Smart Home Survey on Security and Privacy

Nisha Panwar*, Shantanu Sharma*, Sharad Mehrotra*, Łukasz Krzywiecki†, Nalini Venkatasubramanian*
* University of California, Irvine, USA, † Wrocław University of Technology, Poland.

Abstract—Smart homes are a special use-case of the Internet-of-Things (IoT) paradigm. Security and privacy are two prime concern in smart home networks. A threat-prone smart home can reveal lifestyle and behavior of the occupants, which may be a significant concern. This article shows security requirements and threats to a smart home and focuses on a privacy-preserving security model. We classify smart home services based on the spatial and temporal properties of the underlying device-to-device and owner-to-cloud interaction. We present ways to adapt existing security solutions such as distance-bounding protocols, ISO-KE, SIGMA, TLS, Schnorr, Okamoto Identification Scheme (IS), Pedersen commitment scheme for achieving security and privacy in a cloud-assisted home area network.

Index Terms—Internet-of-Things, privacy, security, communication, smart home

1 INTRODUCTION

The rapid growth of IoT is assisting humans in many applications such as healthcare, transportation, entertainment, industrial appliances, sport, building management, and homes. Such IoT environments, while they provide unprecedented opportunities, raise significant security and privacy concerns. This article focuses on smart homes and buildings in which a variety of devices interact over the local network. Table I lists the typical devices in a home area network (HAN) along with the underlying communication protocols. The device connectivity in the HAN can be visualized as a star topology, where devices connect to a central device controller. The device controller is directly connected to the home router that provides connectivity with the external traffic. The design of a HAN brings in many challenges, as follows:

Device heterogeneity encompasses different embedded hardware, operating systems, and user interfaces; e.g., HAN devices might be installed with different operating systems such as Tiny OS (open source), Contiki (open source), and RIOT (micro-kernel based).

Communication heterogeneity encompasses different transmission medium protocols; e.g., different IoT specific communication protocol adaptations such as Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), User Datagram Protocol (UDP), and Transport Control Protocol (TCP).

Technical expertise. Often, smart homeowners are not engineers or security experts. Thus, most of the smart homeowners manually actuate device interaction or to own a fully autonomous home.

Resource constraints and data collection. Most of the HAN devices are lack of computational and storage power. In contrast, HAN devices are a continuous source of data streaming. Hence, one needs to store the data of all devices at one place (possibly the cloud), which imposes secure and privacy-preserving data collection and processing tasks.

Security and privacy. The HAN is susceptible to passive monitoring over wireless channels, which may reveal potential information regarding user interaction, behavior, lifestyle or physical activity. We consider a privacy threat due to inference attacks such that a homeowner might lose the control over meta-information leakage about the user activity via channel activity or device activity. The figure I shows the channel activity for three home devices: CloudCam, Google Home, and Belkin WeMo. The CloudCam shows a peak in the channel activity (up to 400 KB/s traffic rate) as the user enters the home, moves inside the home or exits the home. The Google Home shows a peak in the channel activity (up to 250 KB/s traffic rate) whenever a user initiated a voice command for the light bulbs to turn on/off. Similarly, a bi-state WeMo switch peaks during the on state and creates a channel activity lesser than 20 KB/s.

Undoubtedly, the autonomous layout among the home devices would bring-in convenience. However, if the device interaction is not secure, it would give away the home control to anyone closely listening to the communication. At worse, even an adversary located at a remote location can hijack the device session by tampering with the external traffic. For example, a smart door lock receives encrypted commands through mobile applications over the wireless channel. If an adversary replays those commands used in a session between the door lock and the mobile application, then the door lock can be compromised through an identity mis-binding attack. In this article, we restrict ourselves to a secure HAN model for proximity-based communication. Below we discuss two

![Fig. 1: Traffic activity in correlation with user activity.](image-url)
use-cases to show the need of security and privacy in HANs beyond the naive solutions mentioned above.

Use-Case 1: Privacy. HAN devices are dependent and connected such that controlling few devices would jeopardize the entire HAN. Furthermore, device traffic sniffing can lead to user activity inferences. The naive solution is to isolate the HAN from all types of external traffic entering the home gateway and internal traffic leaving the home gateway, though impractical. Another solution is to use benign duping such that the group of devices inside a home would simulate the owner activity, e.g., the owner is not in the home, yet the lights would turn on/off periodically as if the owner is at home. We provide privacy-preserving owner-to-cloud interaction such that the cloud can verify the owners’ identity but cannot infer the original identity.

Use-Case 2: Vandalism. The door locks, camera surveillance, and motion detectors make a secure physical periphery of the home. However, if these devices communicate through a public channel, then an intruder might be able to first bypass the cyber-security of these devices, and then the physical periphery of the home. Hence, it is required to secure the HAN device connectivity for the overall secure automation.

This article studies security protocols for device-to-device (D2D) and owner-to-cloud (O2C) interactions in HANs. The device interaction is classified as per the proximity communication model that utilizes spatial and temporal attributes to perceive the security requirements. We provide a brief glance through existing security protocols for each of these proximity communication scenarios.

Related standards. Smart homes are a combination of different standards: IEEE P2413 Internet of Things defines the Internet-enabled device communication with the assistance of infrastructure. IEEE 1888.4 Green Smart Homes defines the smart home paradigm from energy resource management perspective. IEEE 1547 Smart Grid Integration defines how to connect distributed renewable energy resources with the smart grid. IEEE 2030 defines how to interoperate various smart grid technologies. IEEE 1901 defines an infrastructure using broadband over power line. IEEE 2302 Intercloud Interoperability defines new means of intercloud connectivity. However, there is no standard security specification for smart home networks.

2 Security Threats and Goals

Existing HANs are vulnerable to a variety of active and passive attacks. Table 2 presents a taxonomy on these threats. Below we provide an overview of the basic security requirements necessary for the proximity-based communication scenarios in the HAN.

- **Authentication**: confirms identities of participants and provides an evidence to each participant as to whom they are communicating with. Digital signatures, secure identity transmission, physically-unclonable functions (PUF), and secure key exchanges are viable solutions for authentication.
- **Authorization**: grants privileges to an authenticated user. Fine-grained policy management and different access-control methods are used to provide access-rights.
- **Confidentiality**: hides the messages from an adversary and reveal to an authorized user only. Encryption, secret-sharing, traffic padding, zero-knowledge proofs, proof-of-knowledge, group signatures, pseudonym systems provide confidentiality.
- **Integrity**: assures the correctness and consistency of the identity and messages. Cryptographic hashing, watermarking, holographic proofs, multi-party computations, timestamping, nonce, and sequence numbers are used to maintain integrity.
- **Availability**: guarantees fair operations, resources, and services to an authorized user. Anomaly detection, firewalls, and special communication hardware preventing external malicious traffic to reach the network are techniques for achieving fairness.
- **Non-repudiation**: prevents either a sender or a receiver from denying a transmitted message or a protocol execution. Digital signatures are commonly used to provide non-repudiation.
- **Accountability or auditing**: is a process of bookkeeping each step at the sender, the receiver, or the network, so that a judge or a participant can verify the transactions in future.
- **Deniability**: The deniable communication empowers a prover to plausibly deny that a protocol instance was ever executed, for which the same prover was an active participant. Therefore, even if all protocol transcripts are stored for later analysis, it does not suffice as an identification-proof for a specific prover that participated during the protocol.
- **Unlinkability**: A user might utilize multiple pseudonyms for authentication regarding different services, however, this must not yield a linking between any two pseudonyms used by the same homeowner for two different services.
- **Non-transferability**: A homeowner has unique privileges over the HAN. For example, only one unique administrator account is the highest privileged on a standalone computer. Thus, the non-transferability over these privileges is crucial to retain the ownership. This property avoids credential forgery and the user identity attacks.
- **Forward secrecy**: In order to enable temporal security for cryptographic credentials perfect forward secrecy [1] is crucial. Forward secrecy guarantees that a session key derived from a long-term public-private key pair is secure even if one of the (long-term) private keys are compromised in the future.
- **Privacy**: The privacy is inherent in establishing and maintaining the HAN because the most economic attack is to do passive learning and gain side-channel information without even breaking the lengthy keys and complex hash codes. Essentially, D2D communication should be able to protect the privacy through hiding: user activity, behavior pattern, device activity patterns.

3 Lightweight Cryptographic Solutions

This section presents ways to incorporate existing lightweight cryptographic solutions for HANs. The HAN consists of computational and storage inefficient devices (e.g., door locks, coffee machines, thermostats) that receive on/off commands vs resource-abundant complex devices (e.g., smart meters, voice-based assistants) that additionally collect data, trigger alarms, inform the owner about device state or any resource shortage. These devices interact with each other using different communication protocols, as explained in Table 1. Thus, the HAN would require a combination of security protocols based on symmetric keys, asymmetric keys, Authenticated Key Exchange (AKE), reactive authentication, out-of-band authentication, or privacy aware identification. Our selection of these adapted protocols guarantees security properties such as secrecy, integrity, confidentiality, authentication, authorization, accountability, forward secrecy, non-transferability, unlinkability, and plausible deniability.
### Usage

| Usage                  | Example devices                                      | Communication protocols | Scenarios                                                                 |
|------------------------|------------------------------------------------------|-------------------------|--------------------------------------------------------------------------|
| Voice-based smart assistants | Alexa, Amazon Echo, Lenovo Smart Assistant, Eufy Genie (far-field Alexa control), iHome iAVS 16, HTC U11, | BLE, WiFi, ZigBee, Z-Wave | D2DWL (Alexa: what is the morning traffic status?)                        |
| Healthcare             | Zio XT Patch, HealthPatch MD, Sleep monitors         | BLE, WiFi               | D2DWL (Alexa: How much did I walk or sleep today?)                       |
| Lightening             | Lutron-Sivoia QS Shading System/Caseta Pro Dimmer, Philips-Hue Bloom/Lightstrip Plus, | BLE, WiFi, ZigBee, Z-Wave | D2DW, D2DWL (Alexa: Put the shades on enough intensity for reading/watching TV/eating.) |
| Door locks             | Schlage Connect Touchscreen Deadbolt, Ring Doorbell Pro | Cellular, WiFi, ZigBee, Z-Wave | D2DWL (Alexa: Who is at the door?)                                        |
| HVAC systems           | ecobee4 Thermostat                                  | BLE, WiFi, ZigBee, Z-Wave | D2DWL (Alexa: Is it more humid today?)                                   |
| Controllers            | Amazon Echo, Amazon Echo Dot, ARRIS SURFboard router (1750Mbps wireless speed, 1.4Gb download speed, gigabit ethernet), ASUS Blue Cave (based on Intel), Belkin WeMo Mini, Logitech Harmony Remote (use with Alexa, upto 15 devices) | WiFi, ZigBee, Z-Wave | D2DW, D2DWL (Alexa: Put lights on daylight bright, music on quite, microwave on grill mode, washing machine on dry mode.) |
| Hygiene                | SteriGrip self-cleaning door handles, Unico smartbrush | WiFi, Cellular          | D2DWL (Alexa: Restore my Unico timer and brush rotation speed.)           |
| Remote surveillance    | Logitech Circle 2, Amazon Echo Show                 | WiFi, Cellular          | D2DW, D2DWL (Alexa: Show me who came at the door between 9am to 5pm?)    |
| Smart meters           | Itron, Elster, Landis+Gyr/Toshiba, Aclara, ABB and Sensus Metering Systems | WiFi, ZigBee, Z-Wave | D2DW, D2DWL (Alexa: Tell me the peak power consumption on hours/days/weeks?) |
| Smart dust             | Vehicle keyfob, eco-cooking tools, smart gardening tools, and wild life saver | Bluetooth, WiFi | D2DWL (Alexa: Keep the smart dust in low-power mode when not in use.)    |
| Entertainment          | Sonos Play: 1, Sonos Playbar, Amazon Fire TV, Sony XBR TV, DISH Hopper 3, | WiFi, ZigBee, Z-Wave | D2DW, D2DWL (Alexa: Play my morning mantra playlist.)                    |

**Notations:** BLE: Bluetooth Low Energy, D2DWL: Device-to-Device Wireless, D2DW: Device-to-Device Wired.

| TABLE 1: Smart home devices and underlying communication protocols.

---

**Lightweight:** Our discussion below is limited to lightweight cryptographic protocols \(^2\) to support secure D2D interactions in HAN wherein devices are connected to and controlled by a central controller over a WiFi network. The protocols execute over cheap commodity hardware with limited processing power, memory, and communication bandwidth as is the case with current IoT devices. We note that the future may support a more decentralized device architecture wherein home devices may execute more autonomously (e.g., a washing machine (in order to reduce cost) may exploit the dynamic pricing model of resources such as water and electricity) and use data payment protocols using cryptocurrency that require proof-of-mining and consume larger processor, memory-usage, and communication.

### 3.1 Proximity Model

The proximity communication model in a HAN can be envisioned on two dimensions: space and time, as follows:

**Timeline.** The notion of the timeline is to incorporate communication activity of every pair/group of devices under the same roof. In addition, each device shares its local timeline with a peer device. The time representation for device activation requires a dependency relation among those devices. Moreover, it would be easier to represent how much *time* and *energy resources* a device spends in communication with other devices. Consequently, these devices and groups can be merged into a single *virtual node* as they stand adjacent on the timeline.

**Space.** The notion of space considers spatial aspects of digital devices in the HAN. The spatial arrangement of smart devices is another important dimension to create a *virtual node*. In addition to the devices that spend most of the time in communication with a specific device, there exist devices that interact less frequently. However, these devices share the spatial locality.

The device interaction can be categorized based on *space* and *time*, as follows:

- **Same Time and Same Space:** These are an indoor group of devices that share an identical timeline and the location. These devices would require wireless communication such as Bluetooth, infrared, wireless LAN, ZigBee, and Z-Wave. This scenario can be adapted with the existing out-of-band verification methods to guarantee multi-factor authentication. We briefly discuss the out-of-band verification protocols for these devices, e.g., distance-bounding.
- **Same Time and Different Space:** These are an outdoor group of devices that share the common timeline but different
TABLE 2: Existing and new security attacks in the HAN.
geographical locations. These devices would require wired or cellular (e.g., LTE) based connectivity. This scenario can be adapted with the existing secure key exchange protocols to guarantee per-session security. We briefly discuss the security protocols ISO-KE, SIGMA, and TLS for these devices.

- **Different Time and Different Space**: This use-case requires the homeowner interaction with a third-party service provider located at a distance. These interactions include service registration, activation, computation, and termination. This scenario can be adapted with privacy-preserving identification and access delegation protocols that assure security features (e.g., non-transferability, unlinkability, and deniability). These solutions provide control over the identification such that committed values can be revealed to a verifier in future and the right to reveal that secret information is uniquely held by the owner only. We briefly discuss Schnorr [3] and Okamoto IS, Pedersen bit-commitment protocol, proxy re-encryption, and homomorphic encryption approaches for these scenarios.

- **Different Time and Same Space**: This use-case requires scheduling protocols [4] such that devices receive an actuation token, perform computation, produce results, and pass the token to next device in the queue. In [5] we present a new verifiable delay scheme that allows an unpredictable amount of artificial delay before the command actuation at any home device. In particular, our solution provides improved privacy guarantees over naive solutions that require command pre-scheduling at these IoT devices.

### 3.2 Key Generation

The general methods for secure key management are key pre-sharing, key evolution, and PUF-enabled key-store. In key pre-sharing techniques, a set of potential keys are pre-shared with devices. Each device shares a subset of pre-shared keys and the common intersection of those subsets is used to generate the future session keys.

The key evolution methods generate a symmetric key or a pair of asymmetric keys for each device. The symmetric key solutions require two devices to share a bi-directional static secret key that encrypts/decrypts messages at both devices. As an advantage, the symmetric key solutions require the minimum number of exponentiation per device than asymmetric key solutions. However, this computation efficiency comes with a vulnerability to key compromise attacks. In contrast, the asymmetric-key-based protocols provide additional security features such as non-repudiation, fewer number of keys to be stored, and the secure storage on insecure media. In addition, key evolution provides methods to frequently change the secret keys while the corresponding public key remains unchanged.

In PUF secured key-store, all secret keys and pre-shared keys are stored in a single key-storage, however, this key-storage is accessible only through a master secret key. Therefore, PUF can be used to secure these secret master keys without explicitly storing the master key on a storage media. Therefore, it avoids the key extraction attacks through physically access to the device storage.

We assume a smart home scenario consists of $n$ home devices $(D_1, D_2, \ldots, D_n)$ and an owner $O$. D2D communication (wired or wireless communication channel) must be augmented with secure sessions. A secure session key agreement requires that communicating parties share a long-term static secret and use a new ephemeral secret for each session. Also, a secure session must begin with the validation of static credentials, i.e., who has generated, distributed and revoked the authentic credentials. A trusted-third-party, called Public Key Infrastructure (PKI) or Key Generation Center (KGC), e.g., Kerberos system, is used to manage these master keys.

We classify device communication into three categories: device-to-device wireless (D2DWL) connectivity, device-to-device wired (D2DW) connectivity, and owner-to-cloud (O2C) connectivity. The classification of these use-cases is derived from time-space analogy defined earlier.

### 3.3 Device-to-Device Wireless (D2DWL) Connectivity

The *same time and same space* based proximity model is applicable in D2DWL, because the device interaction through a wireless channel in a close proximity is less vulnerable to security breaches than a device interaction at a remote location. Home to parking (vehicle/keyfob), home to irrigation controllers, home to garbage bins, and home to surveillance unit are examples of D2DWL. For securing D2DWL, a reactive authentication or out-of-band verification methods can be used. The reactive authentication method combines (i) *static attributes* (e.g., a secret key, hardware-based challenge-response verification using PUF, wireless fingerprinting, time-space localization — triangulation, trilateration, multilateration, and distance-bounding [6]); and (ii) *dynamic attributes* (e.g., the user behavior or the anthropomorphic features including biometrics, gait, voice, and typing patterns) — for generating a random challenge for which only an authentic entity can generate the paired response. We briefly explain the security adaptations based on the distance-bounding protocols.

**Distance-bounding**: Location is a unique attribute for verifiable service access such as providing building access only when the user possesses right magnetic token and the user is close enough to the door. We chose distance-bounding protocol that verifies authentic credentials along with the proximity estimation over wireless channels. The distance-bounding protocols infer an upper bound on the distance between a sender and a receiver by measuring the round-trip time (RTT) of signals. In general, these protocols use $n$ rounds for accurate distance estimation by exchanging unique challenge-response pairs between a sender and a receiver. A sender forwards a unique challenge to an intended recipient within the proximity. Then, the receiver receives the challenge and computes the paired response using a trusted hardware or a shared hash function. The sender will receive this response and check the validity of response such that no older response has been replayed. Now, the sender can estimate the distance of the party sending the correct response by knowing (i) RTT: the interval when the challenge was sent to recipient and the corresponding response was received, (ii) measuring the time while radio signal traversing at the speed of light. Consequently, an estimate distance is computed such that no sender farther than the estimated distance could have transmitted the signals without incurring an additional delay over the current estimation.

### 3.4 Device-to-Device Wired (D2DW) Connectivity

The *same time and different space* based proximity model is applicable in D2DW, because the wired communication is secure for long distance interactions in real-time. For example, device connectivity with the smart meters, smart grid, broadband over power line (IEEE 1901) etc. In D2DW, the security can be...
achieved using either a symmetric key (e.g., Needham Schroeder symmetric key protocol) or asymmetric keys (e.g., AKE, non-interactive Diffie-Hellman, PKI based ISO-KE, SIGMA, TLS/SSL, NAXOS, HMQV, CMQV protocol). We present the asymmetric key based solutions, namely ISO-KE, SIGMA, and TLS protocols using Diffie-Hellman (DH) key exchange.

In Diffie-Hellman algorithm, all computations are done within a cyclic group \( G = \langle g \rangle \) of a prime order \( q \), where Computational Diffie-Hellman (CDH) assumption holds. The CDH assumption satisfies that the computation of a Discrete Logarithm (DL) function on public values \((g, g^a, g^b)\) is hard \([7]\) within group \( G \).

**Computational Diffie-Hellman.** Let \( \langle g \rangle \) be a cyclic group \( G \) generated by an element \( g \) of order \( q \). There is no efficient probabilistic algorithm \( A_{CDH} \) that given \((g, g^a, g^b)\) produces \( g^{ab} \), where \( a, b \) are chosen as random group elements.

**ISO-KE protocol adaptation:** According to ISO-KE (Figure 2a), the sender device \( D_i \) initializes a secure key exchange by sending its certificate, \( Cert_i \), and a DH exponent, \( g^x \), to a receiver, \( D_j \) (Step 1). After the certificate identity verification, \( D_j \) responds with the intended receiver certificate, \( Cert_j \), and its DH exponent, \( g^y \). However, in order to avoid any misbinding/reflection attacks, \( D_j \) concatenates the credentials of the intended recipient \( D_i \), i.e., \( \text{Sign}_j(g^x, g^y, Cert_j) \) (Step 2). \( D_i \) responds with the signature on mutually agreed values, i.e., \( \text{Sign}_i(g^x, g^y, Cert_i) \) (Step 3). The session key is derived from \( g^{xw} \). However, note that the ISO-KE protocol does not support identity hiding, since certificates are transferred in plaintexts.

**SIGMA protocol adaptation:** SIGMA supports sender and recipient identity hiding feature unless the identity of the opposite device is successfully verified. In this protocol, the DH key exchange is authenticated through digital signature, and the device identity is encrypted using a freshly driven key \((K_e)\) to protect the identity against eavesdropping.

When adapting SIGMA protocol in a HAN (Figure 2b), a device \( D_i \) sends DH exponent \( g^x \) to an intended recipient \( D_j \) (Step 1). On receiving \( g^x \), \( D_j \) computes its exponent \( g^y \) and the corresponding session key \( g^{xw} \). Then, \( g^{xw} \) is used to generate three different and computationally independent keys, i.e., a session key \((K_e)\), an encryption key \((K_e)\), and a message authentication key \((K_m)\). \( D_j \) forward an encrypted message with the exponent \( g^x \), signed exponents \((g^x, g^y)\), and the Message Authentication Code (MAC) over identity \( j \) (Step 2). The protocol terminates after \( D_i \) verifies the signed exponents and the identity from \( D_j \) and computes the corresponding session key \( g^{xw} \). In addition, \( D_i \) forwards an encrypted message with the signed exponents \((g^x, g^y)\) and the MAC over identity \( i \) (Step 3).

As an advantage, this protocol has the minimal number of message exchange than any key exchange protocol does to prevent replay attacks. Figure 2c shows a 3-round version of SIGMA protocol with the proper identity protection as a required feature. Similarly, there also exists a 4-round version of SIGMA that provides verifiers’ identity protection. Whenever the indoor home network devices do not want to reveal the identity before verifying the identity of a peer device/service in long-distance communication, then the SIGMA protocol would be preferred over ISO-KE protocol.

**Transport Layer Security (TLS) protocol adaptation:** TLS is a widely accepted standard for Internet security (Figure 2d). TLS handshakes are based on a pre-defined sequence of phases: mutual authentication, random secret exchange, session key establishment, and finish. The handshake between a device \( D_i \) and a server \( S \) starts by sending Hello messages that include supported range of cryptographic standards called cipher-suite (Step 1 and Step 2). Also, the mutual authentication is accomplished through authority signed certificates in a Certificate Exchange message (Step 2 and Step 3). \( D_i \) sends session key to server \( S \) along with its certificate in (Step 3). Also, \( D_i \) sends a finish message to acknowledge the beginning of ciphered communication (Step 4). Similarly, \( S \) sends a finish message to acknowledge the beginning of ciphered communication (Step 5). Afterward, \( D_i \) and \( S \) switch to the symmetric key encryption-based communication using the recently established session key to encrypt and decrypt the message exchange.

**Comparisons.** SIGMA protocol is preferred for the applications where privacy is crucial along with an end-to-end session security. SIGMA protocol provides identity-hiding features which are neither part of ISO-KE nor TLS. In addition, SIGMA protocols can be chosen either for sender identity-protection (3-rounds) or for receiver identity-protection (4-rounds).

### 3.5 Owner-to-Cloud (O2C) Connectivity

The different time and different space based proximity model is considered in O2C, because the data produced by devices in a home is often stored at a different space, particularly, the public cloud at a different time, due to some processing at the home and the network transmission delay.

Some home devices send data to cloud, and this data needs to be accessed in a privacy-preserving manner. Though encrypting the data at the cloud, accessing encrypted data at the cloud can reveal users’ privacy of the user. For example, if the user data is accessed by a cancer hospital, then by access-patterns and background knowledge, an adversary can reveal medical conditions of the user without knowing the encrypted data and users’ identity. Hence, the ease of using cloud services comes with threats to the data and user privacy, and it brings two main challenges:

1. **Secure storage and query processing** to prevent the cloud to learn user’s activities, through usage-patterns or query processing on the data. Hence, data storage and query processing at the cloud must be cryptographically secure using techniques, such as encryption or secret-sharing.
2. **Secure authentication** provides a way to securely authenticate the homeowner at the cloud. For this purpose, we illustrate Schnorr and Okamoto IS executing between the homeowner and the cloud. Despite having a secure connection between the cloud and the homeowner, the cloud can still reveal the proof of communication that can be avoided using secure commitment protocols. Bit-commitment protocols are based on commitment-before-knowledge paradigm and can be used for a private proof of identity possession in HAN. A bit-commitment protocol incorporates a commit phase and a reveal phase such that committed value is revealed only if the DL condition satisfies. We illustrate an adaptation of Pedersen commitment protocol in HAN.

**Schnorr’s Identification Scheme:** The scheme is based on the intractability of DL problem. According to the protocol, an owner \( O \) as identity prover selects a secret DH exponent \( x \) and releases a corresponding public value \( X \) for the verifier cloud.
C. Consequently, the verifier cloud returns a challenge $\mu$ for the identity proving owner. Now the prover generates a combined response $\rho$ such that it is computationally hard to compute $\rho$ without possessing the knowledge of $x$. Furthermore, there exist multiple variations to regular Schnorr like IS. For example, one possible way [8] is to replace $\rho$ with $\hat{\rho}$, i.e., $\hat{\rho}^x + a \mu$ where $\hat{g} = H(X | \mu)$ is the new generator such that the IS remains robust against the ephemeral key leakage.

Common inputs are $(p, q, g, G)$ where $p, q$ are large prime numbers, $g$ is the generator of order $q$ in group $G$.

Keygen$(pk, sk)$: $sk = a \in Z_q$ and $pk = g^x = A$. The protocol $\Gamma(O, C)$ steps for the prover $O(a, A)$ and the verifier $C(A)$ are given below:

- Prover selects $x \in \mathbb{Z}_q$ at random in Schnorr’s group and computes $X = g^x \in G$.
- Prover sends $X$ to the verifier.
- Verifier computes random $\mu \in \mathbb{Z}_q$ and sends $\mu$ to prover.
- Prover computes response $\rho = x + a \mu$ and sends $\rho$ to verifier.
- Verifier accepts the proof of commitment if $g^\rho = g^{x + a \mu} = (g^x)^\mu = X^\mu$.

**Okamoto Identification Scheme:** Okamoto IS [9] is based on DL problem and provides a proof of long-term secret key possession at the prover. The owner $O$ possess secret keys $(a_1, a_2)$ and prove it to the verifier cloud using an ephemeral secret $(x_1, x_2)$. Consequently, the verifier cloud returns a challenge $\mu$ for the identity proving owner. Now the prover generates a combined response $(\rho_1, \rho_2)$ such that it is computationally hard to compute $(\rho_1, \rho_2)$ without possessing the knowledge of $(x_1, x_2)$.

Common inputs are $(p, q, g, G)$ where $p, q$ are large prime numbers, $g$ is the generator of order $q$ in group $G$. Also, $g_1, g_2 \in G$ where $\log_{g_1} g_2$ is unknown to the prover. The protocol $\Gamma(O, C)$ steps are below. The prover $O(m)$ commits a message $m$ and verifier $C(c)$ verifies the commitment $c$.

- **Commit:** Prover randomly selects $r \in \mathbb{Z}_q$ and send $c = g_1^r \cdot g_2^m$ to the verifier.
- **Reveal:** Verifier receives the $(m, r)$ and output $m$ if $g_1^r \cdot g_2^m = c$.

**Comparisons.** Schnorr and Okamoto schemes are used for proving the identity, i.e., the knowledge of the secret key/keys, but Okamoto scheme is secure in stronger model (active adversary) at the cost of higher computational complexity (more exponentiations). Schnorr scheme is secure with passive adversary observing the protocol messages, i.e., the honest-verifier model. Both schemes can also be adapted to provide provably secure digital signatures through Fiat-Shamir transformation [11]. Unfortunately, both Schnorr and Okamoto ISs do not withstand the ephemeral key leakage attacks such that security of long-term keys depends on the security of ephemeral keys. For the immunity against such attacks there exists modified Schnorr and modified Okamoto that require additional computational complexity. Interestingly, Pedersen’s commitment scheme is computationally binding and unconditionally hiding. Therefore, Pedersen’s scheme is useful as a sub-procedure for the applications that require security against an unbounded adversary.

**Computations at the cloud:** For sharing the data with a third user, proxy re-encryption-based access delegation is applicable such that a proxy key is generated by the homeowner and proxy storage node would re-encrypt the data with this key that could only be decrypted using the secret key of the third user. The re-encryption key $rk_{(a,b)}$ is generated using a method...
TABLE 3: Protocol comparison.

| Protocols               | Direct iteration cost | Certificate preprocessing | Exponentiation cost (per party) | Out-of-band verification | Purpose                        |
|-------------------------|-----------------------|---------------------------|---------------------------------|--------------------------|--------------------------------|
| ISO-KE [12]             | 3 rounds              | Yes                       | 1                               | No                       | Secure key exchange (D2DW)     |
| SIGMA [13]              | 3 rounds              | No                        | 1                               | No                       | Secure key exchange (D2DW)     |
| TLS [14]                | >3 rounds             | Yes                       | 1                               | No                       | Secure key exchange (D2DW, D2DWL) |
| Schnorr IS [3]          | 4 rounds              | No                        | 1                               | No                       | Identification (O2C)           |
| Okamoto IS [10]         | 3 rounds              | No                        | 2                               | No                       | Identification (O2C)           |
| Pedersen commitment [10]| 3 rounds              | No                        | 2                               | No                       | Zero-knowledge proof of knowledge (O2C) |
| Distance-bounding [6]   | >4 rounds             | No                        | 1                               | Yes                      | Authentication (D2DWL)         |

Keygen($sk_a$, $pk_a$, $sk_b$, $pk_b$), where $a$ is the delegator (e.g., homeowner) and $b$ is the delegatee (e.g., cloud, law agency or auditor) $sk_b$ is the secret key, and $pk_b$ is the public key. This access delegation could be either unidirectional or bidirectional meaning that the data access rights can either be for a specific delegatee or mutually shared between both the delegator and the delegatee.

Proxy re-encryption converts the ciphered data into non-transferable ciphertext such that a colluding proxy and the delegatee cannot transfer access rights to a third party. Therefore, either the delegator/homeowner can access the data or the delegatee/cloud can access the data, even if the proxy colludes with any third party. For example, a proxy re-encryption scheme based on Elgamal assumes a group $G = (g)$ of a prime order $q$. Also, the long-term public key is computed as $g^x$ where $x \in \mathbb{Z}_q$ is the secret key and $g$ is the group generator. A message $m$ is encrypted as $c_a = (g^x)^r, m g^r$ for a random $r \in \mathbb{Z}_q$ by access delegator $a$. Re-encryption keys are generated as $rk_{(a,b)} = y/x$ mod $q \in \mathbb{Z}_q$, where $x$ and $y$ are the long-term secret key of party $a$ and $b$. Now the re-encryption process at the intermediate proxy node would compute exponentiation by using re-encryption key $y/x$ mod $q$ on the first part of ciphertext ($g^x)^r$ as $((g^x)^y/x, m g^y) = (g^{y/x}, m g^y)$. The access delegatee $b$ then generate original message $m = c_{a,2ca_{a,1}^{-1}(y/x)}$.

For computing on encrypted data, many techniques are proposed in the literate and can be used. For example, homomorphic encryption allows computations over encrypted data without decrypting the data. Homomorphic encryption provides data confidentiality during data aggregation process; however, it is significantly slow. Here, we briefly explain Pailler’s homomorphic encryption scheme that provides additive property, such that product of two encrypted text would be same as addition of the same text in plaintext. According to Pailler’s homomorphic encryption scheme:

$$c = g^m r^n \mod n^2$$

$$m = (c^\lambda \mod n^2 - 1)/n \mu \mod n$$

where $c$ is ciphertext, $m$ is a message in plaintext, $r$ is a random number $r \in \mathbb{Z}_n$, $(n, g)$ is a public key and $(\lambda, \mu)$ is a private key. Overall, neither proxy re-encryption nor homomorphic encryption scheme requires the secret key for data computation or data access. Therefore, both of these approaches are useful in providing the secure computation on encrypted data.

4 PERFORMANCE

Table 3 compares protocols on four different criteria. The second column represents the direct iteration cost that determines the communication complexity. Furthermore, online interaction with the trusted authority for each interaction is time-consuming and vulnerable to congestion. Therefore, these adapted schemes omit the need of an online assistance from a trusted-third-party for every round of authentication. The third column shows whether the certificate pre-processing is required before the protocol execution or not? The fourth column shows the computational complexity, i.e., the required number of exponentiations at each party during the protocol execution. The fifth column shows that only distance-bounding approaches require an out-of-band communication to cross-verify the identity over a wireless channel by measuring its tentative distance. Finally, the last column represents the interaction model (D2D or O2C) where these selected protocols are applicable.

Our implementations have been created using Python 3 with Charm Crypto library [15]. We tested several proof of concept implementations: (i) on the NIST-approved elliptic curve prime192v1 [16]; (ii) in Schnorr groups, with 1024-bit safe primes (e.g., $p = 2q + 1$). Average computational time for each protocol has been measured by running 1000 executions of each protocol. The results are presented in Table 3. The message exchange does not impose a latency overhead and the message latency is unrelated to the computations. Thus, each protocol implementation is created as a single program and the computations of parties are interweaving.

| Protocols               | EC prime192v1 | Integer Group |
|-------------------------|---------------|---------------|
| ISO KE                  | 3.899 ms      | 7.352 ms      |
| SIGMA                   | 5.134 ms      | 10.799 ms     |
| Schnorr IS              | 1.126 ms      | 2.439 ms      |
| Okamoto IS              | 1.969 ms      | 4.154 ms      |
| Pedersen commitment     | 1.898 ms      | 2.552 ms      |

TABLE 4: Execution times for different protocol versions.

5 CONCLUSION

The article studies security protocols for a proximity-based communication model in HANs. Broadly, the HAN model consists of device-to-device and owner-to-cloud interactions. The proximity-based communication model incorporates space and
time as two dimensions. The suitability of these security protocols is guided by these two dimensions. We provide a comparison and evaluation for these security protocols. Subsequently, we find that the computation latency overhead for these protocols is comparable to wireless signal speed as a fraction of second, therefore, is readily applicable in HAN settings.

References

[1] H. Krawczyk, “Perfect forward secrecy,” in Encyclopedia of Cryptography and Security, pp. 457–458, 2005.
[2] 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.
[3] C.-P. Schnorr, “Efficient identification and signatures for smart cards,” in Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology, pp. 239–252, 1990.
[4] S. M. D’Souza and R. Rajkumar, “Time-based coordination in geo-distributed cyber-physical systems,” in 9th USENIX Workshop on Hot Topics in Cloud Computing, HotCloud 2017, 2017.
[5] N. Panwar, S. Sharma, G. Wang, S. Mehrotra, and N. Venkatasubramanian, “Verifiable round-robin scheme for smart homes,” in Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy, CODASPY ’19, pp. 49–60, 2019.
[6] S. Brands and D. Chaum, “Distance-bounding protocols,” in Advances in Cryptology EUROCRYPT: Workshop on the Theory and Application of Cryptographic Techniques, pp. 344–359, 1994.
[7] W. Diffie and M. Hellman, “New directions in cryptography,” vol. 22, pp. 644–654, 2006.
[8] Ł. Krzywiecki, “Schnorr-like identification scheme resistant to malicious subliminal setting of ephemeral secret,” in Innovative Security Solutions for Information Technology and Communications: 9th International Conference, SECITC, pp. 137–148, 2016.
[9] T. Okamoto, “Provably secure and practical identification schemes and corresponding signature schemes,” in Advances in Cryptology, pp. 31–53, 1993.
[10] D. Chaum and T. P. Pedersen, “Wallet databases with observers,” in Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, pp. 89–105, 1993.
[11] M. Abdalla, J. H. An, M. Bellare, and C. Namprempre, “From identification to signatures via the fiat-shamir transform: Minimizing assumptions for security and forward-security,” in Advances in Cryptology — EUROCRYPT, pp. 418–433, 2002.
[12] “ISO/IEC 9798-3, entity authentication mechanisms, part 3: Entity authentication using asymmetric techniques,” 1993.
[13] H. Krawczyk, “SIGMA: the ‘sign-and-mac’ approach to authenticated diffie-hellman and its use in the ike-protocols,” in Advances in Cryptology, 23rd Annual International Cryptology Conference, pp. 400–425, 2003.
[14] T. Dierks and E. Rescorla in The Transport Layer Security (TLS) Protocol, Version 1.2, 2008.
[15] J. A. Akinleye, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, “Charm: a framework for rapidly prototyping cryptosystems,” J Cryptographic Engineering, vol. 3, no. 2, pp. 111–128, 2013.
[16] “NIST, C: digital signature standard available at url: [http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf]”