PrivHome: Privacy-Preserving Authenticated Communication in Smart Home Environment

Geong Sen Poh, Prosanta Gope, *Member, IEEE* , and Jianting Ning

Abstract—A smart home enables users to access devices such as lighting, HVAC, temperature sensors, and surveillance camera. It provides a more convenient and safe living environment for users. Security and privacy, however, is a key concern since information collected from these devices are normally communicated to the user through an open network (i. e. Internet) or system provided by the service provider. The service provider may store and have access to these information. Emerging smart home hubs such as Samsung SmartThings and Google Home are also capable of collecting and storing these information. Leakage and unauthorized access to the information can have serious consequences. For example, the mere timing of switching on/off of an HVAC unit may reveal the presence or absence of the home owner. Similarly, leakage or tampering of critical medical information collected from wearable body sensors can have serious consequences. Encrypting these information will address the issues, but it also reduces utility since queries is no longer straightforward. Therefore, we propose a privacy-preserving scheme, PrivHome. It supports authentication, secure data storage and query for smart home systems. PrivHome provides data confidentiality as well as entity and data authentication to prevent an outsider from learning or modifying the data communicated between the devices, service provider, gateway, and the user. It further provides privacy-preserving queries in such a way that the service provider, and the gateway does not learn content of the data. To the best of our knowledge, privacy-preserving queries for smart home systems has not been considered before. Under our scheme is a new, lightweight entity and key-exchange protocol, and an efficient searchable encryption protocol. Our scheme is practical as both protocols are based solely on symmetric cryptographic techniques. We demonstrate efficiency and effectiveness of our scheme based on experimental and simulation results, as well as comparisons to existing smart home security protocols.

Index Terms—Smart home privacy, encrypted query, searchable encryption.

1 Introduction

Smart home can be loosely defined as a home with connected appliances with software controls. A user, using software installed in computing devices (such as mobile phone, tablet and laptop), is able to control and automate these appliances. The controllable appliances can be set to learn and change their behaviours accordingly and these define how "smart" the appliances are [1]. Example of such appliances are heaters, lights, cameras, doors, fridges, TVs, washing machines, as well as sensors that detect light, temperature, motion and humidity [2]. Furthermore, these devices can be controlled through smart home hubs such as Samsung SmartThings and Google Home using Wifi connection provided by the home gateway (i.e. Wifi router).

A common application scenario is home video surveillance. Cameras are installed in a home at strategic area. A user then remotely monitors his or her home through video streams and images transmitted from the cameras. Other compelling reasons for the adoption of smart homes include health monitoring, which is especially important with the aging populations in many countries, entertainment, better energy distributions and cost savings through smart metering [4] and control of heating and air conditioning (HVAC) [5], for example.

1.1 A Smart Home Architecture

A typical setup of a smart home is shown in Figure 1. This is an adoption of the smart home architecture presented in [3], [6], [7]. In general, a smart home system consists of *users* of the system, a *gateway*, a *service provider* and various *smart devices* that are seamlessly connected through a *computer network*. In more specific terms, a *gateway* can be a home router providing wifi connections to all the smart devices, which typically does not store information. The service provider is provision under a TD-SCDMA network that provides administrative and management functionalities, and may store user information. The user communicates with the service provider through a base station and the Internet, while the service provider communicate with the gateway via a secure channel. We have mentioned examples of smart devices, this includes sensors such as temperature sensor, as well as more capable devices such as smart hubs.

In a smart home setting, normally, a smart home service provider hosts services and an individual subscribes to the services. Alternatively, users may purchase the devices separately, install, register and connect to the services themselves. Once completed, a user can control the various devices through a mobile device. If a smart hub is deployed, then it serves as a smart device used to control many of the other smart devices. The communication between the smart devices and the user is thus via the smart hub.

1.2 Security and Privacy Issues in Smart Homes

All of the stated controls and activities involve collections of data from the sensors and smart devices, which are

- *G. S. Poh is with the NUS-Singtel Cybersecurity R & D Lab, National University of Singapore, Singapore.E-mail: geongsen@gmail.com*
- *P. Gope is with the Department of Computer Science, University of Sheffield, UK. E-mail: p.gope@sheffield.ac.uk*
- *J. Ning is with the School of Mathematics and Computer Science, Fujian Normal University, China and Department of Computer Science, National University of Singapore, Singapore, E-mail: jtning88@gmail.com*

Corresponding author: Prosanta Gope

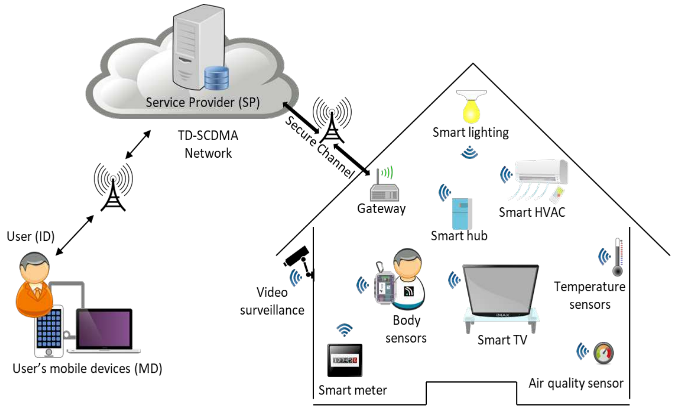

Fig. 1. A Smart Home Architecture

communicated through an open network (i. e. Internet) to the mobile devices via the service provider. These introduce significant security and privacy concerns. As discussed by Apthorpe *et al.* [8], sensitive information recorded by these devices include sleeping patterns, medical information, exercise routines, child behaviours and sexual activities. In fact, the surveys performed by Choe *et al.* [9] has shown that users are generally concerned about sensitive privacy information being recorded by the smart home devices.

Data-in-Transit. Since data are collected and shared through the Internet, unauthorized parties might attempt to access and learn the content of the data. A common solution is to deploy standard cryptographic protocols such as TLS/SSL to authenticate and secure data transmission. Nevertheless, these protocols may still be too computationally expensive for deployment in resource-constrained smart devices, as was discussed by Kumar *et al.* [6]. Due to this, in this work we propose a lightweight entity authentication and key establishment protocol to address security for data-in-transit, using only symmetric cryptographic primitives.

Data-at-rest. In a smart home system, data collected from a device may be stored in the device itself and/or in the cloud storage. For example, smart hub likes Amazon Echo and Google Home store voice clips in their cloud storage [10]. This raises concern on privacy of the stored data. As was discussed by Fernandes et al. [11], flaws were found on one of the smart hub cloud platform, which resulted in an attacker being able to access the underlying data and issue commands to different devices. Also, the service provider becomes high value target for adversary and can potentially be compromised. This means we should not assume the service provider to be fully trusted, in contrast to existing proposals, for example, in [7]. Instead, we assume the service provider to be semi-trusted. A mitigation approach is to encrypt all the data stored in the smart devices and at the service provider. However, using conventional encryption to encrypt data makes it difficult for a smart device (or the smart hub) to process the data and for the user to query them. Solution to this is to allow for privacy-preserving queries on encrypted data, which we proposed.

Host (sensors & smart devices). Finally, the sensors and smart devices must be secured as well. As was discussed by Choi *et al.*, during an event discovered by Proofpoint [12], more than 750,000 consumer devices were compromised and used to disseminate phishing and SPAM emails. Our entity authentication and key establishment protocol allows the user and the smart devices to communicate through a challenge-response secure session, instead of through a simple username password logins.

In summary, we envisage an active adversary who is an outsider that may attempt to study and modify the communication between the user, the service provider and the gateway. We also consider a semi-honest adversary (e.g. service provider, or the gateway) who study the communication between the user and the gateway (or the service provider) in order to learn information from the collected data.

1.3 Our Contributions

Smart home solutions typically protect data privacy through access control mechanism. If the mechanism is compromised, plain data can be accessed by an attacker. Furthermore, from a privacy standpoint, we would not want the service provider to learn our daily behaviour and activities. We thus envisage a stronger security guarantee in smart home system through data encryption, which means even when the access control mechanism is compromised, an attacker will not be able to make sense of the data he extracted successfully. Nevertheless, a user (or a gateway or the service provider) may still need to retrieve information or submit command to the smart devices. In order to perform these tasks even when the data is encrypted, we propose a mechanism for queries on encrypted data. We introduce an efficient scheme, PrivHome, that supports entity and data authentication, as well as secure data storage and privacy-preserving queries in a smart home environment. To the best of our knowledge, there has not been any work on privacy-preserving queries for smart home. Underlying our scheme is a new entity and key establishment protocol and an interactive searchable encryption protocol that we designed, combination of which resulted in our PrivHome scheme that fulfills the stated security goals. We note that the combination of the two protocols requires integrating query on encrypted data into the authentication and key establishment protocol efficiently. It also requires a definition on the security requirements and the combination to be performed in a secure and privacy-preserving manner. On the former, we show that, both asymptotically and through experimentation, our scheme is at least on par with the existing schemes in term of computation and bandwidth efficiency, yet provides both entity authentication and private query. On the latter, we achieve security as per our model through the derivation of authentication and searchable encryption keys from a key established between the entities, and further analyze security of the scheme based on the definition of the two protocols.

2 Related Works

The two main area of study related to our work are the existing proposals on solving the security and privacy issues in smart homes, and the techniques for queries on encrypted data. In the following we review state-of-the-art proposals in these two area.

2.1 Security and Privacy in Smart Homes

Komninos *et al.* [13] surveyed the security and privacy issues on smart home environment, focusing on the smart grid application. Nevertheless, Some of the security issues and countermeasures discussed can be applied on a general smart home environment. They categorized security objectives (e.g. confidentiality, authenticity, availability, non-repudiation) and listed potential countermeasures, such as using cryptographic techniques for confidentiality, and anonymization techniques for data privacy. Kumar *et al.* [6] proposed a lightweight session key-establishment and authentication scheme for smart home environment. With secure key establishment, the scheme provides data confidentiality during message transmission, mutual entity authentication and data integrity protection. The scheme is lightweight since it is designed using only symmetric cryptographic primitives. Song *et al.* [14] also proposed a secure protocol that achieves similar goals. Their main idea is for a server (i.e. the service provider) to generate secret keys for the devices based on chaotic map techniques. Then message authentication code based on the generated secret keys is deployed for encryption, entity and data authentication. In 2017, Kumar *et al.* [15] further proposed a key-establishment and authentication scheme that also supports anonymity of the smart devices, in order to prevent an outsider from learning the types of devices in a smart home. Independently, Wazid *et al.* [16] proposed an efficient secure authentication scheme specifically caters for resource-constrained devices. The underpinning idea is to first register the user and the smart devices through a registration authority. Subsequently authentication is performed based on the secret keys created during the registration phase. Efficiency is achieved through using only symmetric key primitives. In all these schemes, both the service provider and the gateway are assumed fully trusted. In contrast, our work focuses on semi-honest service provider and gateway. Chakravorty *et al.* [17] discussed in general a secure data collection framework that consists of an access control mechanism for authorization, as well as method for de-identification and re-identification of sensitive fields in a smart home record. They suggested the use of message authentication code for de-identification and an identifier map for re-identification, as well as k-anonymity technique in generalization of the information. Their focus is to provide for privacy-preserving data analytics and safeguard data privacy through access control. Lee *et al.* [18] proposed similar solution based on cryptographic hash function and access control but consider a smart community with hierarchical access instead. In contrast, we focus on providing data privacy, yet at the same time allowing the data to be queried in a privacy-preserving manner using cryptographic techniques.

Choi *et al.* [19] discussed attack on firmware of the smart devices, whereby an attacker is able to cause malfunctions on the smart devices through exploitation on the firmware. They propose a scheme for verifiable and validated update on firmware as a way to mitigate this issue. With such a scheme, a firmware can be quickly updated, therefore reducing the window of attack. Underlying their scheme is the combination of pairing-based computation, Schnorr's signature scheme and symmetric-based hash chain. Apthorpe *et al.* [8], [20], [21] demonstrated that information collected and transmitted through the Internet from the smart devices can be inferred by the Internet service provider or observers of the network.Their attack utilises traffic metadata. It involves first learning the device type through device fingerprint, and then infer user information through traffic rates. As stated, knowing the device type (e.g. a pacemaker) in itself is considered a privacy leakage. They suggest using the combination of VPN tunneling and traffic shaping to protect against such attack. In another independent work that considers similar problem (i. e. attack based on network traffic even when the information are encrypted), Liu *et al.* [22] proposed a privacy-preserving traffic obfuscation framework based on differential privacy to thwart traffic analysis attack. Their framework protect against information sent/received from home gateways in a smart community environment. We remark that our scheme protect data privacy and authenticity directly through cryptographic mechanism, and can be combined with the schemes proposed by Choi *et al.* [19] and Apthorpe *et al.* and Liu *et al.* to arrive at a comprehensive privacy-preserving smart home solution that also protect against side-channel attack.

2.2 Queries on Encrypted Data

Song, Wagner and Perrig [23] introduced a practical and efficient searchable encryption scheme using only symmetric primitives. However their scheme requires sequential scan on every encrypted word in a file in order to search for the keyword matching a query. Goh [24] then proposed a secure index-based approach that does not require scanning of every words in a file, but the work effort is still linear to the total number of files for each query irregardless to the number of files matching the query, which can be of small size. Following from these, Curtmola *et al.* [25] proposed a comprehensive formal security model, together with two index-based schemes provably secure under this model. Chase and Kamara [26] then generalised the model for SE scheme that works on arbitrarily-structured data including social network and graph data. Since then, many variants were proposed by adapting the index structure presented by Curtmola *et al.* and Cash and Kamara or inventing new ones. There are schemes that provide richer functionalities, such as Boolean search [27], dynamic Boolean search [28], ranked search [29], [30] as well as semantic-aware scheme [31]. There are also schemes providing better security with less leakage and forward privacy [32], hide access pattern using private information retrieval in a two-server setting for large-scale database [33], as well as efficient blind storage where the server acts only as a commodity server offering only upload and download services [34]. Most of the searchable encryption schemes achieve practicality by allowing leakage of certain information such as access patterns. Analysis of leakage on searchable encryption schemes was first studied by Islam *et al.* [35]. Cash *et al.* [36] then proposed a framework that define the security level of a scheme based on the information being leaked, and an improved analysis. Various analysis have since been presented following these two seminal works that analyse searchable encryption schemes. They include analysis based on active attack [37], passive attacks [38], and analysis on scheme with specific functionality such as range queries based on order-preserving or order-revealing techniques [39], [40].More recently, given both the existing scheme and attack cases, practical schemes with better functionalities, performance and security properties have been proposed. These includes the Boolean search scheme [41], the scheme with improved locality [42] and a leakage suppression scheme [43]. Comprehensive surveys on searchable encryption can be found in [44], [45], [46].

In terms of encrypted queries for smart systems, Wen *et al.* [47], [48] proposed novel and effective mechanisms in providing queries on encrypted data for smart grid system, which is related to our work in the smart home environments. Our underlying motivation is similar to theirs, whereby data privacy must be protected when the various measurements are queried and retrieved from the smart grid (e.g. smart meters) to prevent an attacker from learning the behaviours and activities of a household. The proposals for smart grid deploy public-key based cryptographic primitives (e.g. pairings), which is compute intensive compared to schemes utilizing only symmetric primitives. Their consideration, however, is slightly different from ours, in that range queries must be provided for financial auditing. In our case, we consider queries that require equality match. For instance, an encrypted query on an appliance and one of its functionality (e.g. temperature) would return the current value in encrypted form. Hence, we use symmetric-based searchable encryption instead. In our searchable encryption protocol, we adopt and tweak the index-structure of the recent existing searchable encryption scheme, in which details of the scheme is described in Figure 2. We also develop our model based on the well-established formal security model of searchable encryption. These allow us to design a practical scheme suitable for use with smart devices (sometimes with constrained computation and storage capabilities), while providing security assurance through the security model.

3 Definition and Assumption

In this section, we define the main entities involved in our scheme, and the security goals that we aim to achieve.

3.1 Entities

Our scheme involves four main entities. These are:

- A *user*, $User_i$, who installs and uses a smart home system. We assume the user to be fully trusted. The user also has a mobile device MD_i that has a software installed for the control of the smart home system. We also assume the mobile device and software are fully trusted.
- A *service provider*, SP, which provides the smart home system. We assume the service provider to be semi-trusted, in contrast to many existing studies, such as the proposals presented in Section 2.1. Our main focus in this work is such that the storage of data collected from the smart devices, and queries of these information are secure and privacy-preserved. In other words, the service provider does not learn any information from the stored data and the queries by the user.
- A home *gateway*, which manages, and communicates data to/from the sensors and smart devices. Here, the gateway represents a home router. In contrast to most of the work discussing security in smart homes (as presented in Section 2.1) where the gateway is fully trusted, here we present our work considering the case where the gateway is semi-trusted. This is to simulate the environment where the gateway is corrupted and an adversary is able to persistently listens and captures the communications between the gateway and the other entities. We further note we assume the secret key of the gateway is stored securely.
- Many *smart devices*, which execute command and collect information as per the functionality of the devices (e.g. HVAC, lights, temperature). These devices are trusted in a way that we assume physically compromising these devices is infeasible so that the underlying secret keys for entity authentication and searchable encryption remain protected.

All of which communicate among one another through an open communication network (i.e. home wireless network and the Internet).

3.2 Security Goals

There are four main goals that we aim to address through our scheme, based on our discussions on the security and privacy issues in smart homes in Section 1.2:

- *Data confidentiality.* An adversary should not be able to learn any information from the communicated or stored data.
- *Privacy-preserving queries on encrypted data.* An adversary should not be able to learn the queries, or learn the underlying data based on these queries except for the access and query patterns.
- *Data authentication.* Authenticity of the transmitted data is verifiable. In other words, the entities involved in the scheme (e.g. the user, the gateway, or the service provider) is able to detect modification made by an adversary (an outsider).
- *Entity authentication.* Identities of the parties involved are verifiable and an adversary success probability of masquerading an authorized parties should be negligible.

Our security goals follow the main security objectives for smart homes (except privacy-preserving queries that we define), as discussed by Komninos *et al.* [13]. Here we assume the network is always available, and authorization is such that once the user successfully authenticated himself/herself, he/she has access to all the information available in a smart device, which is common in a smart home setting.

4 PrivHome: Overview & Protocols

4.1 Overview

We now consider an application model for intelligent home network service. The model is composed of a set of mobile users, home gateways, a set of smart home devices, and a service provider that is implanted into the TD-SCDMA network. Our goal is to allow a user with a mobile device to access his home network and communicate with specific home devices via the existing TD-SCDMA network. And the general process is carried out as follows. The user with the mobile device communicates with its nearby base station and requests home network service from the service provider. After receiving the user's request, the service provider contacts the corresponding home gateway through the dedicated secure line (such as X2 interface in 5G HetNet [49]) and sends data request to the home gateway. Then, the home gateway reads the real-time information of the home devices through the one-hop Wireless (802.11) and sends back the information to the service provider. Next, the service provider transmits the information to the home user. Now in order to achieve the security goals stated in Section 3.2 in this application model (depicted in Figure 1) here we introduce two protocols which are the core of our scheme. An authenticated key-establishment protocol negotiates a session key between the entities, that is later used to initiate an instance of a searchable encryption protocol, all of which use only symmetric techniques in order to achieve practical performance. Our scheme can be bootstrapped by each smart device, the gateway and the user sharing a secret key, for example, through an out-of-band channel. This can be performed during the installation of the smart devices, where a key is set by the user.

Our authenticated key-establishment protocol utilizes the secret key to provide mutual authentication between the involved entities and agree on a session key for secure communication and queries of encrypted data. Our searchable encryption protocol, on the other hand, derives an index masking key from the secret key, to construct a searchable inverted index for the smart devices. The index is stored on each of the devices. During queries, the authenticated entities search the masked index using an encrypted queries that is communicated through the secure session.

While one may adopt existing authenticated key-establishment protocol and searchable encryption protocol, we note that the construction and combination of the two protocols are not as straightforward as it seems. Firstly, there is additional computation due to the introduction of private query to preserve data privacy using searchable encryption. We address this issue by designing our searchable encryption protocol based on the efficient inverted index approach. In this approach, we only use symmetric encryption scheme and pseudo-random function. Our index construction bares similar structure and property as in the practical scheme proposed by Cash *et al.* [28], which has been shown to be very efficient even for large database. For instance, their most efficient scheme requires slightly more than 100ms to return $10,000$ query results for a database with 10^8 items using blades with Dual Intel Xeon 2.4GHz E5620 processors. In our case our datasets would be much smaller considering the functionalities of the devices in a smart home. Secondly, there is the difficulty in the fact that authentication and key-establishment must be performed between at least three entities (i.e. the user, the gateway and the smart device) instead of the commonly assumed two-party case. Added to that, such operations must also ensure confidentiality and integrity of the underlying queries. For this, we integrate our authenticated key-establishment protocol and the searchable encryption protocol in such a way that both the security properties are preserved. We achieved this by integrating the query operations and query results in the authenticated session established through the key-establishment protocol. Thirdly, in the case of searchable encryption, the data structure in smart home is slightly different from the common keyword-file (or record) pair as in most schemes. Instead of directly querying a keyword, the query must first identify the specific device to be queried, and then query the functionality and retrieve the data of that functionality of the device. This means we cannot apply directly existing searchable encryption scheme. Furthermore, data in a smart home device is updated frequently, as opposed to the static database and occasional updates considered by most schemes. We address these issues by constructing a new searchable encryption scheme, in which the underlying masked index structure uses randomized device identifier as key that links to the encrypted values of the device's many functionalities, all using only symmetric key techniques. This also means update on the values is efficient, that involves only matching the device's identifier and replacing the existing stored encrypted value with the new encrypted value. In the followings we present the detailed design of the two protocols.

4.2 A Lightweight Authenticated Key-Establishment Protocol

Our authenticated key-establishment (AKE) protocol involves a user ($User_i$), who has a mobile device MD_i with Internet connectivity, requests through the home gateway to know the status of a particular smart device (e.g. a home appliance). In this context, the home gateway will assist both the smart device and the user to establish a session key. We assume the home gateway and the smart device share a secret key k_{gd}. The pre-sharing of key can be performed through an out-of-band channel (e.g. by the user during setting up the device) or in a similar way as described in [6], [16]. The secret key is used to assist three-party authentication process between the user, the gateway and the smart device. We also assume the service provider and the home gateway share a secret key k_{gs}, for mutual authentication between the service provider and the home gateway for data collections and storage at the service provider. The service provider may generates the key a priori and embed the secret key securely in the gateway before delivering the system to the user. Our proposed scheme consists of the following two phases: *setup phase* and *authentication phase*.

4.2.1 Setup (Setup) Phase

The user $User_i$ register a smart device SD_q based on the following steps.

Step S1: $User_i$ registers his/her identity ID_i to the smart device SD_q through a secure (out-of-band) channel.

Step S2: After receiving the registration request, the smart device SD_q randomly generates a unique shadow identity SID_q, a secret key k_i, and also generates a set of pseudo identities PID $=\{pid_1,pid_2,...,pid_n\}$, which are later used in case of loss of synchronization between the device and the $User_i$. Next, SD_q composes a message with

$\{SID_q,\ k_i,\ PID\}$ and sends to the $User_i$ through the said secure channel. Finally, SD_q stores $\{SID_q,\ k_i,\ PID\}$ in its storage for further interaction with the $User_i$. Here SID_q, k_i, PID can be represented as binary strings, and randomly generated using a pseudo-random function with security parameter λ.

Step S3: Upon receiving $\{SID_q,\ k_i,\ PID\}$ from SD_q, the user gives his/her thumb impression β_i and password psw_i and computes α_i=$h(\beta_i)$, ∂_i=$h(\alpha_i||psw_i)$, k_i^*=$k_i \oplus h(\beta_i||psw_i)$. Here h is a hash function and we assume the thumb impression β_i is unique for each user, and is identical every time a user provides it. A fuzzy extractor may be used, if we assume there is slight differences in β_i every time it is captured. Finally, the user $User_i$ stores $\{\partial_i,\ SID_q,\ k_i^*,\ PID\}$ into his/her mobile device for communication with the smart device SD_q at a later stage.

4.2.2 Authentication (`Auth`) Phase

To achieve communication security, a user $User_i$ needs to go through an authentication and the key-establishment process each time before obtaining services from the gateway. The proposed scheme consists of the following steps:

Step A1: $User_i$ inserts his/her thumb impression β_i and password psw_i into his/her mobile device MD_i and then the user's mobile device calculates α_i=$h(\beta_i)$, $\partial_i^{'}$ =$h(\alpha_i||psw_i)$ and checks whether $\partial_i^{'} \overset{?}{=} \partial_i$. If the validation is unsuccessful, MD_i will abort the authentication process. Otherwise, MD_i calculates k_i=$k_i^* \oplus h(\beta_i||psw_i)$. After that, the user generates a nonce N_i and subsequently, finds his/her location through GPS and gets the location area identity LAI_i based on the latitude and longitude of the location, where he/she is positioning. Hereafter, he/she computes $EL = LAI_i \bigoplus h(k_i||N_i)$, a key-hash response V_1=$h(SID_q||N_i||k_i||EL)$ and subsequently composes a message $\mathrm{M}_{\mathrm{A}_1}$:$\{SID_q,\ N_i,\ EL, V_1\}$ and sends to the gateway *via* service provider.

Remark: Here, the reason behind using location identifiers (LAIs) is that, initially the user can select the possible locations with location identifiers (LAIs) [50], from where he/she may request for services and then sends the list of possible locations to the home gateway. Now, during the execution of the authentication phase if the gateway finds LAI_i produced by the user is not in the list, then the gateway will ask the service provider to alert the user and also ask him/her to confirm the location. Besides, in our proposed scheme, because of the usage of the random number N_i in LAI_i an adversary will not be able to distinguish the multiple requests of the same user from a particular location. In this way, we retain the untraceability property. Now, in order to prevent an adversary from delivering incorrect information about the current position of the home user, the message sources must be verified. In our proposed scheme, the GPS server which helps the user to locate himself/herself, is assumed to be trusted. Therefore, using a shared secret key between the GPS server and the user, here we can ensure message source authentication. There are existing literatures [51], however these solutions utilized RSA-based sign-encryption technique, which could be infeasible for the resource constrained mobile devices.

Step A2: Upon receiving the message $\mathrm{M}_{\mathrm{A}_1}$, the gateway generates a random number N_g and calculates V_2=$h(ID_g||N_g||K_{gd}||LAI_g)$, where LAI_g denotes the location of the gateway and K_{gd} represents the share secret key between the gateway and the device SD_q. Next, the gateway composes a message $\mathrm{M}_{\mathrm{A}_2}$:$\{\mathrm{M}_{\mathrm{A}_1},\ ID_g,\ N_g, LAI_g,\ V_2\}$ and sends it to the device SD_q.

Step A3: After receiving the message $\mathrm{M}_{\mathrm{A}_2}$, the device SD_q first locates SID_q in its database and then computes and validates the key hash responses V_1,V_2. Next, the gateway decodes LAI_i from EL and then compares and validate LAI_i with the LAI_g . If the validation is successful, SD_q generates a session key SK, and a new shadow identity SID_q^{new} and then computes SID_q^{new*}=SID_q^{new} $\bigoplus h(SID_q\ ||\ k_i)$, SK_i=$h(ID_i||k_i||N_i) \bigoplus SK$,$SK_g$=$h(ID_g||K_{gd}||N_g) \bigoplus SK$, V_3=h$(SK_g||K_{gd}||N_g)$, V_4=$h(SK_i||k_i||SID_q^{new*})$. Next, SD_q composes a message $\mathrm{M}_{\mathrm{A}_3}$:$\{(SID_q^{new*},\ SK_i, V_4)||(SK_g,\ V_3)\}$ and sends $\mathrm{M}_{\mathrm{A}_3}$to the gateway.

Step A4: Upon receiving the response message $\mathrm{M}_{\mathrm{A}_3}$ from the device SD_q, the gateway first computes and validates the key-hash response V_3. If the validation is successful, the gateway decodes the session key SK=$h(ID_g||K_{gd}||N_g) \bigoplus SK_g$ and composes a new message $\mathrm{M}_{\mathrm{A}_4}$:$\{(SID_q^{new*},\ SK_i, V_4)\}$and then send it to the user $User_i$.

Step A5: After receiving the message $\mathrm{M}_{\mathrm{A}_4}$, $User_i$ first verifies the key-hash response V_4. If the validation is successful, $User_i$ computes and decodes the session key SK=$h(ID_i||k_i||N_i) \bigoplus SK_i$, and the new shadow identity SID_q^{new}=SID_q^{new*} $\bigoplus h(SID_q||k_i)$ for the next round.

Note that, if any steps of the above validation process is unsuccessful, then entities involved in this protocol will abort the execution of the scheme. In case of loss of synchronization, instead of the shadow identity SID_q, the user $User_i$ needs to select one of the unused pseudo identities pid_x from PID =$\{pid_1, pid_2, ..., pid_n\}$ and send it in the message $\mathrm{M}_{\mathrm{A}_1}$. On receiving this message and after successfully validating the user, the device SD_q generates a new shadow identity and securely sends it in the message $\mathrm{M}_{\mathrm{A}_3}$ by using the secret key k_i. During the authentication process, both the $User_i$ and the device deletes the used pseudo identity pid_x from their storage.

Lastly, we present the authentication and key-establishment process between the service provider SP and the gateway. This is to facilitate data submission to the storage of the service provider. A smart home system, for example, may potentially store voice data in the storage of the service provider. This is to facilitate data processing (e. g. voice commands) and return the results to the smart device. In this instance, the queries and the data should be protected from being listen to or modify by an outsider. Furthermore, in the interest of user privacy, the data and the queries should not be learnt by the service provider. In the following we demonstrate how this can be done. We state the authentication protocol operated between the device, the gateway and the service provider based on standard one pass mechanism.

Step A*1: The smart device SD_q generates a sequence number N_q, and computes $V_1^* = \mathtt{Enc}(k_{gd}, N_q, ID_g, (\tau_q, e_q))$, where k_{gd} is a secret key shared between SD_q and the gateway, ID_g the unique identifier of the gateway and (τ_q, e_q) the tuple of the encrypted searchable index and the list of encrypted attribute values (which will be discussed in the section on our searchable encryption protocol). The smart device SD_q then sends $M_{A_1^*} : \{V_1^*\}$ to the gateway.

Step A*2: Upon receiving $M_{A_1^*} : \{V_1^*\}$, the gateway decrypts V_1^* using the shared key k_{gd}. If the decryption is successful, the gateway generates a sequence number $N_{g'}$ and computes $V_2^* = \mathtt{Enc}(k_{gs}, N_{g'}, ID_{sp}, (\tau_q, e_q))$, where k_{gs} is a secret key shared between the gateway and the service provider SP, and ID_{sp} the unique identifier of SP. The gateway then sends V_2^* to the service provider SP. The service provider SP decrypts V_2^* using k_{gs} and stores (τ_q, e_q). We remark that in our protocol (Figure 3), we do not depict the encrypted queries from the gateway to the service provider, but only show encrypted queries by the user through the gateway to the smart devices. This is because many smart appliances (e.g. lighting, HVAC, surveillance camera) can be controlled directly between the user mobile device and the gateway. However, we note that the protocol can be adapted to perform encrypted queries through the service provider instead, by a two steps process where the user query through the authenticated gateway, and then the gateway send the query to the authenticated service provider [52].

4.3 An Efficient Searchable Encryption Protocol

In this section we describe our searchable encryption ($\mathtt{SSE}$) protocol. We utilize a symmetric encryption scheme $\mathcal{E}$ and a pseudorandom function H, which we define in the followings. A randomized symmetric encryption scheme $\mathcal{E}$ = ($\mathtt{Gen}$, $\mathtt{Enc}$, $\mathtt{Dec}$) consists of three PPT algorithms. $\mathtt{Gen}$ takes λ and outputs a secret key K; $\mathtt{Enc}$ takes K and a message $d \in \{0,1\}^*$ and outputs a ciphertext c; For all K from $\mathtt{Gen}$ and $d \in \{0,1\}^*$ we have $\mathtt{Dec}(K, \mathtt{Enc}(K,d)) = d$ with probability 1. We say $\mathcal{E}$ is IND-CPA if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{E},\mathcal{A}}^{ind-cpa}(\lambda) = |\Pr[\mathcal{A}^{K \leftarrow \mathtt{Gen}(1^\lambda), c \leftarrow \mathtt{Enc}(K,d)} = 1] - \Pr[\mathcal{A}^{c \xleftarrow{R} \{0,1\}^*} = 1]|$ is negligible. A function $H : \{0,1\}^\lambda \times \{0,1\}^* \rightarrow \{0,1\}^\lambda$ from $\mathcal{H}$ the set of all functions $\{0,1\}^* \rightarrow \{0,1\}^\lambda$ is pseudo-random if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{H,\mathcal{A}}^{prf}(\lambda)$ $= |\Pr[\mathcal{A}^{H(K,.), K \xleftarrow{R} \{0,1\}^\lambda} = 1] - \Pr[\mathcal{A}^{g(.), g \xleftarrow{R} \mathcal{H}} = 1]|$ is negligible. Detailed treatments of these primitives can be found in [53].

Our protocol involves a user $User_i$ submitting a query to a smart device via the home gateway. Let $\lambda \in \mathbb{N}$ be the security parameter and $||$ as concatenation. The user owns a set of smart devices, $\mathbf{SD} = \{SD_1, SD_2, \ldots, SD_n\}$, and each device SD_q maintains a list of functionalities (e.g. on/off, high/low, values) $\mathbf{F}_q = (f_{q,1}, f_{q,2}, \ldots, f_{q,h})$ for $1 \leq q \leq n$, and each functionality contains value represented as a string, $d_{q,j} \in \{0,1\}^{\phi(\lambda)}$ where ϕ is a polynomial function in λ. Here we denote the list of values as $\mathbf{d}_q = (d_{q,1}, d_{q,2}, \ldots, d_{q,h})$. In our work, we assume every device has equal number of functionalities. In practice this may not be the case, but in order to hide the possibility of a service provider being able to guess the underlying device based on just the differences in the number of functionality, we can pad each device with dummy values. Similarly, the value stored in each functionality is padded to the same length. This means all the encrypted values across different devices will have the same size, thus preserving the privacy of the data and the device from adversary with background knowledge from learning the types of devices via the differences in size of the encrypted values.

For each device SD_q, we say there is an index table of the form $I_q = ID_q||\mathbf{F}_q$, where ID_q is the unique identifier of the device. Given a pair of value $(ID_q, f_{q,j})$, I_q returns the value $d_{q,j}$. We also define $\{1, 2, \ldots, n\}$ as $[n]$, $x \leftarrow A$ to mean x is an output of an algorithm A and $x \xleftarrow{R} X$ to mean random selection of a value x from a set X. We use $\mathtt{negl}(x)$ to denote a negligible function. Now we define our searchable encryption protocol.

Definition 1. *We define a searchable encryption protocol* $\mathtt{SSE}$ *as consisting of the following phases:*

- $K \leftarrow \mathtt{KeyGen}(1^\lambda)$. *It outputs a secret key k and a data encryption key k_e. Set $K = \{k, k_e\}$.*
- $(\tau_q, \mathbf{e}_q) \leftarrow \mathtt{Update}(K, I_q, \mathbf{d}_q)$. *Given as inputs the secret key K, the index I_q and the list of values $\mathbf{d}_q$ of the functionality in I_q, the protocol returns a masked (or encrypted) I_q as τ_q and the list of encrypted value $\mathbf{e}_q = (e_{q,1}, e_{q,2}, \ldots, e_{q,h})$.*
- $(\rho_q, \mu_{q,j}) \leftarrow \mathtt{GenToken}(K, ID_q, f_{q,j})$. *Given as inputs the secret key K, the device identifier ID_q, the functionality $f_{q,j}$, it outputs an encrypted tuple $(\rho_q, \mu_{q,j})$, where ρ_q is the masked (or encrypted) value of ID_q, and $\mu_{q,j}$ the masked (or encrypted) value of $f_{q,j}$.*
- $e_{q,j} \leftarrow \mathtt{Query}(\rho_q, \mu_{q,j}, \tau_q, \mathbf{e}_q)$. *Given as inputs the masked tuple $(\rho_q, \mu_{q,j})$, the masked index τ_q and the encrypted list of values $\mathbf{e}_q$, it returns the matched encrypted value $e_{q,j}$.*

The $\mathtt{SSE}$ *protocol is said to be correct if* $\mathtt{Query}$ *returns the correct encrypted value that matches the functionality of the device that is being searched for. Under the real-ideal paradigm and a leakage profile $\mathcal{L}_{\mathtt{SSE}} = (\mathcal{L}_{\mathtt{SSE}}^{update}, \mathcal{L}_{\mathtt{SSE}}^{query})$, the* $\mathtt{SSE}$ *protocol is $\mathcal{L}_{\mathtt{SSE}}$-secure against chosen query attacks if for all PPT adversary $\mathcal{A}$, there exists a PPT simulator $\mathcal{S}$ such that $\mathbf{Adv}_{\mathtt{SSE},\mathcal{A},\mathcal{S}}(\lambda) = |\Pr[\mathbf{Real}_{\mathtt{SSE},\mathcal{A}}(\lambda) = 1] - \Pr[\mathbf{Ideal}_{\mathtt{SSE},\mathcal{A},\mathcal{S}}(\lambda) = 1| \leq \mathtt{negl}(\lambda)$ where the game is as follows:*

$\mathbf{Real}_{\mathtt{SSE},\mathcal{A}}(\lambda)$. *Given I_q, for $1 \leq q \leq n$, the challenger executes* $\mathtt{Update}$, *where the resulting $(\tau_q, \mathbf{e}_q)$ are given to $\mathcal{A}$. $\mathcal{A}$ then makes a polynomial number of adaptive queries to the* $\mathtt{Query}$ *protocol. For each query the challenger returns the query result to $\mathcal{A}$. Finally $\mathcal{A}$ returns a bit b as the output of the experiment.*

$\mathbf{Ideal}_{\mathtt{SSE},\mathcal{A},\mathcal{S}}(\lambda)$. *The simulator $\mathcal{S}$ simulates $(\tau_q, \mathbf{e}_q)$ based on the leakage information from $\mathcal{L}_{\mathtt{SSE}}^{update}$, and gives $(\tau_q, \mathbf{e}_q)$ to $\mathcal{A}$, who then makes a polynomial number of adaptive queries. The simulator $\mathcal{S}$ returns the query result for every query based on $\mathcal{L}_{\mathtt{SSE}}^{query}$ to $\mathcal{A}$. Finally $\mathcal{A}$ returns a bit b as the output of the experiment.*

Now we present a concrete construction, which is specified in Figure 2. We note that if there is an identical functionality in different smart devices, in order to hide such information, the user can keep a list and compute the masked functionality by $H(k_{D_q,2}, f_{q,j}, 1)$ for the first occurrence, $H(k_{D_q,2}, f_{q,j}, 2)$ for the second and so on.

Proposition 1. We say that our protocol $\mathtt{SSE}$ is $\mathcal{L}_{\mathtt{SSE}}$-secure if the underlying symmetric encryption scheme $\mathcal{E}$ and the pseudorandom function H are secure.

Proof. We state security of the protocol based on the construction of a simulator $\mathcal{S}$ in such a way that an

```
$K \leftarrow \texttt{KeyGen}(1^\lambda)$:
  1) Randomly generate a secret key $k$ and a data encryption key $k_e$.
  2) Set $K = (k, k_e)$.
$(\tau_q, \mathbf{e}_q) \leftarrow \texttt{Update}(K, I_q, \mathbf{d}_q)$
  1) Generate $k_{D_q} = H(k, ID_q)$.
  2) Generate $k_{D_q,1} = H(k_{D_q}||1)$ and $k_{D_q,2} = H(k_{D_q}||2)$.
  3) Extract $ID_q$ from $I_q$. Compute a masked identifier, $h_{ID_q} = H(k_{D_q,1}, ID_q)$.
  4) Extract $\mathbf{F}_q$ from $I_q$. For every functionality $f_{q,j}$, compute a masked functionality as $h_{f_{q,j}} = H(k_{D_q,2}, f_{q,j})$. (Here we assume every functionality is unique).
  5) Encrypt every value of the functionality as $e_{q,j} = \texttt{Enc}(k_e, d_{q,j}, r_{q,j})$, where $r_{q,j}$ is a pseudo-random nonce. (Every update of the value a different pseudo-random nonce is used to mask identical value).
  6) Set $\tau_q = h_{ID_q}||(h_{f_{q,1}}, \ldots, h_{f_{q,n}})$.
  7) Set $\mathbf{e}_q = (e_{q,1}, \ldots, e_{q,n})$
$(\rho_q, \mu_{q,j}) \leftarrow \texttt{GenToken}(k_{D_q}, ID_q, f_{q,j})$:
  1) Generate $k_{D_q,1} = H(k_{D_q}||1)$ and $k_{D_q,2} = H(k_{D_q}||2)$.
  2) Compute $\rho_q = h_{ID_q} = H(k_{D_q,1}, ID_q)$.
  3) Compute $\mu_{q,j} = h_{f_{q,j}} = H(k_{D_q,2}, f_{q,j})$.
$e_{q,j} \leftarrow \texttt{Query}(\rho_q, \mu_{q,j}, \tau_q, \mathbf{e}_q)$:
  1) If $\rho_q = h_{ID_q}$ in $\tau_q$, retrieve $(h_{f_{q,1}}, \ldots, h_{f_{q,n}})$.
  2) If $\mu_{q,j} = h_{f_{q,j}}$, retrieve $e_{q,j}$.
  3) return $e_{q,j}$.
```

Fig. 2. SSE: An Efficient Searchable Encryption Protocol

adversary $\mathcal{A}$ is not able to distinguish between the real execution of the protocol or a simulated one. We first define the leakage profile of our protocol.

$\mathcal{L}_{\mathsf{SSE}}^{update} = (n, \mathbf{F}_q)$: For the update operation, the number of smart devices n, and the number of functionalities $\mathbf{F}_q$ of each device SD_q is known to an adversary. Recall that we assume each smart device has equal number of functionalities to avoid an adversary from learning the type of smart device based only on the differences of the number of functionalities in each device. Since the masked identifier h_{id_q} in the real execution is a pseudorandom string, $\mathcal{S}$ is able to simulate an identifier by selecting a random string of the same size. Similarly, every masked functionality is also a pseudorandom string, hence simulation can be performed in a same way. What is left to be simulated is the encrypted values of the functionalities. Since every encrypted value is also a pseudorandom string, $\mathcal{S}$ simulates the value by selecting a random of the same size. It follows that an ideal execution of the update operation is distinguishable with only negligible probability, since $\mathcal{S}$ is able to simulate the masked identifier, the masked functionality and the underlying encrypted values.

$\mathcal{L}_{\mathsf{SSE}}^{query} = ((\rho_q, \mu_{q,j}), e_{q,j})$: In the case of query, the protocol reveals the query tuple $(\rho_q, \mu_{q,j})$ containing the masked identifier and the masked functionality as well as the access patterns (the returned encrypted value). In order to simulate the result of the query, the simulator $\mathcal{S}$ first creates a list that stores the matching query and the returned value. This allows $\mathcal{S}$ to simulate query that has been performed previously. When a query is made, $\mathcal{S}$ checks whether the query has been made previously. If this is the case, $\mathcal{S}$ retrieves and returns the matching query from the list to the user. If not, $\mathcal{S}$ randomly selects an encrypted value from one of the masked index and returns the value. The query and the output of this new query is then added to the list. □

We note that in the above statement the adversary learns the leakage profile only if it has access to the final stored masked and encrypted index as well as the encrypted query. Since in our scheme, the update and query operations are secured under an authenticated key exchange protocol that we proposed, an outsider will not be able to even learn the leakage information. This is also the case if the provider does not store the masked indexes and encrypted values. Furthermore, the indexes and the values stored in the encrypted form in the smart devices prevent an adversary that managed to access these information from learning the underlying values except for the leakage profile.

5 PrivHome: The Proposed Scheme

In this section we present our scheme that utilizes the protocols presented in the previous section. Figure 3 shows the detailed steps of the scheme, consisting of three phases: *Setup*, *entity authentication and key establishment*, and *private queries on encrypted data*. We note that each entity has an identifier denoted as ID_x, where $x \in \{i, g, q, sp\}$. The gateway shares a secret key k_{gd} with a smart device, and shares a secret key k_{gs} with the service provide *SP*. This secret keys can be setup using an out-of-band channel, such as during installation of the gateway and the smart device. The scheme combines the AKE and SSE protocols to achieve the security goal and address the issues stated in Section 4 as follows. The scheme executes SSE.KeyGen to first generate the secret key $K = (k, k_e)$ for data encryption and query, and transport K as part of the message to the user's device based on the setup protocol of AKE. The scheme then establishes a session key SK and an authenticated session between the user, the gateway and a smart device. The session key SK is used to establish a secure channel for transmitting the encrypted queries and receiving the query results. By doing so we link the queries to the authenticated session so that the device is assured that the queries are from an authorized user. Furthermore, even in the event that the session is compromised, the adversary will only have access to the encrypted token and encrypted data. Without the encryption keys, the adversary learns nothing about the query and the underlying data except for the leakage due to the searchable encryption scheme, as formalized in Section 4.3.

5.1 Setup

In this phase, an instance of the authenticated key-establishment protocol AKE, as well as the searchable encryption protocol SSE is instantiated by the scheme.

SSE: The smart device SD_q with identifier ID_q executes the scheme by first calling SSE.KeyGen, to generate a secret key k_i for $User_i$ and a data encryption key k_e that is used to encrypt the underlying data and measurements of the devices. Both keys are stored securely in SD_q's secure storage. The smart device SD_q then runs SSE.Update to create a masked index τ_q for the device q. For every time interval where device SD_q updates its data, it runs SSE.Update to

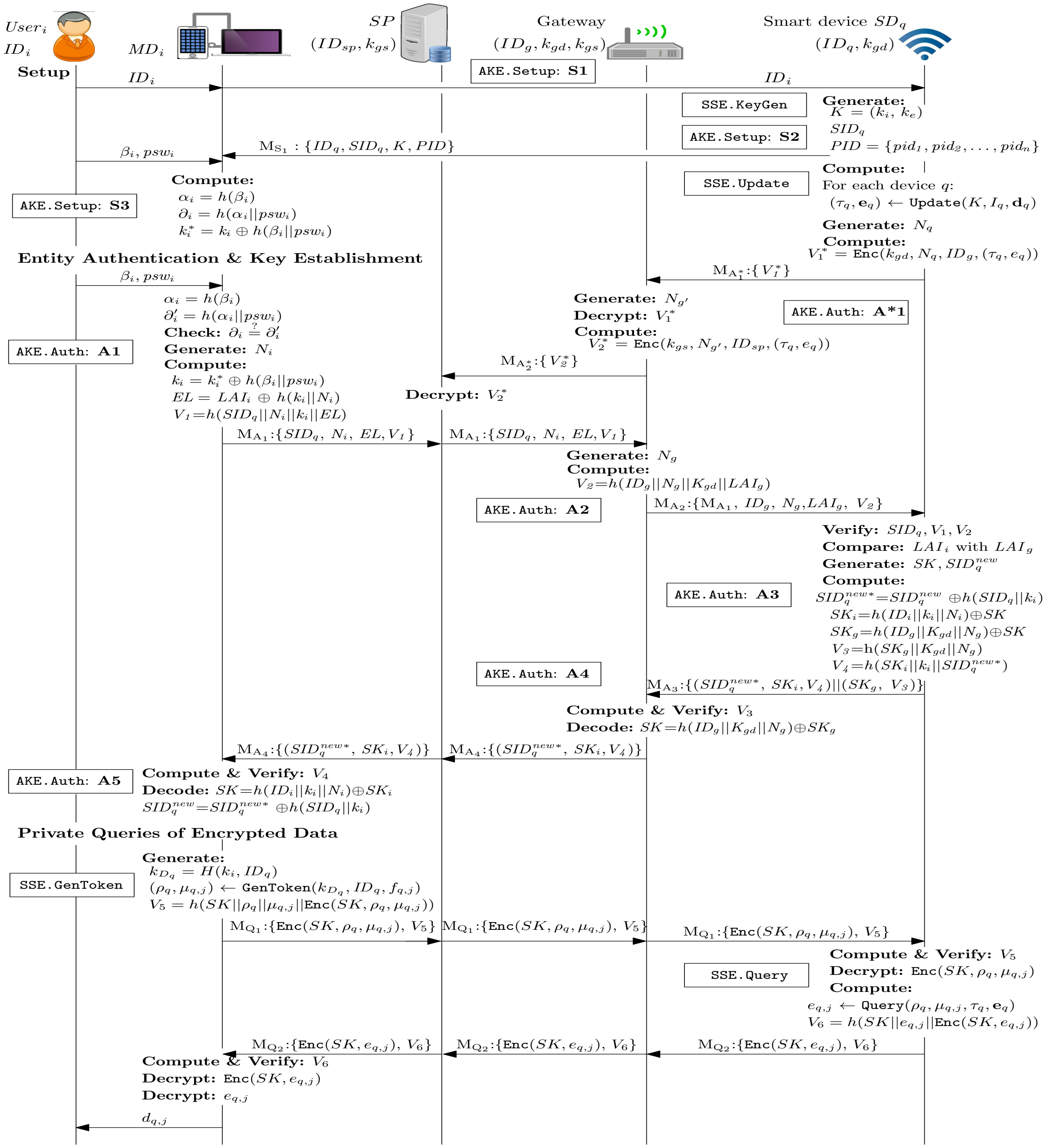
$User_i$
ID_i
MD_i
SP
(ID_{sp}, k_{gs})
Gateway
(ID_g, k_{gd}, k_{gs})
Smart device SD_q
(ID_q, k_{gd})
Setup
ID_i
AKE.Setup: S1
ID_i
SSE.KeyGen
Generate:
$K = (k_i, k_e)$
SID_q
AKE.Setup: S2
$PID = \{pid_1, pid_2, \ldots, pid_n\}$
β_i, psw_i
$\mathrm{M}_{S_1}: \{ID_q, SID_q, K, PID\}$
Compute:
$\alpha_i = h(\beta_i)$
$\partial_i = h(\alpha_i || psw_i)$
$k_i^* = k_i \oplus h(\beta_i || psw_i)$
AKE.Setup: S3
SSE.Update
Compute:
For each device q:
$(\tau_q, \mathbf{e}_q) \leftarrow$ Update$(K, I_q, \mathbf{d}_q)$
Generate: N_q
Compute:
$V_1^* = $ Enc$(k_{gd}, N_q, ID_g, (\tau_q, e_q))$
Entity Authentication & Key Establishment
β_i, psw_i
$\mathrm{M}_{A_1^*}: \{V_1^*\}$
$\alpha_i = h(\beta_i)$
$\partial_i' = h(\alpha_i || psw_i)$
Check: $\partial_i \stackrel{?}{=} \partial_i'$
Generate: N_i
Compute:
$k_i = k_i^* \oplus h(\beta_i || psw_i)$
$EL = LAI_i \oplus h(k_i || N_i)$
$V_1 = h(SID_q || N_i || k_i || EL)$
AKE.Auth: A1
Generate: $N_{g'}$
Decrypt: V_1^*
Compute:
$V_2^* = $ Enc$(k_{gs}, N_{g'}, ID_{sp}, (\tau_q, e_q))$
AKE.Auth: A*1
$\mathrm{M}_{A_2^*}: \{V_2^*\}$
Decrypt: V_2^*
$\mathrm{M}_{A_1}: \{SID_q, N_i, EL, V_1\}$
$\mathrm{M}_{A_1}: \{SID_q, N_i, EL, V_1\}$
Generate: N_g
Compute:
$V_2 = h(ID_g || N_g || K_{gd} || LAI_g)$
AKE.Auth: A2
$\mathrm{M}_{A_2}: \{\mathrm{M}_{A_1}, ID_g, N_g, LAI_g, V_2\}$
Verify: SID_q, V_1, V_2
Compare: LAI_i with LAI_g
Generate: SK, SID_q^{new}
Compute:
$SID_q^{new*} = SID_q^{new} \oplus h(SID_q || k_i)$
$SK_i = h(ID_i || k_i || N_i) \oplus SK$
$SK_g = h(ID_g || K_{gd} || N_g) \oplus SK$
$V_3 = h(SK_g || K_{gd} || N_g)$
$V_4 = h(SK_i || k_i || SID_q^{new*})$
AKE.Auth: A3
AKE.Auth: A4
$\mathrm{M}_{A_3}: \{(SID_q^{new*}, SK_i, V_4) || (SK_g, V_3)\}$
Compute & Verify: V_3
Decode: $SK = h(ID_g || K_{gd} || N_g) \oplus SK_g$
$\mathrm{M}_{A_4}: \{(SID_q^{new*}, SK_i, V_4)\}$
$\mathrm{M}_{A_4}: \{(SID_q^{new*}, SK_i, V_4)\}$
AKE.Auth: A5
Compute & Verify: V_4
Decode: $SK = h(ID_i || k_i || N_i) \oplus SK_i$
$SID_q^{new} = SID_q^{new*} \oplus h(SID_q || k_i)$
Private Queries of Encrypted Data
Generate:
$k_{D_q} = H(k_i, ID_q)$
$(\rho_q, \mu_{q,j}) \leftarrow$ GenToken$(k_{D_q}, ID_q, f_{q,j})$
$V_5 = h(SK || \rho_q || \mu_{q,j} ||Enc(SK, \rho_q, \mu_{q,j}))$
SSE.GenToken
$\mathrm{M}_{Q_1}: \{$Enc$(SK, \rho_q, \mu_{q,j}), V_5\}$
$\mathrm{M}_{Q_1}: \{$Enc$(SK, \rho_q, \mu_{q,j}), V_5\}$
$\mathrm{M}_{Q_1}: \{$Enc$(SK, \rho_q, \mu_{q,j}), V_5\}$
Compute & Verify: V_5
Decrypt: Enc$(SK, \rho_q, \mu_{q,j})$
Compute:
SSE.Query
$e_{q,j} \leftarrow$ Query$(\rho_q, \mu_{q,j}, \tau_q, \mathbf{e}_q)$
$V_6 = h(SK || e_{q,j} ||$Enc$(SK, e_{q,j}))$
$\mathrm{M}_{Q_2}: \{$Enc$(SK, e_{q,j}), V_6\}$
$\mathrm{M}_{Q_2}: \{$Enc$(SK, e_{q,j}), V_6\}$
$\mathrm{M}_{Q_2}: \{$Enc$(SK, e_{q,j}), V_6\}$
Compute & Verify: V_6
Decrypt: Enc$(SK, e_{q,j})$
Decrypt: $e_{q,j}$
$d_{q,j}$

Fig. 3. PrivHome: An Authenticated, Privacy-Preserving Data Query Protocol

encrypts the data and updates the index in τ_q. The masked index τ_q and the encrypted data $\mathbf{e}_q$ are stored in the device. These information, $(\tau_q, \mathbf{e}_q)$, can be stored at a smart hub instead as we remarked in Section 4 if a device lacks storage resources, and can be backup to the service provider from time to time if required. In such a scenario encrypted queries can be performed through the smart hub.

AKE: $User_i$ and the device SD_q jointly execute AKE.Setup in order for $User_i$ to register as a user with an identity ID_i. As a result, $User_i$ stores the thumb print and password information ∂_i securely in the mobile device secure storage, as well as storing the secret key k_i protected by his/her password and thumb print k_i^*, and the pseudo-identity information of the device SID_q, and PID into his/her mobile device. The gateway stores $User_i$'s information $\{SID_q$, k_i, and $PID\}$ in its storage.

5.2 Entity Authentication and Key Establishment

In this phase, the scheme executes the authenticated key-establishment protocol in order to authenticate the user, the gateway and the device SD_q, as well as establish a session key between the user and SD_q.

AKE: $User_i$ and the gateway jointly execute AKE.Auth facilitated by SP in order for $User_i$ and SD_q to establish and agree on a session key SK. It follows the four steps as described in Section 4.2.2. $User_i$ and SD_q uses SK to encrypt the data and queries communicated between them. A symmetric encryption scheme Enc, as was defined and used in our searchable encryption protocol (Section 4.3), can be deployed for encryption using SK.

5.3 Private Queries on Encrypted Data

After the user entity has been authenticated, with the extraction and usage of k_i, and a session key SK established between $User_i$ and SD_q, the scheme executes `SSE.Query` to query for information from a smart device through the communication channel between $User_i$, SP and the gateway. The scheme may also execute `SSE.Update` to refresh the values in SD_q, whenever new data is collected from SD_q.

`SSE`: $User_i$ first prepares a query (which may contain a command on a certain functionality, for example, to increase/decrease temperature of a HVAC unit, or switch on/off of lights, as well as queries for information such as video from a surveillance camera or health status from a body sensor). $User_i$ then generates the device, SD_q's secret key using k_i, resulting in $k_{D_q} = H(k_i, ID_q)$. The query is then masked by executing `SSE.GenToken`, given the query containing the device ID, functionality and k_{D_q} as input. The resulting token is sent over to SD_q through the gateway, through the authenticated session protected by the session key SK. The device SD_q then executes `SSE.Query` using the token as input and the encrypted index, and returns the encrypted functionality value to the $User_i$, also through the authenticated session.

5.4 Security Analysis

PrivHome achieves the security goal as presented in Section 3.2. In the followings we provide discussions on how PrivHome achieve these goals.

- *Data confidentiality.* All data-in-transit for a session between $User_i$ and the smart device SD_q is encrypted using an established session key SK. This provides end-to-end encryption whereby any outsider listening to the traffics will only be able to capture the encrypted data. As long as a well-established and secure symmetric encryption scheme (i. e. AES) is used, it is infeasible for an adversary to learn any information from the encrypted data. Similarly, the data-at-rest in the smart device SD_q, or gateway, or the service provider SP, are encrypted using the symmetric encryption scheme provided through the searchable encryption scheme. This means, as in the above case, an outsider will not be able to learn the content.
- *Privacy-preserving queries of encrypted data.* When $User_i$ submits an encrypted query, it is communicated through the encrypted channel based on the session key SK. Hence an adversary will not be able to know the query. In the event where an adversary get hold of the masked index, the adversary will not be able to learn the content of the index as well, since the values are encrypted. A semi-trusted service provider, who might in its possession the masked index and the encrypted data, may try to learn the content of these two datasets. Similarly, the gateway has access to the session key (**Step A4** in Section 4.2.2) and thus it has in its possession the encrypted queries and the encrypted query results. The gateway may try to learn the underlying content of these encrypted messages. Except for the leakage defined in Section 4.3, since the index, the query and the data are encrypted under the key that is only available to the user, data confidentiality is protected and the scheme is secure up to the security of the leakage profile. We note that leakage during query has since been studied and inference attacks have been devised, in the environment of single keyword search on plaintext and databases [36], [37]. The study on how the leakage profile affects the security of a searchable encryption scheme is an on-going and active research area. We will explore, in our future work, the potential of an adversary being able to infer the plaintext from the indexes and the encrypted values in the smart devices' data structure.
- *Data authentication.* In order to prevent modification to the data by malicious adversary, the data communicated between the user $User_i$ and the device SD_q are authenticated based on a keyed hash function with the session key SK as input. This means if an adversary modifies the data, the user and the device will be able to detect such modification based on the verification of the hash value.
- *Entity authentication.* The $User_i$, through the mobile device, is authenticated through a multi-factor mechanism. This includes biometric and location. Once the session key is established, an authenticated session between the user $User_i$ and the device SD_q is maintained. Hence without the combination of the human factor and the mobile device, it is infeasible for an adversary to impersonate the identity of the user. As for SD_q, as long as the device is not physically tampered with or its software being modified during software upgrade, the secret key of the device serves as an identity authentication mechanism. The above mentioned limitations can be circumvented using, for example, giving the device access to a physically unclonable function (PuF). In our proposed lightweight authentication scheme the user's device authenticates the gateway by using the key-hash response V_4. Similarly, the gateway authenticates the user (User's device) by using the key-hash response V_1. On the other hand, both the smart home device and the gateway authenticates with each other's by using the key-hash response V_2, V_3, respectively. Now, during the execution of the data query protocol, the smart home device authenticates the user and validates his/her query by using the parameter V_5 and similarly the user authenticates the smart device by using the key-hash response V_6.

5.5 Performance Analysis and Comparison

In this section, we analyse the performance of our proposed scheme with respect to existing schemes for smart home environment. Table 1 shows the comparisons of features and performance between our scheme and the related schemes. From Table 1, we observe that the schemes presented in [6], [14], [15], [16] cannot ensure all the desired security properties. For instance, the schemes presented in [6], [14], [15] can only ensure data-privacy in transit. Besides, these schemes does not provide the ”Privacy Query” features. Even though the scheme presented in [16] can support data privacy under ”Data-at-Rest” and ”Privacy Query”, the scheme cannot ensure user authentication feature. On the contrary, the proposed scheme provides better security properties (as

TABLE 1
Features and Performance Comparisons

Schemes	Computation Costs (Auth.)	Computation Costs (Enc. Query)	User Authentication	Data Privacy (T)	Data Privacy (R)	Private Query
Kumar *et al.* [6]	4h + 2mac + 2e	–	✓	✓	×	×
Kumar *et al.* [15]	2h + 3xor + 2e	–	✓	✓	×	×
Wazid *et al.* [16]	22h + 2mac + 4e	–	✓	✓	×	×
Wen *et al.* [48]	–	((10N + 12)l + 14)exp + 10pair	×	✓	✓	✓
PrivHome	6h + 6xor	2h + (n + 2)mac + (Nl + 2)e	✓	✓	✓	✓

Notation: h: cryptographic hash operation (e.g. SHA256); mac: cryptographic message authentication code operation (e.g. HMAC-SHA256); e: symmetric encryption operation (e.g. AES); xor: exclusive-or; In the setting of Wen *et al.*'s scheme [48], N: domain of attribute values, l: number of data dimensions; n: number of smart devices; exp: modular exponentiation; pair: Bilinear pairing, where one pairing roughly equal to 8 modular exponentiations [54]. T: Data-in-Transit; R: Data-at-Rest.

TABLE 2
Experimental Specifications

Specification	User's Mobile Device (MD)	Smart Home Device (SD)	SP/Gateway (GW)
Hardware Specification	HTC One X	Temperature sensor - TMP36	Intel Core i5-2500 processor
Computation Cost of the Cryptographic Operations Used in [6], [15], [16], [48] and PrivHome			
Computational Specification	**MD**	**SD**	**SP/GW**
h (SHA-256)	0.067 ms	1.42 ms	0.037 ms
mac (HMAC-SHA-256)	1.83 ms	3.69 ms	0.078 ms
e (AES-CBC-256 Encryption)	0.072 ms	1.89 ms	0.047 ms
e (AES-CBC-256 Decryption)	0.098 ms	2.47 ms	0.062 ms
exp (Modular Exponentiation Operation)	13.56 ms	21.82 ms	8.77 ms
pair (Pairing Operation)	17.37 ms	27.54 ms	11.23 ms
Communication Cost			
Communication Specification	**MD-SP/GW**	**SP/GW-SD**	
Link Type	TD-SCDMA Network	One-hop Wireless (802.11))	
Average Transmission Time for 896-bits	23.87 ms	14.32 ms	

shown in Table 1), which are desirable for the secure and privacy-aware communication in smart-home environment.

We further compare our proposed scheme by considering the computation cost. Again from Table 1, we can see that the scheme presented in [48], which provides encrypted query, are based on public-key cryptography. Whereas the proposed scheme and the schemes presented in [6], [14], [15] are based on the symmetric key crypto-system. Accordingly, they impose lower computational overhead on the smart devices, as compared to [48]. Now, we present experimental results to analyze the performance of the proposed scheme more comprehensively. Table 2 presents the experiment specifications, including the hardware, computational, and communication specifications. For measuring the computation time of different cryptographic operations used in [6], [15], [16], [48] and/or PrivHome, we conducted simulations of their cryptographic operations on a HTC One X with ARM Cortex-A9 MPCore processor with 890 MHz CPU speed (operating as a user's device MD), a modular sensor board MSB-430 with the T1 MSP430 micro-controller and the temperature sensor - TMP36 (operating as the smart home device SD) and an Intel Core i5-2500 processor laptop with CPU speed 3.3 GHz (operating as the GW/SP). The simulation uses the JPBC library Pbc-05.14, and the JCE library to evaluate the execution time of the cryptographic operations used in the proposed scheme and [6], [14], [15], [16]. Now, based on the above simulation outcomes we can observe that in [6], the computation cost during authentication is 8.86 ms and the communication cost is $\approx$ 76.38 ms. Therefore, the overall authentication and the key establishment process in [6] takes $\approx$ 85.24 ms. Similarly, we find that the authentication and the key establishment process in [15], and [16], takes $\approx$ 38.79 ms, $\approx$ 109.9 ms, respectively. Whereas in case of PrivHome, it takes 1.75 ms of computation cost and 67.74 ms of communication cost. Hence, the entire authentication and the key establishment process in PrivHome takes $\approx$ 69.49 ms, which is slighly more than [15], but it does not support data privacy as well as private query features. Next, we consider the execution time for the encryption query. In this context, we find that the computation cost for each encryption query for the Wen et al.'s scheme [48] takes 5956.07 ms(considering $N = 5, l = 5$). Whereas for the same purpose our proposed scheme takes only 100.7 (12.91 ms at MD + 87.78 at SD) ms (considering $n = 5$, $N = 5$, $l = 5$), which is significantly lower than [48] and hence well suited even for the resource constrained devices.

In terms of the `SSE` update operation to update the values in the smart device from time to time, the scheme is efficient in that only symmetric techniques are used. That is, keyed hash computations (1.42 ms per hashing) and encryption (1.89 ms per encryption). This would be practical for hourly or for every minute update of data in a smart device, though with marginal latency. We note that updating an existing value can be made even more efficient. A smart device can pre-compute and securely store the device's encryption key and masked functionality. Then what is left to be computed for every update is only the generation of a pseudo-random number and a symmetric encryption operation (Figure 2). This means an update of a value takes approximately $1.42 + 1.89$ ms, since generating a pseudo-random number can be performed based on a hash function. Such a performance can be suitable, though with marginal latency, for devices that require very frequent update, such as a temperature sensor that may update its value every seconds.

6 Conclusion & Future Works

We proposed a privacy-preserving and authenticated scheme for securing smart home communication and data storage. The scheme, PrivHome, is built based on two protocols that we also proposed. The first is a lightweight authentication and key-establishment protocol that is constructed to provide practical entity and data authentication as well as data confidentiality for communication between the user and the smart device. The second protocol is a searchable encryption protocol constructed for privacy-preserving encrypted queries on smart devices, which as far as we know, has not been presented before. Encrypted queries and data storage is becoming a more and more important tool in personal privacy especially with the advent of smart home devices that record and store user behaviour data at home. We have shown the protocol to be secure based on our discussion, and practical through our initial experimentations on the various and only symmetric cryptographic primitives deployed. As an on-going work, we are adapting the protocol towards preserving privacy of user data under the smart hub setting.

Acknowledgments

This work was supported in part by the National Research Foundation, Prime Minister's Office, Singapore, under its Corporate Laboratory@University Scheme, National University of Singapore, and Singapore Telecommunications Ltd., in part by the National Natural Science Foundation of China under Grant 61632012, 61672239, 61822202, 61872089, 61872087.

References

[1] A. J. Brush, M. Hazas, and J. Albrecht, "Smart Homes: Undeniable Reality or Always Just around the Corner?" *IEEE Pervasive Computing*, vol. 17, no. 1, pp. 82–86, 2018.

[2] C. Wilsona, T. Hargreaves, and R. Hauxwell-Baldwin, "Benefits and risks of smart home technologies," *Energy Policy*, vol. 103, no. 2017, pp. 72–83, 2017.

[3] P. Gope,"Anonymous mutual authentication with location privacy support for secure communication in M2M home network services,"*Journal of Ambient Intelligence and Humanized Computing* ,DOI: 10.1007/s12652-017-0626-x, 2017.

[4] V. C. Gungor, D. Sahin, T. Kocak, S. Ergüt, C. Buccella, C. Cecati, and G. P. Hancke, "Smart Grid and Smart Homes: Key Players and Pilot Projects," *IEEE Industrial Electronics Magazine*, vol. 6, no. 4, pp. 18–34, 2012.

[5] H. Jo, S. Kim, and S. Joo, "Smart heating and air conditioning scheduling method incorporating customer convenience for home energy management system," *IEEE Trans. Consumer Electronics*, vol. 59, no. 2, 2013.

[6] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, and M. Sain, "Lightweight and Secure Session-Key Establishment Scheme in Smart Home Environments," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 254–264, 2016.

[7] B. L. R. Stojkoska and K. V. Trivodaliev, "A review of Internet of Things for smart home: Challenges and solutions," *Journal of Cleaner Production*, vol. 140, no. 3, pp. 1454–1464, 2016.

[8] N. Apthorpe, D. Reisman, S. Sundaresan, A. Narayanan, and N. Feamster, "Spying on the Smart Home: Privacy Attacks and Defenses on Encrypted IoT Traffic," *CoRR*, vol. abs/1708.05044, 2017. [Online]. Available: http://arxiv.org/abs/1708.05044

[9] E. K. Choe, S. Consolvo, J. Jung, B. L. Harrison, and J. A. Kientz, "Living in a glass house: a survey of private moments in the home," in *UbiComp 2011*. ACM, 2011, pp. 41–44.

[10] B. Barrett, "What Amazon Echo and Google Home Do With Your Voice Data," 2017. [Online]. Available: https://www.wired.com/story/amazon-echo-and-google-home-voice-data-delete/

[11] E. Fernandes, J. Jung, and A. Prakash, "Security analysis of emerging smart home applications," in *2016 IEEE Symposium on Security and Privacy, SP 2016*. IEEE Computer Society, 2016, pp. 636–654.

[12] Proofpoint, "Proofpoint Uncovers Internet of Things (IoT) Cyberattack," 2014. [Online]. Available: https://www.proofpoint.com/us/proofpoint-uncovers-internet-things-iot-cyberattack.

[13] N. Komninos, E. Philippou, and A. Pitsillides, "Survey in Smart Grid and Smart Home Security: Issues, Challenges and Countermeasures," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 4, pp. 1933–1954, 2014.

[14] T. Song, R. Li, B. Mei, J. Yu, X. Xing, and X. Cheng, "A Privacy Preserving Communication Protocol for IoT Applications in Smart Homes," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1844–1852, 2017.

[15] P. Kumar, A. Braeken, A. V. Gurtov, J. H. Iinatti, and P. H. Ha, "Anonymous Secure Framework in Connected Smart Home Environments," *IEEE Trans. Information Forensics and Security*, vol. 12, no. 4, pp. 968–979, 2017.

[16] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure Remote User Authenticated Key Establishment Protocol for Smart Home Environment," *IEEE Transactions on Dependable and Secure Computing*, p. 1. [Online]. Available: doi.ieeecomputersociety.org/10.1109/TDSC.2017.2764083

[17] A. Chakravorty, T. W. Wlodarczyk, and C. Rong, "Privacy preserving data analytics for smart homes," in *2013 IEEE Symposium on Security and Privacy Workshops*. IEEE Computer Society, 2013, pp. 23–27.

[18] Y. Lee, W. Hsiao, Y. Lin, and S. T. Chou, "Privacy-preserving data analytics in cloud-based smart home with community hierarchy," *IEEE Trans. Consumer Electronics*, vol. 63, no. 2, pp. 200–207, 2017.

[19] B. Choi, S. Lee, J. Na, and J. Lee, "Secure firmware validation and update for consumer devices in home networking," *IEEE Trans. Consumer Electronics*, vol. 62, no. 1, pp. 39–44, 2016.

[20] N. Apthorpe, D. Reisman, and N. Feamster, "A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic," *CoRR*, vol. abs/1705.06805, 2017. [Online]. Available: http://arxiv.org/abs/1705.06805

[21] ——, "Closing the Blinds: Four Strategies for Protecting Smart Home Privacy from Network Observers," *CoRR*, vol. abs/1705.06809, 2017. [Online]. Available: http://arxiv.org/abs/1705.06809

[22] J. Liu, C. Zhang, and Y. Fang, "EPIC: A Differential Privacy Framework to Defend Smart Homes Against Internet Traffic Analysis," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1206–1217, 2018.

[23] D. X. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," in *2000 IEEE Symposium on Security and Privacy, SP 2000*. IEEE Computer Society, 2000, p. 44.

[24] E.-J. Goh, "Secure indexes," Cryptology ePrint Archive, Report 2003/216, 2003, http://eprint.iacr.org/2003/216/.

[25] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," in *ACM CCS 2006*, A. Juels, R. N. Wright, and S. D. C. di Vimercati, Eds. ACM, 2006, pp. 79–88.

[26] M. Chase and S. Kamara, "Structured Encryption and Controlled Disclosure," in *ASIACRYPT 2010*, ser. LNCS, M. Abe, Ed., vol. 6477. Springer, 2010, pp. 577–594.

[27] D. Cash, S. Jarecki, C. S. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, "Highly-Scalable Searchable Symmetric Encryption with Support for Boolean Queries," in *Advances in Cryptology - CRYPTO 2013 and IACR eprint archive*, ser. LNCS, R. Canetti and J. A. Garay, Eds., vol. 8042. Springer, 2013, pp. 353–373.

[28] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, "Dynamic Searchable Encryption in Very Large Databases: Data Structures and Implementation," in *21st Annual Network and Distributed System Security Symposium, NDSS 2014 and IACR eprint archive (eprint.iacr.org/2014/853.pdf)*, vol. 2014. Internet Society, 2014.

[29] N. Cao, C. Wang, J. Li, K. Ren, and W. Lou, "Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data," *IEEE Transactions on Parallel and Distributed System (TPDS)*, vol. 25, no. 1, pp. 222–233, 2014.

[30] X. Ding, P. Liu, and H. Jin, "Privacy-Preserving Multi-keyword Top-k Similarity Search Over Encrypted Data," *IEEE Trans. Dependable Sec. Comput. (Preprint.)*, vol. PP, no. 99, 2018.

[31] Z. Fu, L. Xia, X. Sun, A. X. Liu, and G. Xie, "Semantic-Aware Searching Over Encrypted Data for Cloud Computing," *IEEE Trans. Information Forensics and Security*, vol. 13, no. 9, pp. 2359–2371, 2018.

[32] E. Stefanov, C. Papamanthou, and E. Shi, "Practical Dynamic Searchable Encryption with Small Leakage," in *21st Annual Network and Distributed System Security Symposium, NDSS 2014*. The Internet Society, 2014. [Online]. Available: http://www.internetsociety.org/events/ndss-symposium-2014

[33] Y. Ishai, E. Kushilevitz, S. Lu, and R. Ostrovsky, "Private Large-Scale Databases with Distributed Searchable Symmetric Encryption," in *Topics in Cryptology - CT-RSA 2016*, ser. LNCS, K. Sako, Ed., vol. 9610. Springer, 2016, pp. 90–107.

[34] M. Naveed, M. Prabhakaran, and C. A. Gunter, "Dynamic Searchable Encryption via Blind Storage," in *2014 IEEE Symposium on Security and Privacy, SP 2014*. IEEE Computer Society, 2014, pp. 639–654.

[35] M. S. Islam, M. Kuzu, and M. Kantarcioglu, "Access Pattern Disclosure on Searchable Encryption: Ramification, Attack and Mitigation," in *19th Annual Network and Distributed System Security Symposium, NDSS 2012*. The Internet Society, 2012.

[36] D. Cash, P. Grubbs, J. Perry, and T. Ristenpart, "Leakage-Abuse Attacks Against Searchable Encryption," in *ACM CCS 2015*, I. Ray, N. Li, and C. Kruegel, Eds. ACM, 2015, pp. 668–679.

[37] Y. Zhang, J. Katz, and C. Papamanthou, "All Your Queries Are Belong to Us: The Power of File-Injection Attacks on Searchable Encryption," *IACR Cryptology ePrint Archive*, vol. 2016/172, 2016.

[38] J. Ning, J. Xu, K. Liang, F. Zhang, and E. Chang, "Passive Attacks Against Searchable Encryption," *IEEE Trans. Information Forensics and Security*, vol. 14, no. 3, pp. 789–802, 2019.

[39] P. Grubbs, K. Sekniqi, V. Bindschaedler, M. Naveed, and T. Ristenpart, "Leakage-Abuse Attacks against Order-Revealing Encryption," in *2017 IEEE Symposium on Security and Privacy, SP 2017*. IEEE Computer Society, 2017, pp. 655–672.

[40] G. Kellaris, G. Kollios, K. Nissim, and A. O'Neill, "Generic Attacks on Secure Outsourced Databases," in *ACM CCS 2016*, E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, Eds. ACM, 2016, pp. 1329–1340.

[41] S. Kamara and T. Moataz, "Boolean Searchable Symmetric Encryption with Worst-Case Sub-linear Complexity," in *Advances in Cryptology - EUROCRYPT 2017*, ser. LNCS, J. Coron and J. B. Nielsen, Eds., vol. 10212. Springer, 2017, pp. 94–124.

[42] I. Demertzis, D. Papadopoulos, and C. Papamanthou, "Searchable Encryption with Optimal Locality: Achieving Sublogarithmic Read Efficiency," *IACR Cryptology ePrint Archive, and CRYPTO 18*, vol. 2017, p. 749, 2017.

[43] S. Kamara, T. Moataz, and O. Ohrimenko, "Structured Encryption and Leakage Suppression," *IACR Cryptology ePrint Archive, and CRYPTO 18*, vol. 2018, p. 551, 2018.

[44] B. Fuller, M. Varia, A. Yerukhimovich, E. Shen, A. Hamlin, V. Gadepally, R. Shay, J. D. Mitchell, and R. K. Cunningham, "SoK: Cryptographically Protected Database Search," in *2017 IEEE Symposium on Security and Privacy, SP 2017*. IEEE Computer Society, 2017, pp. 172–191.

[45] C. Bösch, P. Hartel, W. Jonker, and A. Peter, "A Survey of Provably Secure Searchable Encryption," *ACM Comput. Surv.*, vol. 47, no. 2, pp. 18:1–18:51, 2014.

[46] G. S. Poh, J.-J. Chin, W.-C. Yau, K.-K. R. Choo, and M. S. Mohamad, "Searchable Symmetric Encryption: Designs and Challenges," *ACM Comput. Surv.*, vol. 50, no. 3, pp. 40:1–40:37, May 2017.

[47] M. Wen, R. Lu, X. Liang, J. Lei, and X. S. Shen, *Querying over Encrypted Data in Smart Grids*, ser. Springer Briefs in Computer Science. Springer, 2014.

[48] M. Wen, R. Lu, K. Zhang, J. Lei, X. Liang, and X. Shen, "PaRQ: A Privacy-Preserving Range Query Scheme Over Encrypted Metering Data for Smart Grid," *IEEE Trans. Emerging Topics Comput.*, vol. 1, no. 1, pp. 178–191, 2013.

[49] Z. Savic, "LTE Design and Deployment Strategies," 2011. [Online]. Available: https://www.cisco.com/c/dam/global/en_ae/assets/expo2011/saudiarabia/pdfs/lte-design-and-deployment-strategies-zeljko-savic.pdf.

[50] P. Gope, and B. Sikdar, "An Efficient Privacy-preserving Authentication Scheme for Energy Internet-based Vehicle-to-Grid Communication," in *IEEE Transactions on Smart Grid*, DOI: 10.1109/TSG.2019.2908698, 2019.

[51] T. Xu and Y. Cai, "Location Anonymity in Continuous Location-based Services," in *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*, ser. GIS '07. ACM, 2007, pp. 39:1–39:8.

[52] P. Gope, and B. Sikdar, "Lightweight and Privacy-Friendly Spatial Data Aggregation for Secure Power Supply and Demand Management in Smart-Grids," *IEEE Transactions on Information Forensics & Security*, vol. 14(6), pp. 1554–1566, 2019.

[53] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 2007.

[54] F. Benhamouda, G. Couteau, D. Pointcheval, and H. Wee, "Implicit Zero-Knowledge Arguments and Applications to the Malicious Setting," in *Advances in Cryptology - CRYPTO 2015, Part II*, ser. LNCS, R. Gennaro and M. Robshaw, Eds., vol. 9216. Springer, 2015, pp. 107–129.

Geong Sen Poh holds a PhD degree in Information Security from Royal Holloway, University of London, UK. He is now a R & D Manager at NUS-Singtel Cyber Security R & D Lab. His main research interests include searchable encryption, cryptographic schemes for computations in the encrypted domain, protocols for distributed systems and privacy-preserving data sharing and integration. He was a committee member in the ISO standard for cryptography working group (Malaysia chapter), and committee members for various international conferences. He has published papers and filed patents in the field of information security.

Prosanta Gope (M'18) received the PhD degree in Computer Science and Information Engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, in 2015. He is currently working as a Lecturer in the department of Computer Science (Cyber Security) at the University of Sheffield. Dr. Gope served as a Research Fellow in the department of Computer Science at National University of Singapore (NUS). His research interests include lightweight authentication, authenticated encryption, access control system, security in mobile communication and cloud computing, lightweight security solutions for smart grid and hardware security of the IoT devices. He has authored over 50 peer-reviewed articles in several reputable international journals and conferences, and has four filed patents. He received the Distinguished Ph.D. Scholar Award in 2014 from the National Cheng Kung University, Tainan, Taiwan. He currently serves as an Associate Editor of the IEEE INTERNET OF THINGS JOURNAL, IEEE SENSORS JOURNAL, the SECURITY AND COMMUNICATION NETWORKS and the MOBILE INFORMATION SYSTEMS JOURNAL.

Jianting Ning Jianting Ning received the Ph.D. degree from Shanghai Jiao Tong University in 2016. He is currently a research fellow at Department of Computer Science, National University of Singapore. He will be a professor at the School of Mathematics and Computer Science, Fujian Normal University, China. His research interests include applied cryptography and cloud security, in particular, Public Key Encryption, Attribute- Based Encryption, and Secure Multiparty Computation.