013. 351-355. 10.1109/DCOSS.2013.78.

[13] U Farooq, M & Waseem, Muhammad & Khairi, Anjum & Sadia Mazhar, Pakistan. (2015). A Critical Analysis on the Security Concerns of Internet of Things (IoT). International Journal of Computer Applications. 111. 1-6.

[14] bin Rabiah, Abdulrahman & K. Ramakrishnan, K & Liri, Elizabeth & Kar, Koushik. (2018). A Lightweight Authentication and Key Exchange Protocol for IoT. 10.14722/diss.2018.23004.