A Straightforward and Efficient Approach to Secure Smart Home Communication using Identify-Based Cryptosystems

M. Mazhar Rathore*, Sushil Chaurasia, Dhirendra Shukla
Dr. J. Herbert Smith Centre
University of New Brunswick
Fredericton, New Brunswick, Canada
{rathore.mazhar, sushil.chaurasia, dshukla}@unb.ca

Elmahdi Bentafat
Division of Information and Computing Technology
College of Science and Engineering
Hamad Bin Khalifa University, Qatar
Email:ebentafat@hbku.edu.qa

***Abstract*—With the growing practical implementation of smart home, the attacks on smart homes are proportionally increasing. Residents can only be benefited from smart home technology if they and their home-assets are secured against cyber-attacks. A number of PKI-based communication security models have been proposed for data authentication and confidentiality in smart homes. However, it is not convenient for a home device with the limited capacity to store, verify, and manage public keys (certificates) of all other devices. Identity-based cryptography (IBC) is one of the asymmetric cryptographic solutions that does not require certificates. However, due to the central storage of the secret at the key generation center (*KGC*), the security fully relies on the KGC in IBC environment. Thus, to resolve these issues while providing the security to smart homes, in this paper, we proposed a straightforward and light-weight security model based on IBC, wheel pairing, and elliptic curves. The proposed model performs distributed key generation where the main secret is generated by all participating home devices, instead of a central KGC. We designed a complete protocol, which illuminates the fundamental steps of new device enrollment, distributed key generation, device to device encryption, data integrity, and entity authentication. Moreover, the commitment procedure is introduced that ensures no party can change its partial-secret after he has committed to it. The elliptic curve cryptography (ECC) based Diffie–Hellman (*DH*) model is deployed for session key generation for device to device data encryption, whereas IBC-based private key is used for signatures. Finally, the feasibility of the model is evaluated by implementing the system on various numbers of IoT machines, while considering them as home devices. Also, the security of the proposed model is verified technically and formally by a software verification tool called Automated Validation of Internet Security Protocols and Applications (*AVISPA*) against popular known attacks.**

I. Introduction

Home area network is one of the areas where the Internet of Things (IoT) has a major impact due to benefits t hat the technology could bring to such an environment. A home that is equipped with IoT devices is named as a smart home. We can find a lot of sensors capturing every phenomenon happening in a smart home, along with a set of controlled actuators reacting to the changing environment. A lot of devices in our homes are connected to the internet and become part of our IoT home network, like security cameras, smart locks, smart screens, new-generation washing machines, etc. Sensors sense the home environment and transmit the information to actuators to behave as per the sensed environment. Providing high level of security to home area network (HAN) is extremely essential in this technological era after the practical implementations carried out to smart homes. If a hacker gets inside a smart home network by compromising one of its connected devices, he will be able to gain control over all the devices connected to that network. So, implementation of a smart home without security measures not only a risk to home assets but also to the residents. For Instance, the home auto-door lock can be opened with a signal from a mobile device. If the door device has weak authentication mechanism or the signal transmission is not fully secured, the hacker might open the door either by capturing and replaying,or regenerating and transmitting the similar signal. Also, with unsecured smart home, your enemy can analyze your daily routine, penetrate into your security surveillance system, and play with your home devices to harm you physically. Thus, the communication security is a vital part of your smart home.

Various communication security protocol for device authentication have been proposed by researchers using public key infrastructure (PKI) [1]. In a PKI environment, every device has to store all the certificates of other devices to communicate with them. These certificates hold the public keys of all the devices that are issued and signed by a certification authority (CA). In a smart environment, where IoT devices have limited storage and computation resource, the option to use PKI is not suitable. In addition, PKI relies on a trusted party known as certificate authority (CA). These CA are consulted by all the devices in the home area network (HAN), especially during the authentication phase. Such a centralized environment causes ‘single point of failure (SPOF)’ and requires a lot of computation power and communication overhead on the CA level [1]. At the same time, distributing and revoking the certificates becomes a very hard task in large home networks. To overcome these drawbacks, other schemes have been proposed in the literature to substitute PKI-based schemes. One of the promising alternatives is the identity-based Encryption (IBE).

IBE allows users in a network to communicate and ex-

change messages securely using identities as public keys. This communication and exchange is facilitated without relying on the CA and without storing any of the peers' keys. IBE needs just a trusted key generation center (KGC) that is responsible for generating private keys for each user of the network. While the public key of a user is nothing but his identity (can be his name, his id#, his IP, or other information) that could uniquely identify the user. Since this information is known by all the users in the network, no one has to send one's public key or certificate to other party. In an IBE environment, the centralized server i.e., KGC is responsible for generating and distributing the private keys of all network users using a common secret. The authentication phase between users becomes less dependent on the centralized server. For instance, if Alice wants to communicate with Bob, she signs the message using her private key, then encrypts the signed message with Bob's public key (which is nothing more than his name or network address), and sends the encrypted message to Bob. When Bob receives the encrypted message, he decrypts it using his own private key to get the signed message, which will be verified using the sender's ID (i.e, Alice's name or network address).

Even though the IBC removes the overhead of storing and managing certificates, but it requires a third party (KGC) with the stored secret for generating private keys for all participating devices. Thus, if the functionality of the KGC breakdowns or is compromised, the whole network (the keys of all the peers) is compromised. Therefore, researchers started focusing on threshold-based cryptographic system as an alternative of IBE for encryption and signature. In this type of cryptography, the ability to sign and decrypt, is shared between n peers. Where joining t+1 peers is required to perform the operations of decrypting or signing. Formally, if we are considering (k, n)-threshold cryptosystem, which refers to k out of n threshold, where $1 < k < n$, we are splitting the encrypting or signing key amongst n peers. More precisely, it's the exponent d that will be split into n pieces. By following this approach, we are making sure that: 1) Any k, or more, out of n peers will be able to perform signing or encrypting messages operations. While always keeping private the value of the global secret, which is the exponent d, on one hand, and the secrets of each peer towards the other peers on the other hand. 2) It is completely impossible to complete the signing or encryption operations if less than k peers try to sign or encrypt. The problem with threshold-based signature is the communication overhead. t number of devices have to be communicated for every message for key reconstruction, message decryption or signature. The overhead grows when we have more devices or frequent communication among devices in a network.

Thus, in a smart home environment of devices with limited resources, PKI and threshold based cryptography is not a suitable security solution. Whereas, identity based cryptography (IBC) requires central storage of secret and key generation at KGC. Therefore, in this paper, we overcome the issues in the existing IBC and introduced a new communication security model by taking benefits of PKC, IBC, and distributed key management. The proposed system used IBC for key management, generation, and authentication but unlike IBC, our system does not require KGC server for generating private keys using a stored secret. Rather, the keys are generated in a distributed environment where all the devices participates in the key generation of other keys. The main secret is ambiguously generated by all the participating devices. Every device generates his private key by asking partial-secret from each of the other devices in such as fashion that their partial-secret are not revealed to the key generating device. Overall, the proposed security system uses an identity (such as, device serial number, name, or IP) as a device public key rather than using certificates It does not rely on KGC, as every device generates its private key by using a distributed secret.

The Contribution of our article is many folded as follow.

- A Complete security model is proposed for smart homes that performs symmetric encryption, IBC-based signature, and distributed private-keys generation by commitment procedure.
- A security protocol is designed that depicts the overall model for secure communication in smart homes. it allows to securely enroll a new device in to the system, generates public and private key using distributed secret, and generates symmetric keys using Diffie-Hellman based encrypted key exchange (DH-EKE) key exchange for symmetric data encryption from each device to any other device. The protocol provides all the security services including confidentiality, integrity, and authentication.
- The security of the system is evaluated logically as well as formally using 'Automated Validation of Internet Security Protocols and Applications (AVIPA)' tool. The results shows that the protocol is safe from all the popular attacks including man-in- the-middle (*MITM*) attack, replay attack, non-repudiation attack, etc.
- Finally, the feasibility of the system is tested by implementing the system on IoT devices, while considering them as home devices. The system's efficiency results are convincing, which shows that the proposed model is perfectly implementable on real-environment of smart home.

The rest of the paper is organized as follows. Section II describes the work done in the field of smart home security and identity-based cryptography. Section III discusses the preliminaries concepts and technologies used in our model. Section IV presents the proposed work in details. Section V evaluates the proposed work in terms of security verification formally and informally. It also evaluates the feasibility of the system by its practical implementation on IoT devices. Lastly, we concludes our work in Section VI.

II. Related work

Smart homes are major components of smart cities where the energy, water consumption, and fire control can be managed through IoT devices [2], [3]. Compromising smart homes may have serious consequences on overall city. To secure smart homes, communication security is now diverting from PKI models to identity-based models due to certain limitations in PKI. Shamir proposed the first identity-based schema in 1984 [4]. He proposed an identity-based scheme for the emailing systems which doesn't rely on public certificates. But, his scheme could not be implemented practically until 2001 [5].

In the literature, IBE schemes have been constructed using pairings, like the studies done by Boneh et al. [6] and Sakai et al.[7], or using quadratic residues which is the case of Cocks' proposed scheme [8]. Recently lattice-based encryption has been also being used in IBE. The first lattice-based IBE scheme has been proposed by Ducas et al. [9] in 2014, which has been practically examined by Sarah et al. [10]. They were the first who practically implemented the lattice-based IBE as a C library with all the functionalities of Sarah et al. scheme.

Due to the obvious advantages of not using certificates by IBC, the security researchers started working to use it for IoT applications. In a constrained IoT network, the ECC-based IBE implementation is most widely used option [11]. Researchers also used other PKC-based cryptosystem such as RSA and ElGamal along with the IBC [12]. However, these approaches are considered to be costly in terms of speed because of their computation on large exponents. Thus, Yang et al. [11] used Boneh and Frankin approach [6] with the coordination of ECDH key exchange [13] to propose an IBE scheme called IBAKA for IoT-based sensor network. With their scheme, while bootstrapping the secret key, they reduced the number of bi-linear mapping to two and point multiplications to three, which are very costly operation in IBC. Same effort was made by Szczechowiak and Collier [14] to propose a lightweight scheme called Tiny IBE for authenticated key distribution in heterogeneous sensor network. They avoided the computation of bi-linear mapping and exchanged just two messages to generate and share session key between two nodes. Alike, Yao et al. [15] also avoided the use of pairing with the practice of ECDDH instead of bi-linear Diffie–Hellman assumption while designing attribute-based encryption scheme for IoT environment. They proved their scheme in attribute based selective-set model. Besides, Mao et al. [16] proposed IBE for secure communication among IoT nodes that uses fuzzy logic and eluded the disadvantages of existing scheme of relying on random oracle models, using long parameters, providing security only to selective-ID model, and loose security reduction.

Smart home is also one of the IoT applications that lies in constrained network type. In 2012, Nicanfar et al. [17] proposed an IBC scheme to manage keys in smart homes. In their proposal i.e., enhanced IBC (EIBC), Nicanfar et al. introduced an efficient private key refreshment method, while providing multicast keys needed in the home area networks (HAN). The solution was improved by the same authors in 2014 [18], where they were able to reduce the number of exchanged packets and the number of steps to three and three, instead of four and five respectively. In another study done by Jacobsen et al. [19], the researchers focused on optimization of the bootstrapping steps of wireless devices in home area network (HAN) based on IBC while establishing the key session. The proposed protocol also protected the HAN from adversaries in its setup phase, as well as in other network operations. In a different study [20], Qinghai introduced new techniques to integrate biometrics for the authentication phase in SG. The author used fingerprints in order to improve the users' privacy in SG communications.

Even though the IBC have advantages over PKI when implemented in smart home environments, IBC requires a secret that is stored on a central server called key generation center (KGC) to generate private keys for all the participating devices. Thus, if the central server is compromised, the whole setup is compromised. Moreover, identity based approaches are unprotected from key-escrow attacks. The KGC knows private key of each of the node in the network. Thus, he can pretend to be any node in the network and intercept all the transmissions on the network. Therefore, the KGC required to be more safe and trustworthy. To cater to this issue, the researchers started focusing on threshold-based cryptosystems, even though this system came up before IBC by Shamir in 1978 [21]. These systems use multiple devices in the network to encrypt or sign the message for authentication. Such a scheme was taken forward in 2005 by Gennaro et al. [22], where they have proposed a new protocol in which the key is divided into n secrets. In order to reconstruct the key, t+1 secrets should be combined. Meanwhile, to produce a signature, $2t + 1$ parties are needed to generate it with no need to reconstruct the key. Later, they improved their scheme and made it more optimal [23] for digital signatures in bitcoin wallets. They used elliptic curve based digital signature algorithm that does not require an honest majority of devices. Similarly, Nguyen [24] presented another such scheme that uses RSA. The problem with these schemes is with more communication overhead, as the same message is communicated among multiple devices for signature and encryption. Thus, our scheme considered IBC with the distributed secret sharing mechanism by removing the drawbacks in existing systems to secure smart home. The overall proposed work is elaborated in the next section.

III. Preliminaries:

A. Smart home

In smart home, we have different sensors, actuators, and devices attached to each other through a communication infrastructure. Sensors sense the environment, whereas actuators behave based on the current environment. For Example, the temperature sensors sense the temperature, smoke sensors sense the smoke in the air, and the smart shoes sense the person's activity. On the other hand, the light, the AC, and the refrigerator work as actuators that adjust themselves according to the sensed environment. Thus, every device needs to be communicated to other devices in a smart home ecosystem. This communication can be done through any of the technology such as Bluetooth, WiFi, Ethernet, the cellular network, or the internet. It is also possible that some of the devices is outside the home and connected to the home network through internet. For example, the mobile admin device might be far from the actual home and connected to the home network through internet. In this case, we need a server (such as, proxy server) that is able to store the IP information of the mobile home device (as mobile device might keep changing IP). Also, In a Smart City environment, the smart home data might be transmitted over the Internet to various authorities and servers based on the nature of the data collecting device. For Instance, the home temperature and smoke data, when exceeds from a serious threshold, may be sent to the fire station. The electricity consumption measurements might be directed to smart grids through smart meters. In this scenario, the server have the responsibility to connect to the external environment. Nest Smart Home Hub*(https://nest.com/)*, that was developed by Nest Lab and now purchased by Google, also works in a same fashion with a cloud services.

All these communications among internal devices and to external entities must be secured in order to avoid severe damage to the property, as well as to the house residents. For example, your smart home offers you to open the main door-lock by sending the signal from your personal mobile device to the door-lock. If the signal is transmitted over an unsecured channel and or the locker is not equipped with a secure authentication mechanism, then the thief or hacker can easily open the door through re-transmitting the same signal. In a worse case, with the unsecured home data, your enemy can analyze your routine schedule to find out the best time to physically harm you. Thus, security is an essential part of the smart home infrastructure. Since, we are using the identity-based cryptography to achieve the internal communication security among home devices, every device has a unique identity (his MAC address, his IP, or his name) that is used to achieve confidentiality and authentication among devices. The proposed security model does not consider server (like proxy server/nest cloud server) as a home device. The purpose of the server is to just provide the communication among devices and pass the message from an internal device A to other external internet-connected device B. Thus, in order to secure the a message from device A to an external home device B (that passes through the proxy server), the system uses the IDs and keys of A and B. Therefore, the server is unable to decode any message from any device A to B.

B. Identity based cryprosystem

The proposed security model utilizes the identity based cryptosystem (IBC) that avoids the exchange of public keys, storing them on constrained IoT devices, and taking services from third parties for digital certificate authentication. As per Shamir's identity based encryption (IBE) [4], any IBC approach should have four fundamental algorithms i.e., 1) setup, 2) extract, 3) encrypt/sig, and 4) decrypt/verify.

1) *Setup*: This algorithm generates public IBC parameters (*PubParams*) and a master secret S that is used to generate network public and the corresponding private keys for each of the participating devices. *Pubparams* includes details of message space *M*, cipher space *C*, *hash* functions, starting values, public key, and others. The master secret remains secret only to Key KGC.
2) *Extract*: The extract algorithm takes a string $\{0,1\}^*$ corresponding to the device ID, and the master secret 'S' as input. It generates a private key *D* for the device corresponding to its ID. *D* = $F(ID, S)$
3) *Encrypt/Sign*: The algorithm to encrypt (or sign) the message using the ID (or the private key in case of signature). Ciphertext 'C' of Message 'M' is computed as, $C = Encrypt(PubParams, ID, M)$, and the signature '$Sig$ of M is extracted as, $Sig = Encrypt(PubParams, D, M)$.
4) *Decrypt/Verify*: Decryption algorithm decrypts 'C' using the private key D corresponding to ID. In case of signature, It should verify the signature using the device ID as the public key. $M = Decrypt(Pubparams, ID, C, D)$.

IV. Proposed ID-based security model for smart Home

A. Overview of security model

Our proposed model assumes that every home device i has a unique identifier ID_i, which serves as its public key. The private key of every device *i* is generated by a common network secret(*S*)–the server's secret in the identity-based cryptosystem–multiplied by its ID (ID_i). In our model, the secret 'S' is generated distributively by cooperation of all the devices, where none of the device actually knows the S. There is also a common network public key that is a function of the secret S and a public parameter *P*. The message is encrypted using ID_i and network public key (K_{pub}), where as the decryption is performed using the corresponding private key (D_i). The signature is a function of private key, where as the verification is a function of ID and the network public key (K_{pub}).

The model also assumes that one of the home devices (can be a computer or a mobile device) serves as an admin or the owner device. The admin device allows a new device to be enrolled in the network. At the start, the admin device has its own secret that serves as the main secret 'S for the generation of K_{pub} and the it's private key. The admin device also generates all the public parameters for overall security model such as, P, hash functions, $MapToPoint$ function, etc., which will be discussed later. We assume that when a new device wants to enroll into the network, only the owner or administrator is responsible to allow it. The device puts a request to the admin device, which generates a temporary one-time password (OTP) for the new device to communicate to the admin device securely. The new device uses the OTP for authentication and to complete the secure enrollment process. When a new device is enrolled, every device in the smart home needs to re-generate its partial-secret S_i. Accordingly, new keys are generated by initiating a key distribution protocol. Thus, the private key of every device and the network public key are re-computed by combining all the secrets (s_i). The devices share their secrets in such a way that the device's partial-secret (s_i) does not reveal to others.

As we know, the PKC is costly in terms of processing time for continuous communication from device to device. Therefore, the routine peer-to-peer device communication is secured using symmetric key encryption. The model performs symmetric session-key exchange using Diffie-Hellman (DH) protocol to generate a symmetric key for every peer-to-peer device pair.

B. Distributed ID-based keys generation for all device

Unlike traditional IBC schemes, our proposal performs the distributed key generation and does not require S to be stored at KGC or at admin device. We used the key generation model presented in [25] where the main key-generation secret is computed distributively across all participating home devices. The scheme is based on Weil pairing over finite fields and elliptic curves. It uses a bilinear mapping that can be defined as: Let G_i (points on the Elliptic curve over F_p) and G_j(a subgroup of F_p^2) are two cyclic groups whose order is large prime *q*. In this case, we consider G_i as an additive group and G_j as a multiplicative group. A map denoted by ê, i.e., ê :

G_i x $G_i \rightarrow G_j$ is a bilinear map, if for all *X, Y* $\in G_i$ and all *a, b* $\in Z$, the mapping ê (*aX, bY*) = $\hat{e}$ $(X,Y)^{ab}$.

The designed scheme composed of three sub-algorithms i.e., 1) MapToPoint, 2) Setup, 3) Extract

MapToPoint: Let p and q are two primes such that $p = 2 \mod 3$ and $p = 6q-1$ and $q > 3$. *E* is elliptic curve $y^2 = x^3+1$ over F_p. *MaptoPoint* converts the *ID* in the form of {0, 1} to a point Q_{ID} of order *q* over *E/F*$_p$ as follows.

1) Find $V_0 = F(ID)$ and $U_0 = (V_0^2 - 1)^{1/3} = (V_0^2 - 1)^{(2p-1)/3} \mod p$. Where *F* is a hash function such that, $F : \{0,1\}^* \rightarrow F_p$.
2) $Q = (U_0; V_0) \in E/F_p$.
3) $Q_{ID} = 6Q$. as Q_{ID} has to be order q.

Setup: The algorithm is run by admin at the start while doing network setup. Since at start, the admin is the only device in HAN so, its secret is the master secret i.e., $S = S_a$. The algorithm works as follows.

1) Select a random point P of order q over E/F_p.
2) Select an admin random secret $S_a \in Z_q^*$
3) $K_{pub} = S_a.P$.
4) Pick a cryptographic hash function $H : \{0,1\}^n \times \{0,1\}^n \rightarrow F_p$. where n is the message size.
5) Thus, the global system parameters are $PubParams = \{p, n, P, Kpub, H\}$. The master secret is $S = S_1 + S_2 + S_3 + \ldots + S_N$. But Meanwhile $S = S_a$ (as we have only one device admin).

Extract: The algorithm is run by every device '*i*' to generate its private key (D_i) by using distributed secret S and its public key 'ID_i'.

1) $QID_i = MapToPoint(ID_i)$.
2) Select a device random secret-share $S_i \in Z_{q^*}$.
3) $D_i = S.QID_i = S_1.QID_i + S_2.QID_i + S_3.QID_i + \ldots + S_N.QID_i$.
   $K_{pub} = S.P = S_1.P + S_2.P + S_3.P + \ldots + S_N.P$

Where *N* is the number of devices in HAN. S_j is the secret of device *j*. ($S_j.QID_i$) and ($S_j.P$) are shared from device *j* to device *i* so that device *i* can generate his private key and the network public key K_{pub}. The secret of device *j* i.e., S_j will not be revealed to device *i* and vice versa because of discrete logarithmic assumption (as it is the multiple of secret value and public value over elliptic curve).

C. Authentication mechanism used in the smart home

Every message in the smart home is authenticated by digital signatures. Our model applies identity-based signature scheme proposed by Choon-Cha-Cheon [26] for authentication and integrity while generating the keys.

IBSign: This algorithm signs a message with the private key (D_i) of device *i*. Thus signature $Sig = F(D_i, M, t \in Z_q)$, which works as follows.

1) Choose a random $t \in Z_q$.
2) $X = t.QID_i, h = H(M, X)$, $Y = (t + h).D_i$.
3) $Sig = (X, Y)$

IBVerify: This algorithm verifies the identity-based signature $Sig = (X, Y)$ signed on message *M* for an identity ID_i, as follows.

1) Calculate $h = H(M, X)$.
2) $Verify(P, K_{pub}, X + h.QID_i, Y)$ is a valid *Diffie-Hellman* tuple. If *YES*, the signature is valid.

   Proof: In a group G of order $q \in E/F_p$, the valid *Diffie-Hellman* tuple is always in the form of $(P, S.P, Q, S.Q)$. Thus, If $Sig = (X, Y)$ is a valid signature of a message *M* for an identity ID_i, then $X = t.QID_i$ and $Y = (t + h).D_i$. Thus,$(P, K_{pub}, X + h.QID_i, Y) = (P, K_{pub}, (t + h).QID_i, (t+h).D_i) = (P, S.P, (t+h).QID_i, S.(t+h).QID_i)$, as desired.

D. Identity-based Commitment scheme to secure distributed key generation process

In the our security model, every smart home device creates its own partial-secret and share it with others while generating keys. As earlier mentioned, the device shares its partial-secret (S_i) by multiplying it to a public parameter 'P' in order to make sure that the secret cannot be revealed to other devices (this is because of the discrete logarithmic assumption). But, it is still possible that a selfish or effected device can perform analysis on them and generate its own partial-secret based on others' partial-secrets, if it collects all the secret-shares from other devices prior and then shares its own later. Thus, such device has a control over the main secret, which compromises the overall network. Hence, to avoid this problem, every device must have to commit its selected secret value to other in a hidden way and reveals it later when everyone is committed their values. Later, every device confirms the partial-secrets from other devices by comparing the shared secrets and the committed values. To achieve this solution, we designed a commitment scheme that makes sure that every home device cannot change the secret value after he has committed to it. The commitment scheme has three phases i.e., 1) commit phase 2) reveal phase, and 3) verify phase.

Commit phase: a home device chooses a secret and specified it to other devices in hidden way. Let S_i is partial-secret generated by device *i*. Admin selects a public commit parameter R, i.e., a random-point on the elliptic curve of order q over E/F_P. The device *i* also selects a random value r_i for commitment that would later be used for its secret value verification. The final commitment from device *i* is ($COM_i := S_i.P + r_i.R$), which shows that the device *i* is committed to secret value S_i.

Reveal phase: When every device received commitments from all other devices, the device has to reveal its secret value to others so that other devices can generates keys. In the reveal phase the device sends the revealed value as $(S_i.P, r_i)$.

Verify phase: In this phase, the receiving device verifies that the partial-secret sent by the device *i* is same as the earlier committed value. The device takes the revealed value as $(S_i.P, r_i)$, calculate $S_i.P + r_i.R$, and compare it with the COM_i. With this procedure, the device *i* cannot change its value after he has committed to it.

TABLE I. SYMBOLS AND FUNCTIONS USED IN THE PROTOCOL DESIGN

Symbols	Details	Symbols	Details
PWD	One time password	*rand()*	Random number generation
ID_i	Identity of device *i*	*IBSign()*	Digital Signature using IBC
QID_i	point generated against Device *i* ID	*Ver()*	signature, time stamp, hash or other parameter verification
IDsList	List of home devices	*SEnc()*	Symmetric key Encryption
S_i	Device i's secret share.	*SDec()*	Symmetric key decryption
K_{pub}	Main public key for the network	K_{mn}	Symetric key for device *n* and device *m*
Ts	Time Stamp	*RegReq*	New Device registration request
r_i	Random value generated by device *i*	*PubParams*	IBE public parameter generated by admin for whole network
COM_i	Commitment sent by device *i*	*SecReq*	New Secret generation and sharing request from new device
		NewCOMReq	New Commitments, secret generation and sharing request from admin device

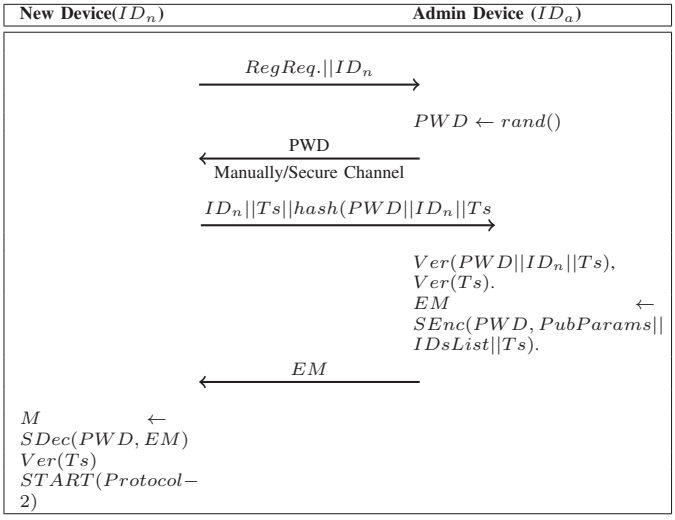
E. Proposed communication security protocol for smart home

The protocol describes the overall communication scenario of the security model. The protocol is aimed at securing the smart home from popular attacks such as, man in the middle (*MITM*) attack, replay attack, non-repudiation attack, etc., while assuring data and device authentication, data integrity and data secrecy. Table I shows the symbols and functions used in the protocol design. Overall the proposed protocol composed of three main phases including 1) new device registration and enrollment phase, 2) distributed key generation, and 3) the session key generation and exchange phase. The design of each of the phases is presented in Table II, III, and IV, respectively. The registration phase allows the new device to communicate with the admin device in a secure way and get enrolled in the system by giving its *ID*. The distributed key generation phase requests each device to re-generate its partial-secret (S_i) and share it with other devices *j* to re-compute private keys (D_i) and the network public key (K_{pub}). The request is normally generated by the admin device when a new device is enrolled or removed. Also, the admin can start this process from time to time to make the keys fresh. The secret-values are shared in such a way that the partial-secret 'S_i from a device *i* cannot be revealed to the other device j.

At the time of enrollment, the new device does not have its private key. So, in the 2nd phase, the commitment for a partial-secret from the new device to other devices is signed by the admin on the behalf of the new device behalf at the time of new key generation request. The third phase (session key generation and exchange) is based on the Diffie-Hellman ($DH-EKE$) protocol and uses private keys, IDs, and random secrets to generate and exchange session keys in a secure way. This phase generates symmetric keys (K_{ij}) between every pair of device *i* and device *j*. The symmetric key K_{ij} is used for the routine communication between device *i* and *j* in order to achieve confidentiality.

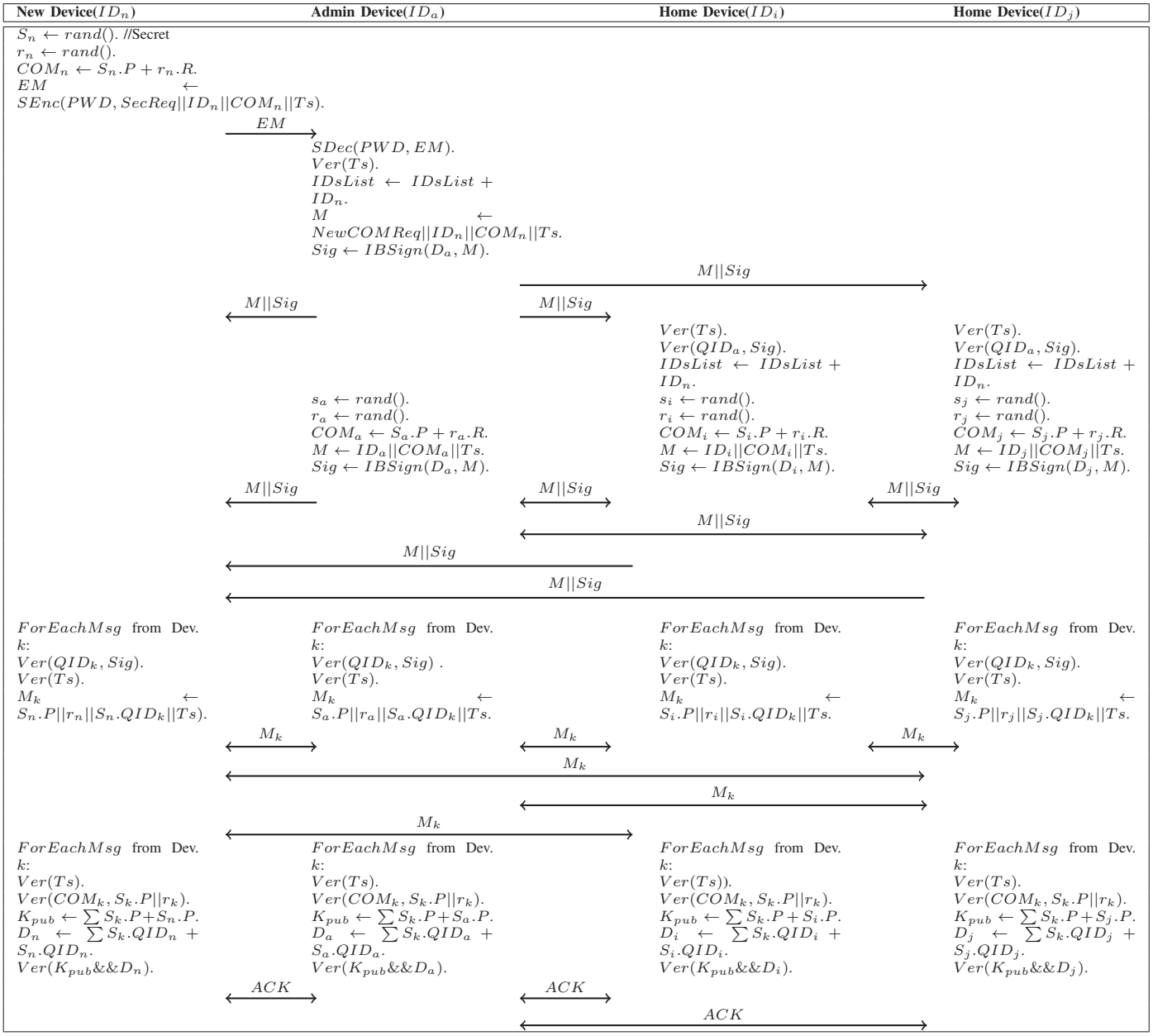
Initially, at the time of device registration and enrollment, we assume that either the admin device generates a password (*PWD*)or the homeowner/administrator select the initial password for the new device that would only be used for one-time session. In the registration phase, the new device request for enrollment in the network while hashing the *PWD*, *ID*, and the *time stamp*. The admin verifies the received information and sends all the public security parameter (P, R, K_{pub}, IDs, etc.) to the new device, encrypted by(*PWD*). The new device stores these public parameters for later use. The authentication of each of the message is achieved by the common *PWD*. The overall scenario of registration is depicted in Table II.

TABLE II. PROTOCOL PHASE-1: NEW DEVICE REGISTRATION AND ENROLLMENT



After the new device is registered, the device generates its partial-secret (S_n) and a random number (r_n). It sends the commitment (COM_n) in the form of ($S_n.P + r_n.R$) as a request to the admin to start distributed secret and keys re-generation process. The request-message is encrypted by the shared *PWD*, so it is pretty sure that it came from the new device. Since, other devices are not aware of the enrollment of the new device, so the admin broadcast the signed secret-regeneration request along with the new device commitment (COM_n). Thus, after receiving and verifying the request, every device re-generates its partial-secret, computes its commitment and broadcasts after signing it. Every device *i* receives the commitment of each of the other device k in the network as COM_k and verifies its signature. Once, all commitments are received, every device i reveals its secret-value to each of the other device *k* by diffusing it with the recipient ID_k (i.e., $s_i.QID_k$) and with the public parameter P (i.e., $s_i.P$). Also, when each device *i* received the revealing message from any other device *k*, the device i verifies the value by matching it with the corresponding COM_k. If it is not matched, then it means either the device k behaves maliciously and changed its partial-secret after the commitment or some intruder changed the value, being a man-in-the middle. In this way, the revealing value is authenticated. Once, every device *i* verified the partial-secret from each of other device *k* , the device *i* compute/recompute its private key as ($D_i \leftarrow \sum S_k.QID_i + S_i.QID_i$) and the

TABLE III. PROTOCOL PHASE-2: DISTRIBUTED IDENTITY-BASED KEYS GENERATIONS



network public key as ($K_{pub} \leftarrow \sum S_k.P + S_i.P$). Finally, all the devices verifies their computed keys by either encryption and decryption or signature and verification. After verification, each device transmit the *ACK* message to the admin. When, the admin receives all the *ACKs*, he confirms the successful generation of keys by all the devices through a broadcast message. The overall keys generation phase is drawn in Table III.

The PKC-based encryption is costly in terms of processing time especially when we have continuous streams of data to send. Hence, such routine communication among devices is encrypted and authenticated by symmetric keys. The symmetric keys are generated and exchange in phase-3 of the protocol as shown in Table IV. For initial communication from any device *i* to other device *j*–when there is not session key is exchange between–,the DH-based temporary key (TK) is used. TK for device *i* and other device *j* is generated by computing bi-linear map $\hat{e}(D_i, QID_j) = \hat{e}(D_j, QID_i)$. Next, the device *i* selects a random r_i and multiply it with the public parameters *P* and send it to device *j*, encrypted by *TK*. In response, the recipient device *j* sends back the random selected secret r_j multiplied with the *P*, encrypted by *TK*. Also, the device *j* computes the session key as ($K_{ij} \leftarrow r_j.r_i.P$) and sends the another random parameter (i.e., R_J) to device *i*, encrypt the parameter with K_{ij}. Later, after the verification and decryption, the device *i* computes the session key K_{ij} as (($K_{ij} \leftarrow r_i.r_j.P$). The device *i* also decrypt the other part (R_J)of the message by

TABLE IV. PROTOCOL PHASE-3: SESSION KEY GENERATION AND EXCHANGE BETWEEN TWO DEVICES

Device i (ID_i)		Device j (ID_j)
$TK \leftarrow \hat{e}\ (D_i, QID_j)$.		
$r_i \leftarrow rand()$.		
EM	$\leftarrow$	
$SEnc(TK, r_i.P\|\|Ts)$.		
	$EM\|\|Sig$ $\rightarrow$	
		$TK \leftarrow \hat{e}\ (D_j, QID_i)$.
		$M \leftarrow SDec(TK, EM)$.
		$Ver(Ts)$.
		$r_j \leftarrow rand()$.
		$K \leftarrow r_j.r_i.P$.
		$EM1$ $\leftarrow$
		$SEnc(TK, r_j.P\|\|Ts)$.
		$R_J \leftarrow rand()$.
		$EM2 \leftarrow SEnc(K, R_J\|\|Ts)$.
	$\leftarrow EM1\|\|EM2$	
$M1$	$\leftarrow$	
$SDec(TK, EM1)$.		
$Ver(Ts)$.		
$K \leftarrow r_i.r_j.P$.		
$M2 \leftarrow SDec(K, EM2)$.		
$R_I \leftarrow rand()$.		
EM	$\leftarrow$	
$SEnc(K, R_I\|\|R_J\|\|Ts)$.		
	EM $\rightarrow$	
		$M1 \leftarrow SDec(K, EM)$.
		$Ver(R_J\|\|Ts)$.
		$ACK \leftarrow SEnc(K, R_I\|\|Ts)$.
	$\leftarrow ACK$	
$M1 \leftarrow SDec(K, ACK)$.		
$Ver(R_I\|\|Ts)$.		
	Communication $\leftrightarrow$	

computed session key K_{ij}. Next, the device i transmits the random value $R_I.P$ along with received value R_J, encrypted by the session key K_{ij}, to the device j to confirm the K_{ij}. Remember, knowing the public parameter P and the product of $r.P$, you cannot extract r, so r_i would not be revealed to device j and vice versa. After the confirmation and acknowledgment, the session key K_{ij} is used for further communication between device i and device j.

The proposed protocol provides the authentication and integrity for each of the transmitted message by using digital signature, password hashes, and/or symmetric encryption. At the same time, the time stamp (Ts) is used in every communication to avoid replay attack. The partial-secrets are always shared by multiplying it to a public parameter, which makes impossible for any intruder to read the partial-secret as per discrete logarithmic assumption.

F. Secret and keys refreshment

It is also possible that the partial-secret a device is compromised. In this case, the overall network has a chance of recovery. The compromised node informs the admin to request for secret regeneration. In addition, to keep keys refreshing, the admin initiates the keys regeneration process from time to time. The session key between two devices can only be used for a fixed duration and needs to be refreshed after a fixed time slots. After session key expiration, the devices are instructed to re-initiate the session key exchange phase.

V. EVALUATION AND PROTOCOL SIMULATION

We evaluated the security of the overall model theoretically, as well as formally using security validation tool 'AVISPA' [27] by implementing the protocol in high-level protocol specification language (HLPSL) [28]. We considered most of the known attacks popular in the HAN communication, including *MITM* attack, replay attack, non-repudiation attack, brute force attack, compromise session key or change of authorized keys attack, attacks on time stamp and message integrity, and compromise key attack.

In addition, we examined the feasibility of the proposed security model by simulating it on IoT devices, considering them as smart home devices.

A. Security Evaluation

1) Informal Security Analysis: The proposed protocol is secured from popular known attacks that can be launched on HAN network.

- Resilience against Sybil attacks: Sybil attack is performed by a malicious device to add more fake devices. Since, we used initial passwords generated by an admin for new node authentication, so it is not possible a new malicious device to be part of the home network. Also, the new device initially communicates to other devices for keys generation through the admin device. Moreover, each transaction is authenticated by either password, identity-based signature (*IBSign*), or the commitments, so none of the fake devices has the secret S to generate his private key to sign and pretend.
- Resistance to a man-in-the-middle (*MITM*) attack: Every transmission of the protocol is either encrypted by symmetric (*SEnc*()), authenticated by Private key D_i, or verified by the commitment process. Secret-values are always sent hidden. So, the *MITM* attack is hard.
- Resistance to replay attack: Every communication between devices uses a distinct time stamp (Ts) that keeps changing based on the current time. The Ts is also part of the signature and encryption. Thus, if the communication is replayed, the time stamp cannot be changed.
- Resistance to eavesdropping: The secret values are sent hidden (after multiplying to a public parameter). So the intruder cannot extract the partial-secret from the product due to the fact of discrete logarithmic problem. Furthermore, the routine communication is encrypted by the symmetric keys, the intruder cannot read the message or analyze the contents within the message.
- Resistance to brute force attack: IBC and ECC use a large key size that makes impossible for an attacker to apply brute force attack.
- Compromised key and secret: Partial secret from every device is communicated after multiplying it with a large size public parameter.Thus, it is not possible

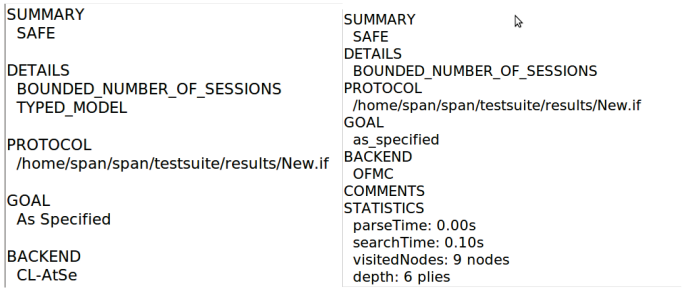


Fig. 1. The result of AVISPA Verification usinf CL-AtSe and OFMC mode

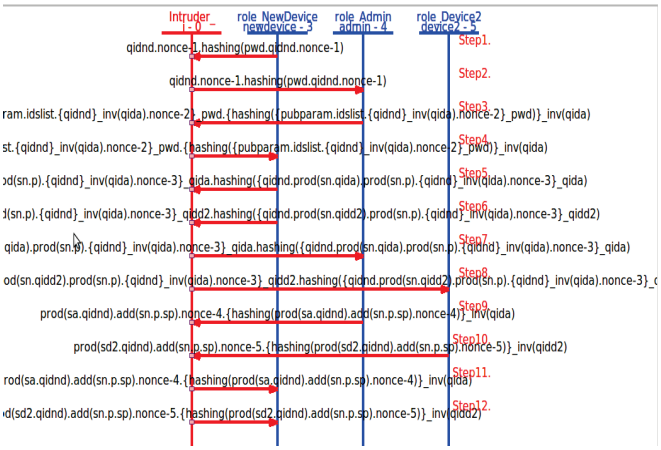


Fig. 2. Protocol implementation expecting intruder in-between.

Protocol implementation expecting intruder in-between.

to extract the secret from the product as by discrete logarithmic assumption.

- Resistance to change of IDs attack: No one can change the device-ID as the private key used for decryption and signature directly dependent on the secret value and the *ID*. So changing the *ID* while in transmission is easily identified. It does not affect the security as the signature/stamp always authenticates the device who sent the message.

2) Formal Security Validation using Software Tool: AVISPA [27] is used as a software and verification tool for security protocol verification. We implemented the protocol in AVISPA using high-level protocol specification language (HLPSL). We mainly focused on replay attack, non-repudiation attack, any type of man-in-the-middle attack, and attack on data integrity, secrecy, and authentication. we used Dolev-Yao model [29] for verification, which places an intruder (*I*) in-between every communication. The intruder is given full access to all the public information, its own keys, and the transmission channel. Two generally practiced AVISPA verification models called 'On the Fly Model-Checker (OFMC)' and 'Constraint-Logic-based Attack Searcher (Cl-AtSe)' are used for security analysis.

AVISPA proves that the protocol is safe from all the mentioned attacks using two of his security analysis models

TABLE V. TIME (MS) CONSUMED BE THE DEVICES ON BASIC SECURITY AND IBC OPERATIONS

	Processing time (ms)	
Operations	**Admin device (desktop PC)**	**Home device (Raspberry Pi)**
Comparing IBC-Parameter's	0.001 ms	0.001 ms
IBC-Parameter's Addition	0.001 ms	0.002 ms
IBC-Parameter's Multiplication	6.041 ms	39.694 ms
Bilinear Mapping	8.595 ms	81.625ms
MaptoPoint Function	18.568 ms	128.194ms
IBC-Commitment Generation	16.467 ms	80.057ms
IBC-Commitment Verification	7.852 ms	40ms
Symetric Encryption (256 B)	0.001 ms	0.013 ms
Symetric Decryption (256 B)	0.002 ms	0.012 ms
IBC Signature (256 B)	12.27 ms	79.241 ms
IBC Sign(256 B) Verification	21.952 ms	203.344ms

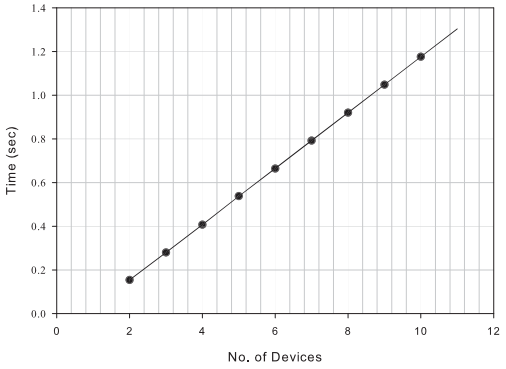


Fig. 3. Processing time: registration and new-device enrollment (phase-1)

Registration and new-device enrollment time

OFMC and CL-AtSe. Figure 1 shows the results of the security verification, and the intruder's simulation is depicted by Figure 2. We can see every messages from the new device to the admin, admin to new device, and new device to other device and vice-versa, are intersected by the intruder. Although, the intruder is in between the communication, but still he is unable to break the security of the proposed model.

B. System Implementation and its Feasibility on IoT Devices

In order to check the feasibility of the model, we implemented the system on a desktop machine and Raspberry Pi devices. Our model assumes that the admin device is more powerful than the other home devices. Hence, the role of the admin device is implemented on Linux-based Intel Xeon (R) desktop machine with 16 CPU cores of 1.2 GHz and 65 GB shared memory. The device can run two parallel threads on each of these sixteen cores. The role of other home devices are implemented on IoT machines (Rasberry Pi3 model B). Each Rasberry Pi device is equipped with four CPU cores of 1.2 GHz and 1GB SDRAM and can only run one thread per core. For basic identity based cryptographic operations, we used Ben Lynn PBC library [30], whereas, the mathematical operations on large-size parameters are perform by GMP library [31]. The symmetric encryption and hashes are performed by AES-CBC-256 cipher and sha-256 respectively.

Initially, we started to observe the cost of basic operations

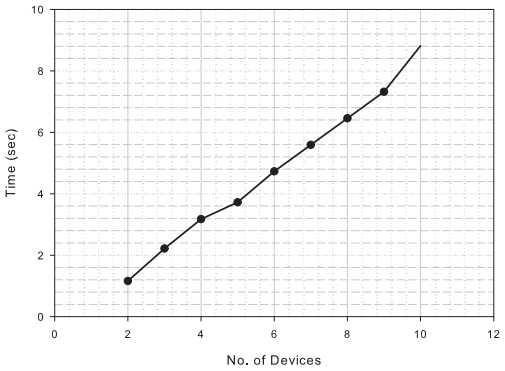


Fig. 4. Processing time: partial-secrets and ID-based pub-lic/private keys generation (phase 2)

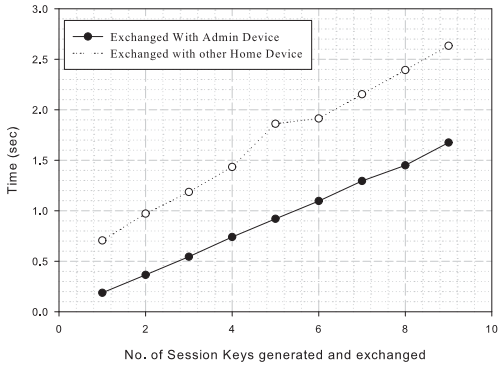


Fig. 5. Processing time: session key exchange with the admin device as well as with the other home device j (phase 3)

used by IBC on a Rasberry Pi device (assumed it as a smart home device) and on desktop machine (considered as an admin device). We noticed that the large-size (256 bytes) element multiplication, *MaptoPoint* function, and the bi-linear mapping take more time, whereas, the addition and comparison requires negligible time on both type of devices. Among cryptographic operations, the signature verification occupies larger cpu time-slot as it performs two bi-linear mappings as a most expensive operation. Since, the IoT board has a limited power and resources, so it definitely takes a longer time for each of these operations as compared to desktop machine (admin device). The comparison of the time consumed by both type of devices for various operations are shown in Table V. As the home device need less than two hundreds and fifty milliseconds for each of the operations, it proves the applicability of IBC on IoT devices.

Next, we monitored the time consumption of each phase of the designed protocol corresponding to the increasing number of devices. Fig. 3 depicts the time consumption on registration phase, Fig. 4 presents the same measurements for the secret-value sharing and keys computing phase, Fig. 5 compares the time consumed by an increasing number of devices for the session key exchange with the admin device and some other

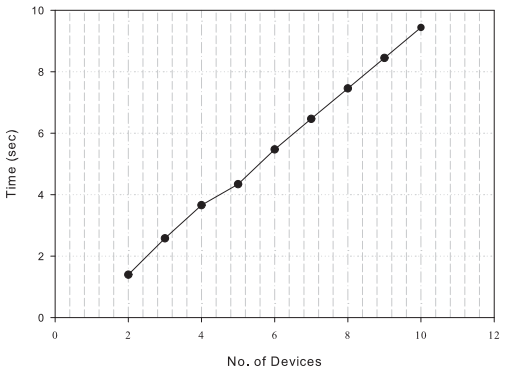


Fig. 6. Over protocol running time

home device, and Figure 6 summarizes the overall protocol running time from a new device enrollment to its session key exchange with each of the home device. The registration phase only allows the new device to communicate to the admin device irrespective of the number of devices in the network. Moreover, In registration phase, the new device has to compute the point on the elliptic curve (i.e., QID) corresponding to the *ID* using *MapToPoint* function for each of the existing home devices. Thus, the use of *MapToPoint* function will increase with the increasing number of devices, which results in linear increase in the registration time as shows in the Fig. 3. Also, the proposed model performs the computation in parallel (through threads) in the keys generation phase, especially when a device receives commitments and the revealed partial-secrets from multiple devices at the same time. This parallelism dramatically increases the performance of the system, proved by the linear rise in the processing time. Remember, corresponding to a linear increase in the number of devices, there is an exponential rise in the number of messages exchanged, but the effect on the processing time remains linear, as shown in Fig. 4. However, in the session key exchange phase, when a device generates a session key with other n devices, we assumes the key exchange is done sequentially (so, parallelism cannot be achieved). Thus, the time is increasing function of the number of session keys generated. Fig. 5 compares the rise in the time corresponding to the increasing number of session keys generated and exchanged with the admin device and the other home device. Obviously, in case of home device to other home device, the session key exchange takes more time than the exchange with the admin device (as admin is more powerful).

Finally, we analyzed the overall protocol running time. We know that, with the increasing size of the network, the message communication exponentially increases because of the exchange of secret-value from each device i to other device j. However, with the ability of parallel and efficient implementation of the security model, the corresponding increase in the processing time is almost a linear, as shown by a graph in Fig. 6. In addition, the overall time required to run the protocol is not more than 10 seconds with 10 home devices in the network. With this network size, it takes 2.6 second for a device to generate and share session keys with other home devices, the registration requires 1.3 second, and the public-

private keys re-computation needs 8 seconds only.

VI. Conclusion

Security is a vital requirement for every smart system. The best security system also takes care of the resource limitations of the system. Thus, in this paper, we proposed a security model for a smart home that considered the power and storage limitations of HAN devices. The proposed security mechanism is based on IBC with distributed creation of keys. Unlike IBC, it does not require central KGC to generates keys for every device and central storage of main secret. Only the device identity (ID) is needed to encrypt the message and verify the signature. A protocol is proposed that allows new device to be enrolled in to the network, shares secrets among devices to generate and update their private keys and the network public key. The protocol uses the IBC-based signatures for authentication and DH key exchange for every pair of devices to communicate securely using a shared key. The security of the protocol is verified using mathematical analysis and using security verification tool called AVISPA. Also We have tested the feasibility of the system in terms of efficiency by implementation the model on IoT-based boards by considering them as home devices.

Acknowledgment

The current work is supported by Atlantic Innovation Fund and MITACS (IT24468).

References

[1] D. J. Malan, M. Welsh, and M. D. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*. IEEE, 2004, pp. 71–80.

[2] M. M. Rathore, A. Ahmad, A. Paul, and S. Rho, "Urban planning and building smart cities based on the internet of things using big data analytics," *Computer networks*, vol. 101, pp. 63–80, 2016.

[3] M. M. Rathore, A. Paul, W.-H. Hong, H. Seo, I. Awan, and S. Saeed, "Exploiting iot and big data analytics: Defining smart digital city using real-time urban data," *Sustainable cities and society*, vol. 40, pp. 600–610, 2018.

[4] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Workshop on the theory and application of cryptographic techniques*. Springer, 1984, pp. 47–53.

[5] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Annual international cryptology conference*. Springer, 2001, pp. 213–229.

[6] ——, "Identity-based encryption from the weil pairing," *SIAM journal on computing*, vol. 32, no. 3, pp. 586–615, 2003.

[7] R. Sakai and M. Kasahara, "Id based cryptosystems with pairing on elliptic curve." *IACR Cryptology ePrint Archive*, vol. 2003, p. 54, 2003.

[8] C. Cocks, "An identity based encryption scheme based on quadratic residues," in *IMA International Conference on Cryptography and Coding*. Springer, 2001, pp. 360–363.

[9] L. Ducas, V. Lyubashevsky, and T. Prest, "Efficient identity-based encryption over ntru lattices," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2014, pp. 22–41.

[10] S. McCarthy, N. Smyth, and E. O'Sullivan, "A practical implementation of identity-based encryption over ntru lattices," in *IMA International Conference on Cryptography and Coding*. Springer, 2017, pp. 227–246.

[11] L. Yang, C. Ding, and M. Wu, "Establishing authenticated pairwise key using pairing-based cryptography for sensor networks," in *2013 8th International Conference on Communications and Networking in China (CHINACOM)*. IEEE, 2013, pp. 517–522.

[12] C. Gentry, "Practical identity-based encryption without random oracles," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2006, pp. 445–464.

[13] G. De Meulenaer, F. Gosset, F.-X. Standaert, and O. Pereira, "On the energy cost of communication and cryptography in wireless sensor networks," in *2008 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*. IEEE, 2008, pp. 580–585.

[14] P. Szczechowiak and M. Collier, "Tinyibe: Identity-based encryption for heterogeneous sensor networks," in *2009 International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. IEEE, 2009, pp. 319–354.

[15] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the internet of things," *Future Generation Computer Systems*, vol. 49, pp. 104–112, 2015.

[16] Y. Mao, J. Li, M.-R. Chen, J. Liu, C. Xie, and Y. Zhan, "Fully secure fuzzy identity-based encryption for secure iot communications," *Computer Standards & Interfaces*, vol. 44, pp. 117–121, 2016.

[17] H. Nicanfar, P. Jokar, and V. C. Leung, "Efficient authentication and key management for the home area network," in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 878–882.

[18] H. Nicanfar, P. Jokar, K. Beznosov, and V. C. Leung, "Efficient authentication and key management mechanisms for smart grid communications," *IEEE systems journal*, vol. 8, no. 2, pp. 629–640, 2014.

[19] R. H. Jacobsen, S. A. Mikkelsen, and N. H. Rasmussen, "Towards the use of pairing-based cryptography for resource-constrained home area networks," in *Digital System Design (DSD), 2015 Euromicro Conference on*. IEEE, 2015, pp. 233–240.

[20] Q. Gao, "Biometric authentication in smart grid," in *Energy and Sustainability Conference (IESC), 2012 International*. IEEE, 2012, pp. 1–5.

[21] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[22] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust threshold dss signatures," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1996, pp. 354–371.

[23] R. Gennaro, S. Goldfeder, and A. Narayanan, "Threshold-optimal dsa/ecdsa signatures and an application to bitcoin wallet security," in *International Conference on Applied Cryptography and Network Security*. Springer, 2016, pp. 156–174.

[24] L. Nguyen, "Partially interactive threshold rsa signatures," *Oxford computing technical report*, 2005.

[25] M. M. Rathore, E. Bentafat, and S. Bakiras, "Smart home security: a distributed identity-based security protocol for authentication and key exchange," in *2019 28th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2019, pp. 1–9.

[26] J. C. Choon and J. H. Cheon, "An identity-based signature from gap diffie-hellman groups," in *International workshop on public key cryptography*. Springer, 2003, pp. 18–30.

[27] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani *et al.*, "The avispa tool for the automated validation of internet security protocols and applications," in *International conference on computer aided verification*. Springer, 2005, pp. 281–285.

[28] D. Von Oheimb, "The high-level protocol specification language hlpsl developed in the eu project avispa," in *Proceedings of APPSEM 2005 workshop*. APPSEM'05, Tallinn, Estonia, 2005, pp. 1–17.

[29] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[30] B. Lynn, "Pbc library manual 0.5. 11," 2006.

[31] T. Granlund *et al.*, *GNU MP 6.0 Multiple precision arithmetic library*. Samurai Media Limited, 2015.