SN Computer Science (2022) 3: 364
https://doi.org/10.1007/s42979-022-01269-9

ORIGINAL RESEARCH

A Formally Validated Authentication Algorithm for Secure Message Forwarding in Smart Home Networks

Vincent Omollo Nyangaresi[1]

Received: 4 December 2021 / Accepted: 24 June 2022 / Published online: 9 July 2022

Abstract

The many devices connected in smart homes increase the attack surfaces from which adversaries can invade the network. In addition, majority of these smart devices have numerous vulnerabilities that can be exploited to wreck havoc in smart homes. As such, a myriad of security schemes have been presented based on technologies such as bilinear pairing operations, public key infrastructure, blockchains and elliptic curve cryptosystems. However, some of these protocols are not robust against conventional smart home attacks. In addition, some of the deployed techniques inadvertently result in excessive processing at the smart devices. It is, therefore, imperative that provably secure protocols be developed to offer efficiency and sufficient protection to the exchanged packets. In this paper, an elliptic curve symmetric key-based algorithm for secure message forwarding is presented. Formal security verification is executed using the Burrows–Abadi–Needham (BAN) logic which demonstrates strong mutual authentication and session negotiation among the communicating entities. In addition, the informal security analysis carried out shows the robustness of this scheme under the Canetti–Krawczyk threat model. Moreover, it is relatively efficient in terms of storage, communication, energy and computation requirements.

Keywords Algorithm · Attacks · Authentication · Elliptic curve · IoT · Privacy · Security · Symmetric key

Introduction

Smart homes offer users enhanced quality of life, convenience as well as comfort through the exchange of real-time data. A typical Smart Home (SH) can provide intelligent and automated remote monitoring of the various activities [1]. It may comprise of remote users, registration authority, gateways and smart devices [2]. The smart devices in these networks may include refrigerators, cameras, television sets, motion sensors, lighting systems, doorbells, voice assistants and thermostats [3]. The smartness in these devices is reflected in their ability to monitor and control activities, as well as offering the required support to the users. As pointed out in [4], the smart homes can potentially result in energy efficiency and, hence, reduction in power bills. Another significant technique for improving safety as well as energy efficiency is through the incorporation of Artificial Intelligence (AI). Here, the big data generated by sensors and other smart devices can be deployed to train machine learning algorithms to distinguish between normal and anomalous smart home activities. Essentially, any network traffic or events that are flagged as anomalous can be indicators of possible attacks in these smart homes. Once attacks are detected, various actuators can be activated to protect the smart homes. Evidently, AI can be deployed to offer safety, boost productivity as well as the well-being of users [5]. It is also possible to deploy AI for home users' behavior analysis, which can help predict their needs and optimize both device and resource usage. To accomplish this, machine algorithms such as decision trees, support vector machines and neural networks can be trained and deployed [6]. Thereafter, any behavioral change may be construed to imply anomalies.

Although smart homes have numerous benefits, many threats, vulnerabilities and attacks lurk in these networks. This may be attributed to the massive number of heterogeneous connected devices which increase the surfaces from which attacks can be launched. In addition, majority of the connected devices do not adhere to security practices as well as standards of typical computing systems. As such, they can be hacked and deployed to spread malware. The message exchange between the smart devices and the remote

✉ Vincent Omollo Nyangaresi
vnyangaresi@tmuc.ac.ke

[1] Faculty of Biological and Physical Sciences, Tom Mboya University College, Homabay 40300, Kenya

users is via the public internet and hence is open to several attacks [7]. As such, smart homes inadvertently create new privacy, security and authentication challenges [8]. The possible attacks in this environment include unauthorized access, data forgery, tampering, impersonation, Distributed Denial of Service (DDoS) and offline-guessing attacks [9, 10]. Since most of the communication is through the gateways, this presents a single point of failure. As such, any successful attack on this centralized architecture can lead to user privacy leaks, device malfunction and even harm to the home occupiers [11].

Based on the discussion above, it is evident that smart home devices have become pervasive with increased connectivity. As such, vulnerabilities in any of these devices may lead to unauthorized access to entire home networks. For instance, any adversarial access to the smart home heating system may raise the house temperature. This may lead to fires, smart home hardware failures or even death. Therefore, it is clear that the security and privacy of the users need to be protected [1]. Failure to accomplish this may result in slow uptake of this important technology. Strong mutual authentication presents one of the most promising mechanisms of protecting against the aforementioned attacks [12]. For anomaly detection, there is need for highly efficient and accurate detection models capable of adapting to novel attacks, increasing number of devices as well as threat landscape. Blockchain technology has been proposed as another most effective way of enhancing privacy, security and transparency in smart homes [13]. Unfortunately, majority of the smart devices are resource constrained [14] to handle the high processing required in blockchains.

Research Contributions

Smart homes enhance quality of life and convenience to its users. However, the sensed data from the smart home devices are relayed to remote users over public channels. As such, the exchanged data are exposed to numerous threats and attacks. Due to the existence of many interconnected devices, any successful compromise of a single smart device can have devastating ripple effect to all other devices. Although proper authentications can prevent these attacks, the resource constrained nature of smart devices such as sensors limit the deployment of strong cryptographic primitives. In addition, majority of the conventional smart home protocols only mutually authenticate smart devices to the servers but fail to execute authentication among the devices. To this end, this paper makes the following major contributions.

- Elliptic curve cryptography is amalgamated with symmetric key and one-way hashing operations to develop a scheme that thwarts most of the conventional smart home attacks.
- The algorithm executes mutual authentication and key negotiation among the smart devices, in addition to authenticating the smart devices to the trusted authority. Here, the trusted authority cannot derive session keys negotiated among the smart home devices. In addition, other devices can never derive session keys established between a particular smart device and the trusted authority. As such, privileged insider, impersonation, session hijacking, denial of service, man-in-the-middle and stolen verifier attacks are prevented.
- Session delay tolerance and time-stamping are incorporated in the generated security tokens to avert packet replay attacks.
- Extensive formal security validation is executed on this scheme using BAN logic, which shows the existence of strong mutual authentication and session key negotiation among the communicating entities.
- Informal security analysis is carried out to demonstrate the robustness of this scheme under the Canetti–Krawczyk threat model.

As discussed in "Security Analysis" and "Performance Evaluation", these approaches advance the state of the art authentication protocols in two ways. First, the deployed cryptographic primitives are fairly lightweight, and hence the developed scheme boosts network efficiency. Second, the negotiated session keys help in establishing secure communication channels among smart home devices as well as with the trusted authority.

Paper Organization

In "Related Work", the authentication and key agreements protocols that have been presented to curb security and privacy issues in smart homes are discussed, including their shortcomings. "Mathematical Preliminaries", "Security Goals and Requirements" and "Motivation" present mathematical preliminaries, security goals and requirements, as well as the motivation of this work, respectively. This is followed by the description of the proposed protocol in"The Proposed Scheme". On the other hand, the security and performance analyzes are presented in "Security Analysis" and "Performance Evaluation", respectively. Finally, "Conclusion" concludes the paper and offers some insights on future work in this particular domain.

Related Work

Security challenges in smart homes have prompted a lot of research, resulting in numerous schemes for anomaly detection and authentication. For example, authors in [15] have presented a scheme to distinguish between normal

and anomalous activities. Similarly, the techniques in [10] and [16] deploy Hidden Markov Model (HMM) and are trained on sensor data. On the other hand, the approaches in [17] and [18] utilize Bayesian networks in their anomaly detection. In addition, flow-based attack detection scheme is presented in [19]. Using inbound and outbound sensor packets, a neural network-based anomaly detection scheme is developed in [20]. The attacks detected this way include Man-in-the-Middle (MitM) and Distributed Denial of Service (DDoS). However, authors in [21] have utilized a combination of machine learning and statistical techniques for behavioral analysis in smart homes. On the other hand, artificial neural networks and support vector machines have been utilized in [22] for network intrusion detection.

Although anomaly detection techniques play a significant role in securing smart homes, they are mainly concerned with activity detection. As such, they cannot execute the required access control in this environment. To address this shortcoming, many authentication and key negotiation protocols have been introduced based on various techniques. For instance, a three-factor authentication scheme based on Elliptic Curve Cryptography (ECC) is developed in [23] while a two-factor user authentication protocol is presented in [24]. However, the scheme in [23] cannot mutually authenticate all network entities and fails to uphold forward key secrecy. On the other hand, the protocol in [24] is susceptible to both stolen user device and insider attacks [25]. To offer decentralization [26] and prevent single point of failure in schemes based on centralized architectures, many protocols based on blockchain technology have been proposed in [8] and [27–31]. The security goals attained by these schemes include data integrity, availability, authentication, access control, user and data privacy. Unfortunately, these schemes have extensive computation requirements which are detrimental for majority of resource-limited smart home devices [32]. In addition, the protocol in [31] deploys a pair of private and public keys which further require high execution time.

The scheme in [33] can potentially address performance issues in blockchain-based techniques due to its reduced registration overheads. However, it is vulnerable to stolen smart device attacks. Similarly, the protocol in [2] is vulnerable to stolen smart device, impersonation, offline password guessing, privileged insider and packet replay attacks. In addition, it fails to provide anonymity and the usage of simple password exposes it to shoulder-surfing attacks [34]. To address some of the issues in [2], a two-factor authentication technique is developed in [25]. However, this protocol is still susceptible to session key disclosure and impersonation attacks. In addition, it has excessive communication and computation overheads [35]. On the other hand, the failure to incorporate random nonces and timestamps in [36] renders it susceptible to replay attacks. Similarly, the schemes in [37] and [38] are vulnerable to MitM and DoS attacks.

A scheme for remote user authentication is introduced in [39], while an identity-based security framework is presented in [40]. However, the protocol in [39] fails to offer forward key secrecy and cannot withstand stolen device, session key compromise and replay attacks [25]. On the other hand, identity-based scheme in [40] has key escrow issues [41]. To boost performance in smart home networks, a lightweight authentication protocol is developed in [35]. However, this scheme cannot provide integrity protection of the exchanged messages, leading to DoS [34]. Although the scheme in [42] is lightweight and hence applicable in most smart home devices, it cannot provide perfect forward key secrecy. This is because any compromise of its long terms key can facilitate adversarial computation of the session keys. On the other hand, the context-aware authentication approach developed in [43] has excessive execution time. Similarly, the multi-factor mutual authentication protocol in [44] has high computation overheads due to the bilinear pairing operations [45]. On its part, the protocol in [46] offers mutual authentication only between the devices and the server.

Mathematical Preliminaries

In this section, the cryptographic primitives for one-way hashing are provided, together with their collision resistant properties. The mathematical formulations for the deployed elliptic curve cryptography can be found in [7]. The second part of this section provides the mathematical preliminaries for symmetric key primitives, as elaborated below.

Symmetric Primitives

Symmetric algorithms (SA) comprise of symmetric key encryption algorithms (SKEA) and cryptographic hash functions. Here, SKEA can be stream or block ciphers. In the former, encryption is through the combination of plaintext and pseudo-random sequences. Each stream cipher takes key δ and initial value σ to produce pseudo-random key stream that is utilized in data encryption and decryption through bitwise exclusive or (XOR) operations.

Taking l as the block length and m as the cipher key size, then a block cipher is a transformation:

$$F : \mathbb{C}_2^l X \mathbb{C}_2^m \rightarrow \mathbb{C}_2^l, \tag{1}$$

where $F_m \underline{\underline{\text{def}}} F(., m)$ is a bijection of $\mathbb{C}_2^l$ for $m \in \mathbb{C}_2^m$.

Suppose that $y = F_m(\mathbb{r})$; then r becomes the plaintext, while m and y are the key and cipher-text of $\mathbb{r}$ under key m, respectively.

One-Way Hashing

A cryptographic hash function denotes a map F whose input is a string of arbitrary length. It then transforms this input string into an output string of fixed length l. Every one-way hash function:

(a) Takes argument a of arbitrary length and outputs $h(a)$ that is of some fixed length l bits.
(b) Given that b is the image of h, it is computationally infeasible to find message x such that $h(a) = b$. This one-way property is referred to as the pre-image resistance.
(c) Given a in the domain of h and $h(a)$, it is computationally cumbersome to find message $a' \neq a$ such that $h(a') = h(a)$. In terms of F, this can also be written as $F(a') = F(a)$. This one-way property is referred to as the second pre-image resistance.

Suppose that $\mathbb{N}$ is a set of all integers and a binary alphabet is denoted as $\sum = \{0, 1\}$. Then, for $l \in \mathbb{N}$, the set of all binary strings of length l is denoted as $\sum^l$. On the other hand, the set of all strings of arbitrary length is expressed as Σ^*. Let F be a function whose domain and range are denoted as $\acute{H} = \Sigma^*$ and $\bar{G} = \sum^l$, respectively. Let us consider only inputs of bit length $\acute{k}(l)$, where $\acute{k}(l)$ is a function satisfying the condition $\acute{k}(l) > l$. With these definitions, the pre-image and second pre-image resistances can be written as follows:

(d) A one-way hash function h is a function whose domain $\acute{H} = \sum^{\acute{k}(l)}$ and range $\bar{G} = \sum^l$ satisfy the following conditions:

Pre-image resistance: Suppose that x is uniformly chosen in $\acute{H}$ and let $\tilde{\tilde{A}}$ be an attacker, who on inputting $h(x)$, he utilizes time $\leq$ t to output $\tilde{\tilde{A}}(h(x)) \epsilon\ \acute{H}$. For every attacker $\tilde{\tilde{A}}$:

$$\underset{x \in' H}{P_r}\left\{h\left(\tilde{\tilde{A}}(h(x))\right) = h(x)\right\} < \theta. \tag{2}$$

In (2), the probability is taken over some stochastically selected attacker $\tilde{\tilde{A}}$.

Second pre-image resistance: Suppose that x is uniformly chosen in $\sum^{\acute{k}(l)}$ and let $\tilde{\tilde{A}}^*$ be an attacker, who on inputting x, he utilizes time $\leq$ t to output $x^* \in \acute{H}$ and $x^* \neq x$. For every attacker $\tilde{\tilde{A}}^*$:

$$\underset{x \in' H}{P_r}\left\{\tilde{\tilde{A}}^*(x) = h(x)\right\} < \theta. \tag{3}$$

In (3), the probability is taken over some stochastically selected attacker $\tilde{\tilde{A}}$. The pre-image and second pre-image resistance conditions are significant when t/θ is large and $t/\theta \leq 2^n$

(e) A collision-resistance hash function is a function h that satisfies condition (a), is one way (satisfies conditions (b) and (c)) and it is infeasible to find two distinctive messages that produce the same hash value.

Security Goals and Requirements

Massive and sensitive information flows in smart home networks. It is therefore paramount that proper security and privacy measures be instituted before the remote users can begin accessing data in these smart home devices. In light of this, the following security goals and requirements are pursued in this paper:

Backward and Forward Key Secrecy

An attacker located between the remote user and the smart home devices may have the ability of capturing the session key deployed for traffic encryption. As such, it should be infeasible for an adversary to derive the session key used for the previous and subsequent authentication sessions based on the current session key.

Mutual Authentication

To ensure that only legitimate entities access the smart home networks, all the entities initializing any connection requests should have their identities verified before any such access is granted.

Resilience Against Attacks

To ensure strong security in smart home networks, it should be infeasible for an adversary to launch typical smart home network attacks such as MitM, packet replays, privileged insider, session hijacking, DoS, impersonation, offline dictionary and stolen verifier.

Confidentiality

The smart home network-based sensors collect high volumes of sensitive and private data. As such, only authorized and fully authenticated entities should be allowed to access the sensed data.

Integrity

During data transmission across the public networks, the communicating parties should ensure that no malicious modifications are made to the data.

Availability

Within the smart home network, the remote users should be able to access the sensed data anywhere and at any time.

Scalability

It should be easy for the smart home to support additional smart devices without compromising the underlying security and privacy architecture.

Motivation

The transmission of senses data to the remote users over public wireless channels opens up the smart home networks to numerous attacks. Any successful attack on the smart devices may lead to malfunction of other devices or malicious control of the smart home devices. For instance, hacked heating systems may result in temperature increments that can endanger the lives of home occupiers. In addition, due to the interconnectivity of the smart home devices, any successful compromise of a single device can lead to privacy leaks and attacks on other devices. Although many protocols have been developed for smart home authentications, majority of them only authenticate the smart devices to the servers. As such, authentication among the smart home devices is largely ignored. This is detrimental as it facilitates attacks on other systems using vulnerabilities in other devices. In addition, the security solutions developed to address these issues are either inefficient or have security holes that can be exploited. Therefore, the inefficiency, privacy and security holes in most of the current authentication protocols need urgent solution.

The Proposed Scheme

The network entities in the proposed algorithm include the Smart Home Owners (SHOs), Trusted Authority (TA), the Smart Home Devices (SHDs) and the Mobile Devices (MDs) through which remote users interact with their SHDs. As shown in Fig. 1, the smart home devices may include smart doors, TV, thermostats, cameras, lighting systems and refrigerators.

All smart home devices as well as mobile devices are registered at the trusted authority before they are permitted to communicate with each other. After registration, all MDs and SHDs have to execute mutual authentication with the TA and negotiate a session key. Similarly, the MDs and the SHDs must also mutually authenticate each other before exchanging any messages. As such, the proposed scheme is highly scalable to support additional smart devices within this authentication architecture. The communication channel between the MDs and the SHDs may be cellular network such as the Fifth Generation (5G). Table 1 presents the symbols used in this paper together with their brief descriptions.

In terms of the actual execution, the proposed algorithm comprises of the registration phase, SHDs-TA authentication, MD-TA authentication, and SHDs-MD authentication. The sub-sections below describe these phases in more details.

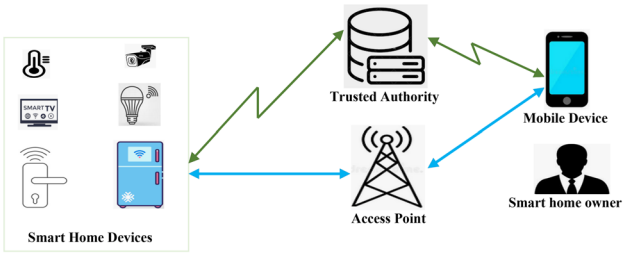

Fig. 1 Network architecture

Table 1 Notations and their descriptions

Symbol	Description
TA_{SV}	TA secret value
ID_{SHD}	Smart home device unique identity
R_i	Random number *i*
T_i	Timestamp *i*
AG_{EC}	Additive group implemented by an EC
G	Generator of AG_{EC}
$\mathbb{Q}_{ST}$	Session key between SHD and TA
$\mathbb{Q}_{MT}$	Session key between MD and TA
$\mathbb{Q}_{SM}$, $\mathbb{Q}_{MS}$	Session key between SHD and MD
ID_{MD}	Mobile device unique identity
E_K	Encryption using key *K*
D_K	Decryption using key *K*
ΔT	Delay tolerance
$\|\|$	Concatenation operation
$\oplus$	XOR operation
Z_q	Finite field over *q*

Registration Phase

In this phase, the smart home devices as well as the remote user mobile devices are registered at the trusted authority.

Basically, the MD and SHD registration procedures are the same and hence only the SHD registration is described here. This is a three-step process as detailed below.

Step 1: The smart home device selects ID_{SHD} as its unique identity, which it forwards to the trusted authority over secure channels.

Step 2: Upon receiving ID_{SHD}, the trusted authority generates random number R_1 which it utilizes to derive $A_1 = h(R_1 \| TA_{SV} \| T_1 \| ID_{SHD})$, $A_2 = A_1 \times G$, $A_3 = R_1 \oplus h(TA_{SV})$, $A_4 = h(R_1 \oplus h(TA_{SV}) \| A_2)$ and $A_5 = A_4 \times G$. Finally, the trusted authority stores parameter set $\{ID_{SHD}, T_1, A_3, A_5\}$ before sending A_2 to the smart home device as shown in Fig. 2.

Step 3: After getting A_2 from the TA, the SHD stores it in its memory for use in the authentication phase.

Device–TA Authentication Phase

In this phase, both the SHD and the MD mutually authenticate themselves to the TA. After successful authentication, they negotiate session keys between themselves and the TA. This is a five-step process as discussed below.

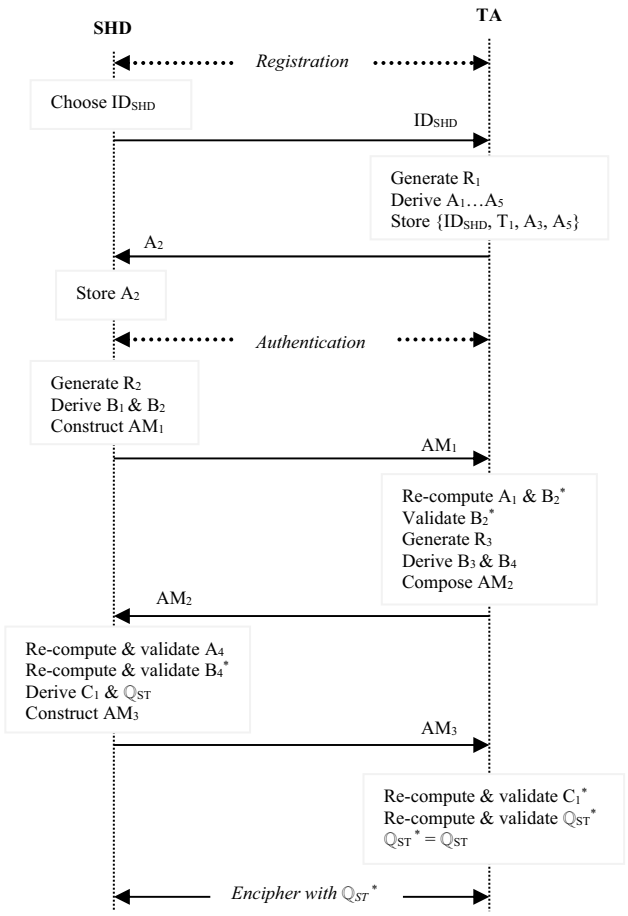

Fig. 2 Registration and device–TA authentication

Step 1: The SHD generates random number R_2 that it uses to compute security parameters $B_1 = R_2 \times G$ and $B_2 = h(B_1 \| R_2 \times A_2)$. Next, it sends these two parameters in authentication message $AM_1 = \{B_1, B_2\}$ to the TA for verification as shown in Fig. 2.

Step 2: On receiving parameters B_1 and B_2, the TA deploys T_1, TA_{SV} and A_3 to compute A_1 and utilize it to confirm the validity of both B_1 and B_2. To accomplish this, parameter B_2^* is computed as $B_2^* = h(B_1 \| A_1 \times B_1)$. Next, it checks if $B_2^* \stackrel{?}{=} B_2$ such that the session is terminated when this verification fails. Otherwise, it generates random number R_3 that is used to derive parameters $B_3 = R_3 \times G$ and $B_4 = h(B_2^* \| R_3 \times A_5)$. Finally, the TA sends authentication response message $AM_2 = \{A_3, B_3, B_4\}$ back to the SHD over public channels.

Step 3: After obtaining $\{A_3, B_3, B_4\}$ from the TA, the SHD validates the TA by computing parameter A_4 as $A_4 = h(A_3 \times A_2)$. On condition that the computed A_4 is legitimate, the SHD proceeds to derive parameter $B_4^* = h(B_2 \| A_4 \times B_3)$. Next, it checks whether $B_4^* \stackrel{?}{=} B_4$ such that the session is terminated if the two parameters do not match.

Step 4: The SHD uses B_3 and B_4^* to compute parameter C_1 and the $\mathbb{Q}_{ST}$ as $C_1 = h(B_4^* \| R_2 \times B_3)$ and $\mathbb{Q}_{ST} = h(B_3 \| R_2 \times B_3)$. Finally, it sends the computed parameters to the TA in authentication message $AM_3 = \{C_1, \mathbb{Q}_{ST}\}$ over public channels.

Step 5: After getting parameter C_1 and session key $\mathbb{Q}_{ST}$ from the SHD, the TA re-computes them as $C_1^* = h(B_4 \| R_2 \times B_3)$ and $\mathbb{Q}_{ST}^* = h(B_3 \| R_3 \times B_1)$. Next, it confirms whether $C_1^* \stackrel{?}{=} C_1$ and $\mathbb{Q}_{ST}\ \mathbb{Q}_{ST}^* \stackrel{?}{=} \mathbb{Q}_{ST}$. Here, the session is terminated when these verifications are unsuccessful.

Otherwise, the TA and SHD set $\mathbb{Q}_{ST}^* = \mathbb{Q}_{ST}$ as the session key for the current session. Similar procedures are followed by the MD and the TA to authenticate themselves and establish session key $\mathbb{Q}_{MT}$ between themselves.

SHD–MD Authentication Phase

At the onset of the SHD and MD communication process, they have to mutually authenticate each other. As stated earlier, the SHD and the MD must have authenticated themselves to the trusted authority and agreed on session keys $\mathbb{Q}_{ST}$ and $\mathbb{Q}_{MT}$. The next task is for the SHD and the MD to authenticate themselves to each other. Before the commencement of data exchanges between the SHD and the MD, the following 7 procedures are executed. Here, it is assumed that the session is initiated by the remote MD to access some data on the SHD.

Step 1: The MD generates random number R_4 that it deploys to derive parameters C_2, C_3 and C_4 as $C_2 = R_4 \times G$, $C_3 = R_4.h(\mathbb{Q}_{MT})$ and $C_4 = h(C_2 \| C_3 \times G)$. It then constructs

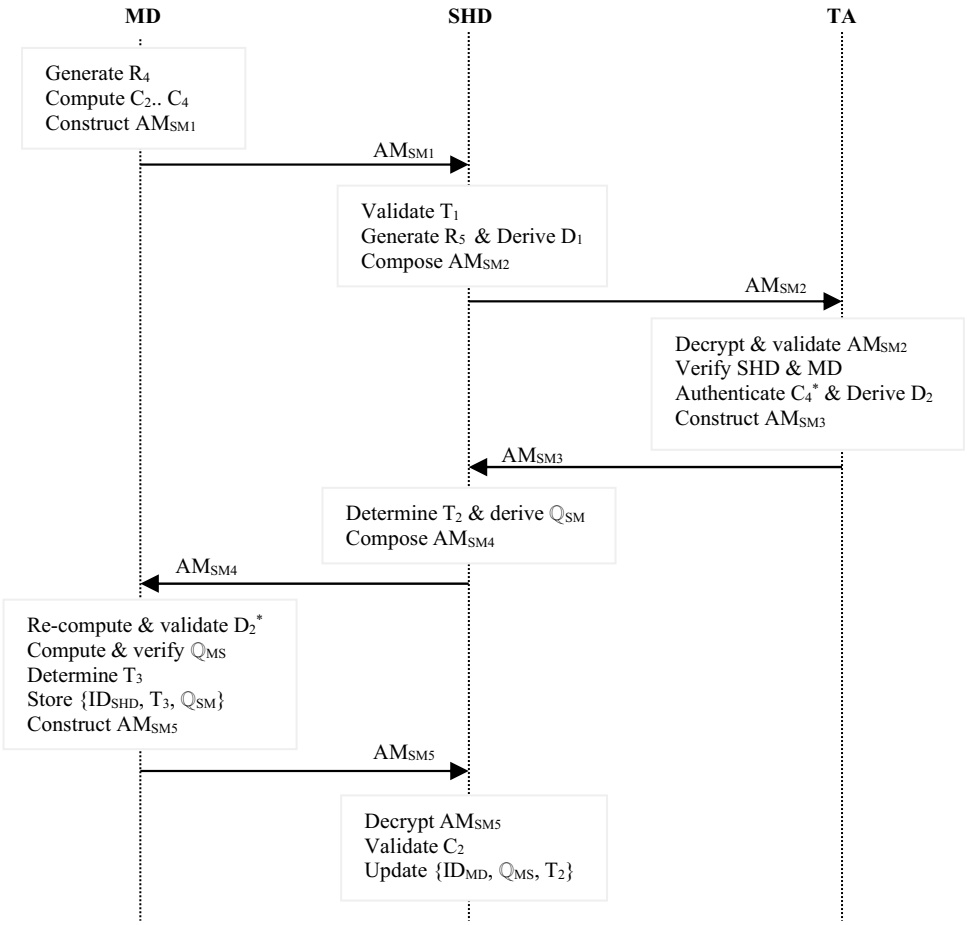

Fig. 3 SHD–MD authentication

authentication request $AM_{SM1} = \{C_2, C_4, ID_{MD}, T_1\}$ that it transmits to SHD as shown in Fig. 3.

Step 2: Upon receiving message AM_{SM1}, the SHD checks its freshness using timestamp T_1 and delay tolerance ΔT. It then generates random number R_5 for the derivation of security parameter $D_1 = R_5 \times G$. Next, it composes authentication response message $AM_{SM2} = E_{\mathbb{Q}_{MT}}(C_2, C_4, D_1, ID_{MD}, T_1)$ which it then sends to the TA for verification. Evidently, this message is protected using session key $\mathbb{Q}_{MT}$ to prevent against eavesdropping and tampering.

Step 3: After getting message AM_{SM2}, the TA decrypts it using $\mathbb{Q}_{MT}$. Next, it retrieves SHD and MD data from its database to determine whether the two had been registered and authenticated themselves with it. Using timestamp T_1 and ΔT, the TA also establishes the freshness of the received request. If all these verifications are successful, TA proceeds to derive $C_4^* = h\,(C_2 \| h(\mathbb{Q}_{MT}) \times C_2)$. It then checks if $C_4^* \stackrel{?}{=} C_4$. This proof proceeds as follows:

$$\begin{aligned} C_4^* &= h\,(C_2 \| h(\mathbb{Q}_{MT}) \times C_2) \\ &= h\,(C_2 \| h(\mathbb{Q}_{MT}) \times (R_4 \times G)); \text{ since } C_2 = R_4 \times G \\ &= h\,(C_2 \| (h(\mathbb{Q}_{MT}).\, R_4) \times G \\ &= h\,(C_2 \| C_3 \times G); \text{ since } C_3 = R_4.h\,(\mathbb{Q}_{MT}) \\ &= C_4; \text{ since } C_4 = h\,(C_2 \| C_3 \times G). \end{aligned}$$

As such, the TA will terminate the authentication session between the SHD and MD whenever $C_4^* \neq C_4$. Otherwise, it computes parameter $D_2 = h\,(D_1 \| h\,(\mathbb{Q}_{MT}) \times C_2)$. Finally, it encrypts D_2 using session key $\mathbb{Q}_{ST}$ in authentication message $AM_{SM3} = E_{\mathbb{Q}_{ST}}(D_1, D_2)$ that is forwarded to the SHD.

Step 4: On obtaining message AM_{SM3}, the SHD is assured that MD is a legitimate entity. As such, it proceeds to determine the current timestamp T_2 before deriving session key $\mathbb{Q}_{SM} = h\,(C_2 \| R_5 \times C_2)$. However, it waits for the confirmation message from MD before storing $\mathbb{Q}_{SM}$ in its memory. This is particularly important in ensuring that both the SHD and MD have generated the same session key. Next, it constructs authentication message $AM_{SM4} = \{D_1, D_2\}$. Finally, it forwards AM_{SM4} to the MD over public channels.

Step 5: After obtaining message AM_{SM4}, the MD re-computes parameter $D_2^* = h\,(D_1 \| C_3 \times G)$ and compares it with parameter D_2 it received from the SHD. This is important for authenticating SHD such that the MD is sure it is communicating with a legitimate SHD. This proof proceeds as follows:

$$D_2^* = h\,(D_1 \| C_3 \times G)$$

$= h\,(D_1 \| h(R_4.h\,(\mathbb{Q}_{MT}))\times G)$; since $C_3 = R_4.h\,(\mathbb{Q}_{MT})$
$= h\,(D_1 \| h(\mathbb{Q}_{MT})\times(R_4\times G))$
$= h\,(D_1 \| h(\mathbb{Q}_{MT})\times C_2)$; since $C_2 = R_4 \times G$
$= D_2$; since $D_2 = h\,(D_1 \| h\,(\mathbb{Q}_{MT})\times C_2)$.

Here, only the TA can generate D_2 using session key $\mathbb{Q}_{MT}$ before transmitting it to SHD. As such, the MD is sure that SHD is a legitimate entity and not any other entity impersonating it.

Step 6: The MD computes session key $\mathbb{Q}_{MS} = h\,(C_2 \| R_4 \times D_1)$, which is then validated against session key $\mathbb{Q}_{SM} = h\,(C_2 \| R_5 \times C_2)$ derived at the SHD. This proof is elaborated below:

$\mathbb{Q}_{MS} = h\,(C_2 \| R_4 \times D_1)$,
$= h\,(C_2 \| R_4 \times (R_5 \times G))$; since $D_1 = R_5 \times G$
$= h\,(C_2 \| (R_4.R_5) \times G)$
$= h\,(C_2 \| R_5 \times C_2)$; since $C_2 = R_4 \times G$.

Provided that $\mathbb{Q}_{SM} = \mathbb{Q}_{MS}$, the SHD and MD have successfully established a common session key. Next, it determines the current timestamp T_3 before storing parameter set $\{ID_{SHD}, T_3, \mathbb{Q}_{SM}\}$ in its memory. Finally, it encrypts parameter C_2 using session key $\mathbb{Q}_{MS}$ and sends it to SHD in authentication message $AM_{SM5} = E_{\mathbb{Q}_{MS}}(C_2)$.

Step 7: Upon receiving confirmation message AM_{SM5}, from the MD, the SHD decrypts it using $\mathbb{Q}_{MS}$. Provided that parameter C_2 can be retrieved successfully from AM_{SM5}, the SHD updates ID_{MD}, session key $\mathbb{Q}_{MS}$ and timestamp T_2 in its memory. This serves to thwart any adversarial packet replay attacks. This confirmation is critical since the SHD might mistakenly update an invalid session key, hence compromising its current communication session with the MD.

Security Analysis

In this section, both formal and informal security analyses are carried out to demonstrate the resilience of the proposed protocol against conventional smart home attacks.

Formal Security Analysis

The Burrows–Abadi–Needham logic (BAN logic) has proofed to be the widely deployed formal analysis model for the correctness of many authentication and key negotiation protocols. As such, the aim of this analysis here is to demonstrate that this protocol successfully attains mutual authentication and session key negotiation between the communicating entities. To achieve this, the notations in Table 2 are utilized.

As shown in Table 2, eleven notations are critical during the execution of the BAN logic proofs (BLPs). On the other hand, Table 3 gives the BAN logic postulates that are applied during these security proofs.

Table 2 BAN Logic Notation

Notation	Description
E^a	EC multiplication by integer a
$\{Z\}_p$	Encryption of Z with key p
$M \equiv Z$	M believes that Z holds
$\langle Z \rangle p$	Z combined with formula p
$M \lhd Z$	M sees statement Z
$G \overset{k}{=} H$	Formula k only known to G and H
$M \Rightarrow Z$	M has full control over formula Z
$M \mid\sim Z$	M once said Z
$\#(Z)$	Z is fresh
$G \overset{k}{\leftrightarrow} H$	G and H share secret key k
$\overset{k}{\mapsto} H$	H has k as its public key

Table 3 BAN Logic postulates

Rule	Description
$\frac{M \equiv (N,Q)}{M \equiv N}$	Believe rule (BR_1)
$\frac{M \equiv E \equiv (N,Q)}{M \equiv E \equiv N}$	Believe rule (BR_2)
$\frac{M \equiv M \overset{k}{\leftrightarrow} E, M \lhd \{N\}_K}{M \equiv E \mid\sim N}$	Message-meaning rule (MMR) for shared key (MMR_1)
$\frac{M \equiv \overset{k}{\mapsto} E, M \lhd \{N\}_{K^{-1}}}{M \equiv E \mid\sim N}$	MMR for public key ((MMR_2) K^{-1} is the inverse of K
$\frac{M \equiv M \overset{k}{=} E, M \lhd N_K}{M \equiv E \mid\sim N}$	MMR for shared secret (MMR_3)
$\frac{M \equiv \#(N), M \mid\equiv E \mid\sim N}{M \equiv E \equiv N}$	Nonce verification rule (NVR)
$\frac{M \mid\sim (N,Q)}{M \mid\sim N}$	Said rule (SAR)
$\frac{M \lhd (N,Q)}{M \lhd N}$	Seeing rule (SER_1)
$\frac{M \equiv M \overset{k}{\leftrightarrow} E, M \lhd \{N\}_K}{M \lhd N}$	Seeing rule (SER_2)
$\frac{M \mid\equiv \overset{k}{\mapsto} E, M \lhd \{N\}_K}{M \lhd N}$	Seeing rule (SER_3)
$\frac{M \mid\equiv \#(N)}{M \mid\equiv \#(N,Q)}$	Random rule (RR)
$\frac{M \mid\equiv \#(k), M \mid\equiv E \mid\equiv N}{M \mid\equiv M \overset{k}{\leftrightarrow} E}$	Session key rule (SKR)
$\frac{M \mid\equiv \#(N)}{M \mid\equiv \#(N,Q)}$	Fresh-promotion rule(FPR_1)
$\frac{M \mid\equiv \#(N)}{M \mid\equiv \#(E^a)}$	Fresh-promotion rule(FPR_2)
$\frac{M \mid\equiv E \Rightarrow N, M \mid\equiv E \mid\equiv N}{M \mid\equiv N}$	Jurisdiction rule (JR)

To show the existence of strong mutual authentication between SHD and MD, the following four goals are formulated:

Goal 1:MD| $\equiv$ MD $\overset{\mathbb{Q}_{MS}}{\leftrightarrow}$ SHD;
Goal 2:MD|$\equiv$ SHD| $\equiv$ MD $\overset{\mathbb{Q}_{MS}}{\leftrightarrow}$ SHD;
Goal 3:SHD| $\equiv$ MD $\overset{\mathbb{Q}_{MS}}{\leftrightarrow}$ SHD;
Goal 4:SHD|$\equiv$ MD| $\equiv$ MD $\overset{\mathbb{Q}_{MS}}{\leftrightarrow}$ SHD.

The five messages that are exchanged during MD and SHD mutual authentication are thereafter translated into idealized format as follows:

MD → SHD: AM_{SM1} $\{C_2, C_4, ID_{MD}, T_1\}$
Idealized format: $E^{R_4}, \langle E^{R_4} \rangle_{\mathbb{Q}_{MT}}$

SHD → TA: AM_{SM2} $\{E_{\mathbb{Q}_{MD}}(C_2, C_4, D_1, ID_{MD}, T_1)\}$
Idealized format: $\{E^{R_4}, \langle E^{R_4} \rangle_{\mathbb{Q}_{MT}}\}_{\mathbb{Q}_{ST}}$
TA → SHD: AM_{SM3} $\{E_{\mathbb{Q}_{ST}}(D_1, D_2)\}$
Idealized format: $\{E^{R_5}, \langle E^{R_4}, E^{R_5} \rangle_{\mathbb{Q}_{MT}}$
SHD → MD: AM_{SM4} $\{D_1, D_2\}$
Idealized format: $\{E^{R_5}, \langle E^{R_4}, E^{R_5} \rangle_{\mathbb{Q}_{MT}}\}$
MD → SHD: AM_{SM5} $\{E_{\mathbb{Q}_{MS}}(C_2)\}$
Idealized format: $\langle E^{R_4} \rangle_{\mathbb{Q}_{MS}}$

For effective proofs using the BAN logic, the following Initial Assumptions (IAs) are then made:

IA_1: $MD \mid \equiv MD \overset{\mathbb{Q}_{MT}}{\leftrightarrow} TA$
IA_2: $MD \mid \equiv MD \overset{\mathbb{Q}_{MT}}{\rightleftharpoons} TA$
IA_3: $SHD \mid \equiv SHD \overset{\mathbb{Q}_{ST}}{\leftrightarrow} TA$
IA_4: $SHD \mid \equiv SHD \mid \overset{\mathbb{Q}_{ST}}{\rightleftharpoons} TA$
IA_5: $TA \mid \equiv MD \overset{\mathbb{Q}_{MT}}{\leftrightarrow} TA$
IA_6: $TA \mid \equiv MD \mid \overset{\mathbb{Q}_{MT}}{\rightleftharpoons} TA$
IA_7: $TA \mid \equiv SHD \overset{\mathbb{Q}_{ST}}{\leftrightarrow} TA$
IA_8: $TA \mid \equiv SHD \mid \overset{\mathbb{Q}_{ST}}{\rightleftharpoons} TA$
IA_9: $MD \mid \equiv SHD \Rightarrow R_5$
IA_{10}: $MD \mid \equiv SHD \Rightarrow E^{R_5}$
IA_{11}: $SHD \mid \equiv MD \Rightarrow R_5$
IA_{12}: $SHD \mid \equiv SHD \Rightarrow E^{R_4}$
IA_{13}: If $MD \mid \equiv TA \mid \equiv N$ then $MD \mid \equiv SHD \mid \equiv N$.

Afterwards, the BAN logic notations, rules, idealized messages and initial assumptions are deployed to execute the BAN logic proofs as follows:

Since the MD is charged with the generation of random number R_4, then:

BLP_1: $MD \mid \equiv R_4$.

Applying RR to the MD's selection of random number R_4, BLP_2 is obtained:

BLP_2: $MD \mid \equiv \#(R_4)$.

The application of FPR_2 to BLP_2 yields BLP_3:

BLP_3: $MD \mid \equiv \#(E^{R_4})$.

Since the random number R_5 is generated by the SHD, then:

BLP_4: $SHD \mid \equiv R_5$.

The application of RR to the SHD's selection of random number R_5, BLP_5 is obtained:

BLP_5: $SHD \mid \equiv \#(R_5)$.

Using FPR_2 in BLP_5 results in BLP_6:

BLP_6: $SHD \mid \equiv \#(E^{R_5})$.

Based on message AM_{SM4}, it is clear that:

BLP_7: $MD \lhd \{E^{R_5}, \langle (E^{R_4}, E^{R_5}) \rangle_{\mathbb{Q}_{MT}}\}$.

The application of SER_1 to BLP_7 yields BLP_8:

BLP_8: $MD \lhd \langle (E^{R_4}, E^{R_5}) \rangle_{\mathbb{Q}_{MT}}$.

Using MMR_3 in IA_2 and BLP_8 yields BLP_9:

BLP_9: $MD \mid \equiv TA \mid\sim (E^{R_4}, E^{R_5})$.

On the other hand, the application of FPR_1 in IA_2 and BLP_3 results in BLP_{10}:

BLP_{10}: $MD \mid \equiv \#(E^{R_4}, E^{R_5})$.

To obtain BLP_{11}, NVR is applied to both BLP_9 and BLP_{10}:

BLP_{11}: $MD \mid \equiv TA \mid \equiv (E^{R_4}, E^{R_5})$.

However, to obtain BLP_{12} and BLP_{13}, BR_1 is applied to BLP_{11}:

BLP_{12}: $MD \mid \equiv TA \mid \equiv E^{R_4}$,
BLP_{13}: $MD \mid \equiv TA \mid \equiv E^{R_5}$.

Considering IA_{13}, BLP_{12} and BLP_{13}, it is evident that:

BLP_{14}: $MD \mid \equiv SHD \mid \equiv E^{R_4}$,
BLP_{15}: $MD \mid \equiv SHD \mid \equiv E^{R_5}$.

On the other hand, the application of SKR to both BLP_{15} and IA_{10} yields BLP_{16}:

BLP_{16}: $MD \mid \equiv E^{R_5}$.

Since session key $\mathbb{Q}_{MS}$ can be expressed as $\mathbb{Q}_{MS} = h(E^{R_4} || E^{R_4 R_5})$, then based on both BLP_2 and BLP_{16}:

BLP_{17}: $MD \mid \equiv \#(\mathbb{Q}_{MS})$.

However, the application of SKR to both BLP_{15} and BLP_{17} results in BLP_{18}:

BLP_{18}: $MD \mid \equiv MD \overset{\mathbb{Q}_{MS}}{\leftrightarrow} SHD$, and as such, **Goal 1** is attained.

To obtain BLP_{19}, BR_2 is applied to BLP_{15}:

BLP_{19}: $MD \mid \equiv SHD \mid \equiv R_5$.

On the other hand, BLP_{20} is easily obtained from BLP_{14} and BLP_{19}:

BLP_{20}: $MD \mid \equiv SHD \mid \equiv MD \overset{\mathbb{Q}_{MS}}{\leftrightarrow} SHD$, achieving **Goal 2**.

Based on the difficulty of solving both the elliptic curve discrete logarithm and the elliptic curve Diffie–Hellman problems, then the belief of MD and SHD can be expressed as in BLP_{21} and BLP_{22}:

BLP_{21}: $MD \mid \equiv MD \overset{E^{R_4 R_5}}{\rightleftharpoons} SHD$,
BLP_{22}: $SHD \mid \equiv MD \overset{E^{R_4 R_5}}{\rightleftharpoons} SHD$.

Regarding idealized message AM_{SM5}, it can be re-written as:

$AM_{SM5}^{*}: \langle E^{R_4}, E^{R_5} \rangle_{E^{R_4 R_5}}$.

Based on AM_{SM5}^{*}, BLP_{23} can be obtained:

BLP_{23}: $SHD \lhd \langle E^{R_4}, E^{R_5} \rangle_{E^{R_4 R_5}}$.

Using MMR_3 in both BLP_{22} and BLP_{23}, it is clear that:

BLP_{24}: $SHD \mid \equiv MD \mid\sim (E^{R_4}, E^{R_5})$.

On the other hand, using FPR_1 in BLP_6 results in BLP_{25}:

BLP_{25}: $SHD \mid \equiv \#(E^{R_4}, E^{R_5})$.

To obtain BLP_{26}, NVR is applied to both BLP_{24} and BLP_{25}:

BLP_{26}: $SHD \mid \equiv MD \mid \equiv (E^{R_4}, E^{R_5})$.

However, to get BLP_{27} and BLP_{28}, BR_2 is applied to BLP_{26}:

BLP_{27}: $SHD \mid \equiv MD \mid \equiv E^{R_4}$.
BLP_{28}: $SHD \mid \equiv MD \mid \equiv E^{R_5}$.

The application of JR to both BLP_{27} and IA_{12} yields BLP_{29}:

BLP_{29}: $SHD \mid \equiv E^{R_4}$.

Since session key can be expressed as $\mathbb{Q}_{MS} = h\ (E^{R_4} || E^{R_4 R_5})$, then based on both BLP_2 and BLP_{16}, BLP_{30} is obtained.

BLP_{30}: SHD $| \equiv \#\ (\mathbb{Q}_{MS})$.

In addition, using SKR in both BLP_{27} and BLP_{30} results in BLP_{31}:

BLP_{31}: SHD$| \equiv$ MD $\overset{\mathbb{Q}_{MS}}{\leftrightarrow}$ SHD, effectively attaining **Goal 3**.

However, using BR_2 in BLP_{15} results in BLP_{32}:

BLP_{32}: SHD $| \equiv$ MD $| \equiv R_4$.

Based on both BLP_{28} and BLP_{32}, BLP_{33} is obtained:

BLP_{33}:SHD$| \equiv$ MD$| \equiv$ MD $\overset{\mathbb{Q}_{MS}}{\leftrightarrow}$ SHD, hence **Goal 4** is realized.

The successful attainment of all the four goals formulated earlier shows the existence of mutual authentication between the MD and the SHD. In addition, it demonstrates the existence of a session key that enciphers traffic exchanged between these entities.

Informal Security Analysis

In this section, it is demonstrated that the proposed scheme is secure under the Canetti–Krawczyk threat model. The assumptions of the CK threat model are given in [7]. To accomplish these security proofs, the following lemmas are formulated and proofed.

Lemma 1 *The proposed algorithm prevents man-in-the-middle attacks.*

Proof The goal of this attack is to capture the exchanged messages, modify and forward them to the unsuspecting receivers. Suppose that the attacker has captured parameters C_2, C_4, D_1 and D_2. Here, $C_2 = R_4 \times G$, $C_4 = h\ (C_2 || C_3 \times G)$, $D_1 = R_5 \times G$ and $D_2 = h\ (D_1 || h\ (\mathbb{Q}_{MT}) \times C_2)$. Next, an adversary tries to construct messages $AM_{SM1} = \{C_2, C_4, ID_{MD}, T_1\}$, $AM_{SM2} = E_{\mathbb{Q}_{MT}}(C_2, C_4, D_1, ID_{MD}, T_1)$, $AM_{SM3} = E_{\mathbb{Q}_{ST}}(D_1, D_2)$, $AM_{SM4} = \{D_1, D_2\}$ and $AM_{SM5} = E_{\mathbb{Q}_{MS}}(C_2)$. Clearly, the construction of valid messages require additional parameters such as random numbers R_4 and R_5, the MD's real identity, session key between MD and TA ($\mathbb{Q}_{MT}$), session key between SHD and TA ($\mathbb{Q}_{ST}$) and the session key between SHD and MD ($\mathbb{Q}_{MS}$). Since these security parameters are unavailable to the adversary, this attack flops. In addition, the derivation of R_4 from C_2 is computationally infeasible.

Lemma 2 *The communicating entities are properly authenticated to each other.*

Proof In this protocol, upon receiving message AM_{SM2} from the SHD, the TA utilizes $\mathbb{Q}_{MT}$ to decrypt it. Afterwards, it retrieves SHD and MD data from its database to determine whether the two had been registered and authenticated themselves with it. In addition, the TA authenticates the SHD by checking whether $C_4^* \overset{?}{=} C_4$. On the other hand, on obtaining message AM_{SM4} from the SHD, the MD re-computes $D_2^* = h\ (D_1 || C_3 \times G)$ and compares it with parameter D_2 it received in AM_{SM4}. Here, it is only legitimate TA that can derive D_2 using $\mathbb{Q}_{MT}$ before forwarding it to SHD. Consequently, the MD is confident that SHD is a legitimate entity and not any other masquerading entity.

Lemma 3 *Packet replay attacks are effectively thwarted in this scheme.*

Proof The purpose of this attack is to intercept the transmitted messages, store them and re-transmit them later to the intended receivers. Suppose that an adversary captures message $AM_{SM1} = \{C_2, C_4, ID_{MD}, T_1\}$ sent from the MD towards the SHD. After sometimes, the attacker re-sends it to the SHD in an effort to fool the SHD that the MD is requesting another communication session. However, any replayed message will fail the freshness checks at the SHD. Similarly, any adversarial effort to replay message $AM_{SM2} = E_{\mathbb{Q}_{MT}}(C_2, C_4, D_1, ID_{MD}, T_1)$ will be detected at the TA using T_1. As such, the proposed protocol is robust against packet replay attacks.

Lemma 4 *This protocol offer backward and forward key secrecy.*

Proof At the SHD, the session key $\mathbb{Q}_{SM}$ is derived as $\mathbb{Q}_{SM} = h\ (C_2 || R_5 \times C_2)$, where $C_2 = R_4 \times G$. Similarly, session key $\mathbb{Q}_{MS}$ is computes as $\mathbb{Q}_{MS} = h\ (C_2 || R_4 \times D_1)$ at the MD. Here, $D_1 = R_5 \times G$ and $C_2 = R_4 \times G$. Evidently, these session keys incorporate random numbers R_4 and R_5. As such, they are stochastic such that different sessions have different keys. Consequently, the capture of any key belonging to the current session cannot facilitate the derivation of keys used in the previous and subsequent communication sessions. Similarly, an attacker with captured session keys cannot utilize them to decrypt messages for the current as well as subsequent communication sessions.

Lemma 5 *Privileged insider attacks are prevented in this scheme.*

Proof In this protocol, the trusted authority mediates the authentication and key agreement between the MD and the SHD. In this attack, it is assumed that the TA is a privileged entity that may attempt to derive the MD-SHD session keys using the security parameters it has access to. Here, the SHD derives session key $\mathbb{Q}_{SM}$, where $\mathbb{Q}_{SM} = h\ (C_2 || R_5 \times C_2)$ and $C_2 = R_4 \times G$. On the other hand, the MD computes session

key $\mathbb{Q}_{MS}$, where $\mathbb{Q}_{MS} = h\,(C_2 \| R_4 \times D_1)$ and $D_1 = R_5 \times G$. Evidently, the derivation of session keys $\mathbb{Q}_{SM}$ and $\mathbb{Q}_{MS}$ requires random numbers R_4 and R_5. Here, random number R_4 is generated at the MD while random number R_5 is generated at the SHD. As such, although the TA supervises the authentication between the MD and SHD, it cannot derive the session keys for traffic enciphering between the two entities. Consequently, it is unable to encrypt or decrypt the exchanged messages between the MD and the SHD.

Lemma 6 *The proposed protocol is robust against session hijack and denial of service attacks.*

Proof The ultimate objective of these attacks is to cut off the communication between the MD or SHD and the TA. To carry out this attack, an adversary tries to derive legitimate session keys $\mathbb{Q}_{MT}$ and $\mathbb{Q}_{ST}$. Here, $\mathbb{Q}_{ST} = \mathrm{h}\,(B_3 \| R_2 \times B_3)$ and $B_3 = R_3 \times G$. The session key between the MD and the TA is derived in a similar version. Clearly, the derivation of any legitimate session key requires knowledge of the random numbers R_2 and R_3. Here, R_2 is generated at the SHD and MD while R_3 is generated at the TA. As such, adversarial computation of these session keys will fail due to the difficulty of deriving R_3 from B_3. Suppose that an attacker attempts to construct authentication messages AM_1, AM_2 and AM_3. Here, $\mathrm{AM}_1 = \{B_1, B_2\}$, $\mathrm{AM}_2 = \{A_3, B_3, B_4\}$, $\mathrm{AM}_3 = \{C_1, \mathbb{Q}_{ST}\}$, $A_1 = h\,(R_1 \| \mathrm{TA}_{SV} \| T_1 \| \mathrm{ID}_{SHD})$, $A_2 = A_1 \times G$, $A_3 = R_1 \oplus h\,(\mathrm{TA}_{SV})$, $A_4 = h\,(R_1 \oplus h\,(\mathrm{TA}_{SV}) \| A_2)$, $A_5 = A_4 \times G$ $B_2 = h\,(B_1 \| R_2 \times A_2)$, $B_1 = R_2 \times G$, $B_2 = h\,(B_1 \| R_2 \times A_2)$, $B_3 = R_3 \times G$, $B_4 = h\,(B_2^* \| R_3 \times A_5)$, $C_1 = h\,(B_4^* \| R_2 \times B_3)$, $B_4^* = h\,(B_2 \| A_4 \times B_3)$ and $\mathbb{Q}_{ST} = h\,(B_3 \| R_2 \times B_3)$. It is evident that in addition to random numbers R_1, R_2 and R_3, the attacker needs TA's secret value TA_{SV}, timestamp T_1 and the SHD's unique identity ID_{SHD} to construct these messages. Since all these parameters are unavailable to the adversary, the construction of these messages fails. Therefore, the sessions of the MD and SHD are sufficiently protected and cannot be hijacked. Ultimately, availability is upheld and the remote users are able to access the sensed data anytime and from any location.

Lemma 7 *Impersonation attacks are prevented in this protocol.*

Proof The aim of these attacks is to send connection requests using the identities of other network entities. Suppose that an adversary wants to impersonate the SHD and send connection request AM_1 to the TA. Here, $\mathrm{AM}_1 = \{B_1, B_2\}$, $B_1 = R_2 \times G$, $B_2 = h\,(B_1 \| R_2 \times A_2)$, $A_2 = A_1 \times G$ and $A_1 = h\,(R_1 \| \mathrm{TA}_{SV} \| T_1 \| \mathrm{ID}_{SHD})$. It is clear that to construct a legitimate connection request AM_1, the adversary requires the TA's secret value TA_{SV}, timestamp T_1 and the SHD's unique identity. In addition, random numbers R_1 and R_2 are needed. In this protocol, one-way hashing operations prevent an attacker from obtaining T_1, TA_{SV} and ID_{SHD} from parameter A_1. Although hashing functions can have collisions, only the underlying hashing algorithm can be discerned while the protected data remains secure. On the other hand, the stochastic nature of random numbers makes it computationally infeasible for the attacker to derive them with high success probability.

Lemma 8 *This protocol preserves the confidentiality and integrity of the communication process.*

Proof To uphold confidentiality, all exchanged messages are sufficiently enciphered using the derived session keys. These session keys can only be derived between the communicating devices after successful mutual authentication process. For instance, authentication message $\mathrm{AM}_{SM2} = \mathrm{E}_{\mathbb{Q}_{MT}}(C_2, C_4, D_1, \mathrm{ID}_{MD}, T_1)$ is encrypted using the session key set between MD and TA ($\mathbb{Q}_{MT}$). Similarly, authentication message $\mathrm{AM}_{SM3} = \mathrm{E}_{\mathbb{Q}_{ST}}(D_1, D_2)$ is enciphered using the session key set between SHD and TA ($\mathbb{Q}_{ST}$). On the other hand, authentication message $\mathrm{AM}_{SM5} = \mathrm{E}_{\mathbb{Q}_{MS}}(C_2)$ is encrypted using Session key set between SHD and MD ($\mathbb{Q}_{MS}$). As such, the enciphered parameters cannot be modified on transit and hence, their integrity is preserved.

Lemma 9 *Offline dictionary attacks are thwarted in this scheme.*

Proof The goal of an adversary in this attack is to capture the exchanged messages and attempt to discern sensitive information in them. Finally, the learned messages are utilized to compute the session keys $\mathbb{Q}_{SM}$, $\mathbb{Q}_{MS}$, $\mathbb{Q}_{ST}$ and $\mathbb{Q}_{MT}$. Here, $\mathbb{Q}_{ST} = \mathrm{h}\,(B_3 \| R_2 \times B_3)$, $\mathbb{Q}_{SM} = h\,(C_2 \| R_5 \times C_2)$ and $\mathbb{Q}_{MS} = \mathrm{h}\,(C_2 \| R_4 \times D_1)$. On the other hand, the exchanged messages between the MD and SHD include AM_{SM1}, AM_{SM2}, AM_{SM3}, AM_{SM4} and AM_{SM5}. Here, $\mathrm{AM}_{SM1} = \{C_2, C_4, \mathrm{ID}_{MD}, T_1\}$, $\mathrm{AM}_{SM2} = \mathrm{E}_{\mathbb{Q}_{MT}}(C_2, C_4, D_1, \mathrm{ID}_{MD}, T_1)$, $\mathrm{AM}_{SM3} = \mathrm{E}_{\mathbb{Q}_{ST}}(D_1, D_2)$, $\mathrm{AM}_{SM4} = \{D_1, D_2\}$ and $\mathrm{AM}_{SM5} = \mathrm{E}_{\mathbb{Q}_{MS}}(C_2)$. Here, both $\mathbb{Q}_{ST}$ and $\mathbb{Q}_{MT}$ were derived in early device-TA authentication phase. As such, the attacker lacks random number R_2 and security parameter B_3 needed to derive them. In addition, the one-way hashing function renders it infeasible to obtain these two parameters through reverse engineering. Although messages AM_{SM1} and AM_{SM4} are transmitted in plaintext, their contents do not contain random numbers R_4 and R_5 required to derive session keys $\mathbb{Q}_{MS}$ and $\mathbb{Q}_{SM}$. Consequently, the proposed protocol is resilient against offline-guessing attacks.

Lemma 10 *The proposed protocol is robust against stolen verifier attacks.*

Proof Suppose that an attacker manages to steal the session keys $\mathbb{Q}_{SM}$ and $\mathbb{Q}_{MS}$ used to encrypt and decrypt traffic between the MD and the SHD. An attempt may then be made to derive

session keys $\mathbb{Q}_{MT}$ and $\mathbb{Q}_{ST}$ deployed to secure device–TA communication. Here, $\mathbb{Q}_{MS}=h\,(C_2\|R_4\times D_1)$, $\mathbb{Q}_{SM}=h\,(C_2\|R_5\times C_2)$ and $\mathbb{Q}_{ST}=h\,(B_3\|R_2\times B_3)$. Evidently, the captured session keys can only be deployed to encrypt and decrypt MD–SHD traffic for only the current session. They cannot be deployed for encryption and decryption in the past or subsequent sessions due to the random numbers R_4 and R_5 which imply that these session keys are different for dissimilar sessions. The one-way hashing operation prevents an attacker from sniffing the contents of these session keys for other malicious verifications. It is also evident that session keys $\mathbb{Q}_{ST}$ derivations requires random number R_2 and security parameter B_3, all of which cannot be discerned from the captured session keys.

Performance Evaluation

In this section, the lightweight nature of the proposed algorithm is demonstrated. To accomplish this, performance metrics such as computation and communication overheads, memory requirements and energy consumption are used. In addition, experiments are run to investigate energy consumption variations under different transmission loads. Moreover, the security features provided by this scheme are compared with the ones offered by other related schemes.

Computation Overhead

In this sub-section, the execution time of the various cryptographic primitives during the MD–SHD mutual authentication is taken into consideration. During this phase, only three cryptographic operations are executed. These are elliptic curve point multiplication (T_{EP}), symmetric encryption and decryption (T_{SED}), and one-way hash function (T_H). At the MD, $4T_H+3T_{EP}+4T_{SED}$ operations are carried out. On the other hand, $1T_H+2T_{EP}+14T_{SED}$ operations are executed at the SHD. Similarly, $4T_H+2T_{EP}+14T_{SED}$ operations are carried out at the TA. As such, the total computation overhead of this algorithm is $9T_H+7T_{EP}+32T_{SED}$. Based on the values in [2], Table 4 gives the execution time for the various cryptographic primitives.

Based on the values in Table 4, the total computation overhead in the proposed protocol is 3.728 ms. On the other hand, Table 5 presents the computation overheads of other related schemes.

Table 4 Cryptographic primitives execution time

Cryptographic primitive	Time (ms)
One-way hashing	0.0052
Symmetric encryption/decryption	0.0215
EC point multiplication	0.4276
Bilinear operation	5.811
ECC point addition	0.0288

Table 5 Computation overheads

Scheme	Time (ms)
[2]	1.366
[24]	6.93
[25]	1.366
[35]	2.24
[39]	7.84
[43]	3.2
Proposed	3.728

As shown in Fig. 4, the protocol in [39] has the highest computation overheads of 7.84 ms. This is followed by the protocol in [24] with execution time of 6.93 ms. The proposed scheme is third, followed by the schemes in [35, 43] and the schemes in [2] and [25], respectively. Although the schemes in [2] and [25] have low computation complexities, they have a number of security issues. For instance, the scheme in [2] is vulnerable to session hijacking, privileged insiders, packet replays, offline dictionary and stolen verifier attacks. In addition, its design doesn't consider confidentiality, integrity and MitM attacks. Similarly, the scheme in [25] fails to consider MitM, integrity and confidentiality in its design. On its part, the protocol in [35] cannot withstand DoS attacks.

In addition, its design fails to consider communication integrity and confidentiality, as well as attack models such as offline dictionary, privileged insiders and session hijacking. Similarly, the protocol in [43] cannot offer confidentiality, integrity and protection against session hijacking attacks.

Communication Overheads

In this section, the bandwidth requirement of the proposed scheme is derived. During MD and SHD mutual authentication, messages AM_{SM1}, AM_{SM2}, AM_{SM3}, AM_{SM4} and

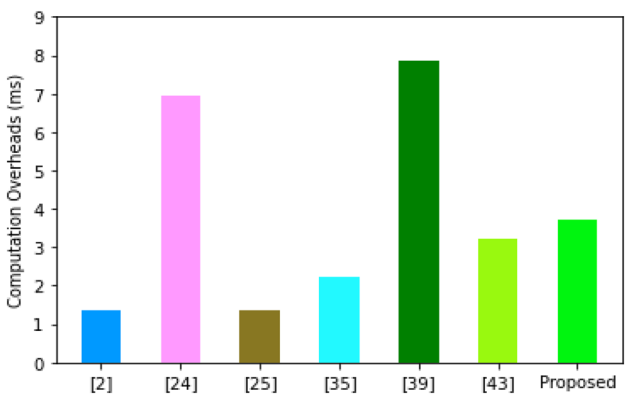

Fig. 4 Computation overheads

Table 6 Cryptographic output sizes

Cryptographic output	Size (bits)
One-way hashing	160
Identity	128
Symmetric encryption/decryption	256
EC point multiplication	320
Timestamps	32
Random number	128

Table 7 Message sizes derivations

Message	Size (bits)
$AM_{SM1} = \{C_2, C_4, ID_{MD}, T_1\}$ $C_2 = C_4 = ID_{MD} = 128; T_1 = 32$	416
$AM_{SM2} = E_{\mathbb{Q}_{MT}}(C_2, C_4, D_1, ID_{MD}, T_1)$	256
$AM_{SM3} = E_{\mathbb{Q}_{ST}}(D_1, D_2)$	256
$AM_{SM4} = \{D_1, D_2\}$ $D_1 = 128; D_2 = 160$	288
$AM_{SM5} = E_{\mathbb{Q}_{MS}}(C_2)$	256
Total	1472

Table 8 Communication overheads

Scheme	(bits)
[2]	1728
[24]	3296
[25]	1856
[35]	794
[39]	986
[43]	2304
Proposed	1472

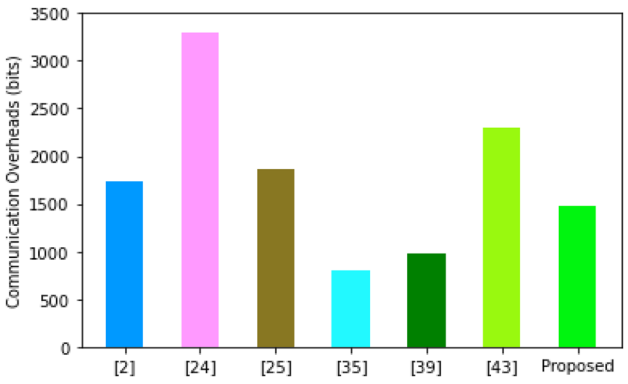

Fig. 5 Communication overheads

AM_{SM5} are exchanged. Here, $AM_{SM1} = \{C_2, C_4, ID_{MD}, T_1\}$, $AM_{SM2} = E_{\mathbb{Q}_{MT}}(C_2, C_4, D_1, ID_{MD}, T_1)$, $AM_{SM3} = E_{\mathbb{Q}_{ST}}(D_1, D_2)$, $AM_{SM4} = \{D_1, D_2\}$ and $AM_{SM5} = E_{\mathbb{Q}_{MS}}(C_2)$. Table 6 gives the output sizes of the various cryptographic primitives.

Table 9 Memory Requirements

Scheme	(bytes)
[2]	64
[24]	120
[25]	48
[35]	–
[39]	80
[43]	48
Proposed	48

Using the values in Table 6, the derivation of the communication overheads of the proposed scheme is carried out as shown in Table 7. Based on these derivations, the cumulative communication overhead of this scheme is 1472 bits.

However, the communication costs for the protocols in [2, 24, 25, 35, 39] and [43] are 1728 bits, 3296 bits, 1856 bits, 794 bits, 986 bits and 2304 bits, respectively, as shown in Table 8.

Based on the values in Fig. 5, the protocol in [24] has the highest communication overheads. This is followed by the scheme in [25, 43] and [2]. On the other hand, the proposed protocol has the third lowest communication costs, followed by the protocols in [39] and [35], respectively.

Although the protocol in [35] has the lowest communication costs, it is susceptible to DoS attacks. In addition, it does not consider attack models such as offline dictionary, privileged insiders and session hijacking. Moreover, communication integrity and confidentiality are never catered for in this scheme. On the other hand, the scheme in [39] is vulnerable to packet replay attacks. In addition, it cannot offer key secrecy as well as communication integrity and confidentiality.

Memory Requirements

The number of bytes stored during the MD and SHD authentication process is considered in this section. Here, each device is required to store only the session key for each of the other device. As such, MD needs to store $\mathbb{Q}_{SM}$ while the SHD is required to store $\mathbb{Q}_{MS}$. For packet replay prevention, the delay tolerance time may also need to be stored in both the SHD and MD. Based on the values in [47], $\mathbb{Q}_{SM} = \mathbb{Q}_{MS} = 160$ bits. On the other hand, timestamp $T_1 = T_2 = T_3 = 32$ bits. As such, both the SH and MD require 192 bits of storage each, which is equivalent to 24 bytes. Consequently, the total memory requirement of this protocol during MD–SHD mutual authentication is 48 bytes as shown in Table 9.

It is evident from Fig. 6 that the protocol in [24] has the largest memory requirements. This is followed by the schemes in [39] and [2], respectively. On the other hand, the

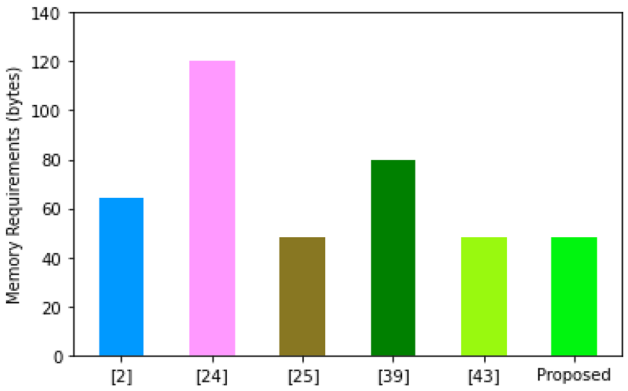

Fig. 6 Memory requirements

proposed scheme together with the protocols in [25] and [43] have the lowest memory requirements.

As such, the proposed scheme together with these two other protocols put the least strain on the sensor memory. Consequently, they are the most suitable for deployments in smart home networks.

Energy Consumptions

In this section, the number of bits exchanged during the mutual authentication and key agreement between the MD and the SHD are deployed to derive the energy consumption of these two devices. The messages transmitted and received are as follows.

MD → SHD: $AM_{SM1} = \{C_2, C_4, ID_{MD}, T_1\}$
SHD → TA: $AM_{SM2} = E_{\mathbb{Q}_{MT}}(C_2, C_4, D_1, ID_{MD}, T_1)$
TA → SHD: $AM_{SM3} = E_{\mathbb{Q}_{ST}}(D_1, D_2)$
SHD → MD: $AM_{SM4} = \{D_1, D_2\}$
MD → SHD: $AM_{SM5} = E_{\mathbb{Q}_{MS}}(C_2)$

Based on the values in Table 7, $AM_{SM1} = 416$ bits, $AM_{SM2} = 256$, $AM_{SM3} = 256$, $AM_{SM4} = 288$, and $AM_{SM5} = 256$. As such, the MD sends and receives a total of 672 bits and 288 bits, respectively. On the other hand, the TA sends and receives 256 bits in each case. However, the SHD sends and receives 544 bits and 928 bits, respectively. As pointed out in [42], single-bit transmission and reception on TelosB requires 0.00072 mJ and 0.00081 mJ, respectively.

Table 10 Energy consumptions

Entity	Sending (bits)	Energy (mJ)	Receiving (bits)	Energy (mJ)	Total (mJ)
MD	672	0.48384	288	0.23328	0.71712
SHD	544	0.39168	928	0.75168	1.14336
TA	256	0.18432	256	0.20736	0.39168
Total	2.25216				

Using these values, Table 10 presents the energy consumptions of these three entities.

Based on the values in Table 10, the MD consumes more energy to send requests than to receive and process requests. On the other hand, both the SHD and TA consume more energy to receive and process requests than to send requests. In the face of active DoS attacks, the MD will be least affected, while the SHD will be the most affected. This is because the SHD will commit sufficient computational resources to process and authenticate incoming requests when compared with the MD. To further investigate the implication of concurrent requests on the energy consumptions, experimentations were run in Python programming language. The specifications of the host machine were as follows: i5-4210U, Windows 10 Pro 64-bit, CPU1.70 Ghz × 4 and 4G RAM. Figure 7 shows the variation of the energy consumptions as a function of the number of concurrent requests and number of SHDs.

As shown in Fig. 7, as the number of SHDs increase, there is a corresponding increase in energy consumptions. This is attributed to the increased processing at the terminals. It is also evident that at a particular SHDs density, there is more energy consumed when the number of concurrent requests surge.

Security Features

To appreciate the security features offered by the proposed scheme, comparisons are made with other related schemes as shown in Table 11. It is clear that the scheme in [2] supports only 5 security features while the protocol in [35] offers 7 security features. This was followed by the scheme in [24, 25, 39] and [43] which provide 8, 8,10 and 10 security features, respectively.

On the other hand, the proposed scheme supports 14 security features, which is the highest number. As such, although this scheme has relatively higher computation

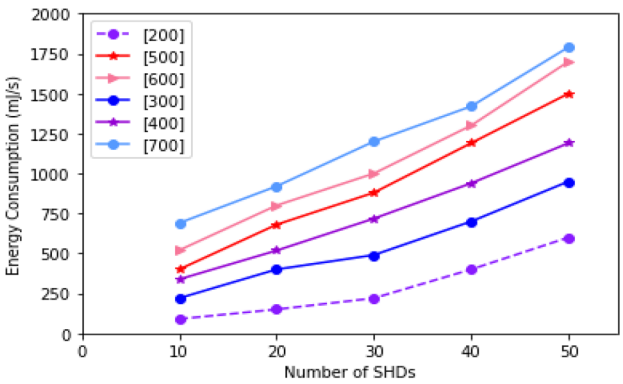

Fig. 7 Energy variations at varying loads

Table 11 Security features

Feature	[24]	[2]	[35]	[25]	[43]	[39]	Proposed
Mutual authentication	√	√	√	√	√	√	√
Session key agreement	√	√	√	√	√	√	√
Key secrecy	√	√	√	√	√	x	√
Confidentiality	–	–	–	–	–	–	√
Integrity	–	–	–	–	–	–	√
Attacks resilience							
Impersonations	√	√	√	√	√	√	√
DoS	√	√	x	√	√	√	√
Session hijacking	√	x	–	√	–	√	√
Privileged insiders	x	x	–	√	√	√	√
Packet replays	√	x	√	√	√	x	√
MitM		–	√	–	√	–	√
Offline dictionary	√	x	–	√	√	√	√
Stolen verifier	x	x	√	√	√	√	√

√ Effective

x Ineffective

– Not considered

and communication overheads, it is the most secure among all these other schemes.

Conclusion

Smart homes introduce convenience, comfort and energy consumptions through automated management and control of activities. However, security and privacy challenges are very pertinent setbacks that may hamper the adoption of smart homes. Although many authentication techniques have been developed to address these issues, these schemes have some performance challenges. In addition, none of these schemes addresses all the required security and privacy goals. Consequently, security and privacy provisioning in smart homes is still a challenging issue. The presented scheme has been demonstrated to have lower computation and communication overheads compared to other conventional approaches. On the other hand, its memory requirement is the lowest among other similar protocols. The security features such as strong mutual authentication, backward and forward key secrecy, session key agreement, as well as the preservation of both integrity and confidentiality make it attractive for deployment in smart homes. On the other hand, its resilience against impersonations, denial of service, session hijacking, privileged insiders, packet replays, man-in-the-middle, offline dictionary and stolen verifier attacks render it superior to other related schemes. Future work will involve the evaluation of the proposed scheme using security attack models and performance metrics that were not within the scope of this paper. There is also need to come up with new innovative techniques for reducing the communication and computation costs of this scheme.

Declarations

Conflict of Interest The author declares that he has no conflict of interest.

References

1. Alam MR, St-Hilaire M, Kunz T. Peer-to-peer energy trading among smart homes. Appl Energy. 2019;238:1434–43.
2. Shuai M, Yu N, Wang H, Xiong L. Anonymous authentication scheme for smart home environment with provable security. Comput Secur. 2019;86:132–46.
3. Park JH, Salim MM, Jo JH, Sicato JCS, Rathore S, Park JH. CIoT-Net: a scalable cognitive IoT based smart city network architecture. Human Compu Inf Sci. 2019;9(1):1–29.
4. Ringel M, Laidi R, Djenouri D. Multiple benefits through smart home energy management solutions—a simulation-based case study of a single-family-house in algeria and Germany. Energies. 2019;12:1537.
5. Gonçalves I, Gomes A, Antunes CH. Optimizing the management of smart home energy resources under different power cost scenarios. Appl Energy. 2019;242:351–63.
6. Bakar U, Ghayvat H, Hasanm S, Mukhopadhyay S. Activity and anomaly detection in smart home: a survey, next generation sensors and systems. Berlin: Springer; 2016. p. 191–220.
7. Nyangaresi VO. ECC based authentication scheme for smart homes. In 2021 International Symposium (ELMAR), 5–10, IEEE, 2021.
8. Dang TLN, Nguyen MS. An approach to data privacy in smart home using blockchain technology. In: 2018 International Conference on Advanced Computing and Applications (ACOMP), 58–64, IEEE, 2018.
9. Gu K, Yang L, Yin B. Location data record privacy protection based on differential privacy mechanism. Inf Technol Control. 2018;47(4):639–54.
10. Ramapatruni S, Narayanan SN, Mittal S, Joshi A, Joshi K. Anomaly detection models for smart home security. In 2019 IEEE 5th International Conference on Big Data Security on Cloud (BigDataSecurity),

IEEE International Conference on High Performance and Smart Computing,(HPSC) and IEEE Internationl Conference on Intelligent Data and Security (IDS), 19–24, IEEE, 2019.
11. Singh S, Sharma PK, Park JH. SH-SecNet: an enhanced secure network architecture for the diagnosis of security threats in a smart home. Sustainability. 2017;9:1–19.
12. Nyangaresi VO, Ogundoyin SO. Certificate based authentication scheme for smart homes. In: 2021 3rd Global Power, Energy and Communication Conference (GPECOM), 202–207, IEEE, 2021.
13. Moniruzzaman M, Khezr S, Yassine A, Benlamri R. Blockchain for smart homes: Review of current trends and research challenges. Comput Electr Eng. 2020;83: 106585.
14. Jia Y, Xiao Y, Yu J, Cheng X, Liang Z, Wan Z. A novel graph-based mechanism for identifying traffic vulnerabilities in smart home IoT. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, 1493–1501, IEEE, 2018.
15. Fahad LG, Tahir SF. Activity recognition and anomaly detection in smart homes. Neurocomputing. 2021;423:362–72.
16. Forkan ARM, Khalil I, Tari Z, Foufou S, Bouras A. A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living. Pattern Recogn. 2015;48(3):628–41.
17. Saqaeeyan S, Amirkhani H. Anomaly detection in smart homes using bayesian networks. KSII Trans Internet Inf Syst (TIIS). 2020;14(4):1796–816.
18. Zhu C, Sheng W, Liu M. Wearable sensor-based behavioral anomaly detection in smart assisted living systems. IEEE Trans Autom Sci Eng. 2015;12(4):1225–34.
19. Sivanathan A, Sherratt D, Gharakheili HH, Sivaraman V, Vishwanath A. Low-cost flow-based security solutions for smart home IOT devices. In: 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), 1–6, IEEE, 2016.
20. Anton K, Aleksandr N, Catherine B, Denis N, Aleksei S, Aleksandr E, Kseniia N. Anomaly detection in wireless sensor network of the smart home system. In: 2017 20th Conference of Open Innovations Association (FRUCT), 118–124, IEEE, 2017.
21. Spanos G, Giannoutakis KM, Votis K, Tzovaras D. Combining statistical and machine learning techniques in IoT anomaly detection for smart homes. In: 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 1–6, IEEE, 2019.
22. Zeng Y, Meikang Q, Zhong M, Meiqin L. Senior2local: a machine learning based intrusion detection method for vanets. In: International conference on smart computing and communication. Berlin: Springer; 2018. p. 417–26.
23. Challa S, Das AK, Odelu V, Kumar N, Kumari S, Khan MK, Vasilakos AV. An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks. Comput Electr Eng. 2018;69:534–54.
24. Sureshkumar V, Amin R, Vijaykumar V, Sekar SR. Robust secure communication protocol for smart healthcare system with fpga implementation. Future Gener Comput Syst. 2019;100:938–51.
25. Kaur D, Kumar D. Cryptanalysis and improvement of a two-factor user authentication scheme for smart home. J Inf Secur Appl. 2021;58: 102787.
26. AbuNaser M, Alkhatib AA. Advanced survey of blockchain for the internet of things smart home. In: 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), 58–62, IEEE, 2019.
27. Lee Y, Rathore S, Park JH, Park JH. A blockchain-based smart home gateway architecture for preventing data forgery. HCIS. 2020;10(1):1–14.
28. Almadhoun R, Kadadha M, Alhemeiri M, Alshehhi M, Salah K. A user authentication scheme of IoT devices using blockchain-enabled fog nodes. In: 2018 IEEE/ACS 15th international conference on computer systems and applications (AICCSA), 1–8, IEEE, 2018.
29. Bahga A, Madisetti VK. Blockchain platform for industrial internet of things. J Softw Eng Appl. 2016;9(10):533–46.
30. Dorri A, Kanhere SS, Jurdak R, Gauravaram P. Blockchain for ToT security and privacy: The case study of a smart home. In: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 618–23, IEEE, 2017.
31. Dwivedi AD, Srivastava G, Dhar S, Singh R. A decentralized privacy-preserving healthcare blockchain for IoT. Sensors. 2019;19(2):326.
32. Nyangaresi, VO, Abduljabbar, ZA, Al Sibahee, MA, Abduljaleel IQ, Abood EW. Towards Security and Privacy Preservation in 5G Networks. In: 2021 29th Telecommunications Forum (TELFOR),1–4, IEEE, 2021.
33. Ba Z, Piao S, Fu X, Koutsonikolas D, Mohaisen A, Ren K. ABC: enabling smartphone authentication with built-in camera. In: Network and Distributed System Security Symposium, 18–21, 2018.
34. Nimmy K, Sankaran S, Achuthan K, Calyam P. Lightweight and privacy-preserving remote user authentication for smart homes. IEEE Access. 2021;10:176–90.
35. Yu S, Das AK, Park Y. Comments on "ALAM: anonymous lightweight authentication mechanism for SDN enabled smart homes. IEEE Access. 2021;9:49154–9.
36. Santoso FK, Vun NCH. Securing IoT for smart home system. In 2015 international symposium on consumer electronics (ISCE), 1–2, IEEE, 2015.
37. Parne BL, Gupta S, Chaudhari NS. Pse-aka: performance and security enhanced authentication key agreement protocol for iot enabled lte/lte-a networks. Peer-to-Peer Netw Appl. 2019;12(5):1156–77.
38. Shen J, Yang H, Wang A, Zhou T, Wang C. Lightweight authentication and matrix-based key agreement scheme for healthcare in fog computing. Peer-to-Peer Netw Appl. 2019;12(4):924–33.
39. Wazid M, Das AK, Odelu V, Kumar N, Susilo W. Secure remote user authenticated key establishment protocol for smart home environment. IEEE Trans Dependable Secure Comput. 2020;17(2):391–406.
40. Sankaran S. Lightweight security framework for IoTs using identity based cryptography. In: 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2016, pp. 880–6.
41. Nyangaresi VO. Provably secure protocol for 5G HetNets. In: 2021 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS), IEEE, 2021, pp. 17–22.
42. Kumar P, Gurtov A, Iinatti J, Ylianttila M, Sain M. Lightweight and secure session-key establishment scheme in smart home environments. IEEE Sens J. 2016;16(1):254–64.
43. Fakroon M, Alshahrani M, Gebali F, Traore I. Secure remote anonymous user authentication scheme for smart home environment. Internet of Things. 2020;9: 100158.
44. Nikravan M, Reza A. A multi-factor user authentication and key agreement protocol based on bilinear pairing for the internet of things. Wirel Pers Commun. 2020;111(1):463–94.
45. Nyangaresi VO, Rodrigues AJ, Abeka SO. Machine learning protocol for secure 5G handovers. Int J Wirel Inf Netw. 2022. https://doi.org/10.1007/s10776-021-00547-2.
46. Tewari A, Gupta B. A lightweight mutual authentication protocol based on elliptic curve cryptography for IoT devices. Int J Adv Intell Paradigms. 2017;9(2–3):111–21.
47. Mo J, Chen H. A lightweight secure user authentication and key agreement protocol for wireless sensor networks. Secur Commun Netw. 2019. https://doi.org/10.1155/2019/2136506.