IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 19, 2021, accepted September 17, 2021, date of publication September 20, 2021,
date of current version September 29, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3114152

A Survey on Smart Home Authentication: Toward Secure, Multi-Level and Interaction-Based Identification

SALEM ALJANAH, NING ZHANG, AND SIOK WAH TAY
Department of Computer Science, The University of Manchester, Manchester M13 9PL, U.K.

Corresponding author: Salem AlJanah (salem.aljanah@manchester.ac.uk)

This work was supported by the University of Manchester.

ABSTRACT With the increased number and reduced cost of smart devices, Internet of Things (IoT) applications such as smart home (SHome) are increasingly popular. Owing to the characteristics of IoT environments such as resource constrained devices, existing authentication solutions may not be suitable to secure these environments. As a result, a number of authentication solutions specifically designed for IoT environments have been proposed. This paper provides a critical analysis of existing authentication solutions. The major contributions of the paper are as follows. First, it presents a generic model derived from an SHome use-case scenario. Secondly, based on the model, it performs a threat analysis to identify possible means of attacks. The analysis leads to the specification of a set of desirable security requirements for the design of authentication solutions for SHome. Thirdly, based on the requirements, existing authentication solutions are analysed and some ideas for achieving effective and efficient authentication in IoT environments are proposed.

INDEX TERMS Authentication, Internet of Things (IoT), security, smart home (SHome).

I. INTRODUCTION

Smart Home (SHome) is one of the Internet of Things (IoT) applications [1]. SHome applications have recently become more popular due to the increase in the number of affordable smart devices, e.g., Amazon Echo [2]. The aim of the SHome is to enhance the quality of life for its residents through automating household tasks, such as energy management, security surveillance, and healthcare services [3], [4].

Despite the many benefits provided by the SHome, it is typically accompanied by a range of security issues [5]. One of the issues is how to authenticate a large number of heterogeneous and possibly resource constrained devices in a secure and efficient manner [6].

Authentication is about verifying the identity of an entity which can be a human, software process, or hardware device. Authentication is the first line of defence in an SHome environment as it is a prerequisite for other security services such as access control, logs and auditing and intrusion detection, etc. Without reliable authentication provisioning, the whole system will be put at risk [7]. Owing to the characteristics of IoT, such as the diversity of devices, the existence of resources (data, services and other resources) with different sensitivity levels, and automatic machine to machine (M2M) communications or interactions, existing authentication solutions are not readily applicable [8]. As a result, new research efforts [9]–[32] are being made to develop secure authentication solutions which are applicable to the IoT environment.

While some of these efforts have advanced in satisfying most of the desirable security requirements, there is limited amount of work on designing an authentication service that can adapt the level of assurance offered by the authentication service in adaptation to the sensitivity level of the action (an access or an interaction) for which the authentication is performed. Different resources and interactions may have different sensitivity levels. For example, payment data is more sensitive than temperature data; opening a safe is a more sensitive task than turning off the lights, etc.

Access to a more sensitive resource or performing a more sensitive task should be supported by an authentication process producing a higher assurance level. However, a higher assurance level is usually accompanied with a higher (level of) overhead cost. To support effective and efficient authentication, there is a need for an authentication solution that supports different Levels of Assurance (LoA). In this way,

The associate editor coordinating the review of this manuscript and approving it for publication was Shaohua Wan.

we can strengthen security while, at the same time, keeping overhead costs as low as possible.

Different from existing surveys of IoT authentication solutions [7], [33]–[36], this paper conducts the survey from the perspective of the characteristics of an IoT environment, its heterogeneous nature, types of interactions during communication and the fact that the resources hosted, and interactions taking place, in such an environment typically have varying levels of sensitivity, thus arguing that, a suitable authentication solution in such an environment should be one that is both effective and efficient, and the design of such a solution should take into account those characteristics.

To scope the work without affecting generality, we have chosen the SHome as an underlying IoT environment. An SHome environment hosts diversified IoT devices and applications [37], so any findings of the survey should be applicable to other IoT environments. Before discussing related work, the major contributions of this paper are as follows.

- A generic smart home model, based on a use-case scenario, has been proposed.
- A comprehensive threat analysis in an SHome environment has been carried out.
- A set of requirements for achieving effective and efficient authentication has been specified.
- A comprehensive analysis of existing authentication solutions has been carried out to identify research gaps.
- Ideas for achieving a more effective and efficient protection in IoT environments have been proposed.

In detail, the rest of the paper is structured as follows. Section II describes an SHome use-case scenario. Section III presents the generic SHome model. Section IV performs the threat analysis based on the model. Section V specifies the set of security requirements. Section VI analyses existing authentication solutions against the specified security requirements. Section VII presents further discussions, and Section VIII concludes the paper.

II. USE-CASE SCENARIO

It is anticipated that an SHome will typically house a fair number of IoT or smart devices [38]. These devices are smart, capable of performing some computing (such as sensing environment, processing data or controlling other devices) and communication tasks autonomously. As depicted in Figure 1, the devices are of heterogeneous nature and some may be resource constrained. They connect and interwork (with each other) to form a platform that provides smart services [39]. This section presents a generic description of the SHome environment to cover different SHome scenarios discussed in [40]–[46].

A. SHome DEVICES

An SHome device can be any smart household device, appliance, or anything that is equipped with a sensor and/or an actuator. The sensors on a device are used to monitor the environment, while the actuators are used to control that environment or another device. Let us assume that the SHome is Alice's house. The house hosts a range of smart devices, such as a smart doorbell and camera at the front door, a refrigerator, toaster and oven in the kitchen, a television (TV) in the living room, three computers, one in each of the bedrooms and an air conditioner located in the hall of the house. The house also hosts several sensing devices for sensing gas, smoke, light, temperature, pressure, and indoor position and actuators for switching on and off gas, closing and opening curtains and windows, etc. Like all the smart devices, these sensing devices, in addition to performing the designated sensing function, are also capable of communicating with other devices.

B. SHome ACCESS

Accesses to the smart devices and services in the SHome are typically controlled by the owner (i.e., Alice). Alice has the full control over the SHome. She is in charge of the initial system setup, and access right management in the SHome. All residents living in the SHome can access all the devices when they are at home or away from home, whereas a guest or a non-resident user can only access the devices or the smart services in the SHome if and only if they are inside the house. Any of these users may use or access single or multiple devices at a time. The accesses to the devices or services are typically done using a smartphone application. For example, Alice's son can use the application installed on his smart phone to turn on the TV, the TV can send an alert message to Alice once her favourite show is about to start, the light sensor can send a request to the light control to turn off the lights once it is day time or when Alice's family has all gone to work and school, and at night when it is time for everybody to go to sleep, a light sensor can send simultaneous requests to all the window shutters to close themselves, and so on.

III. A GENERIC MODEL: ENTITIES, ACCESS DOMAINS, AND INTERACTIONS

Based on the use-case scenario, we can construct a generic use-case model (hereafter referred to as the G-SH model) for the SHome environment. In addition to the entities and access domains, the model also captures the mode and the location of each interaction. The consideration of the interactions is important in the design of an IoT authentication solution, as they affect the security level needed to protect the interaction. For instance, an external access request (coming from outside the SHome) may pose a higher security threat compared with internal access requests. Thus, a higher security level may be needed to secure that interaction. The G-SH model is shown in Figure 2.

A. ENTITIES AND ROLES

SHome has two types of entities: human users (hereafter referred to as users) and smart devices (hereafter referred to as devices). A service request may be made by a user or

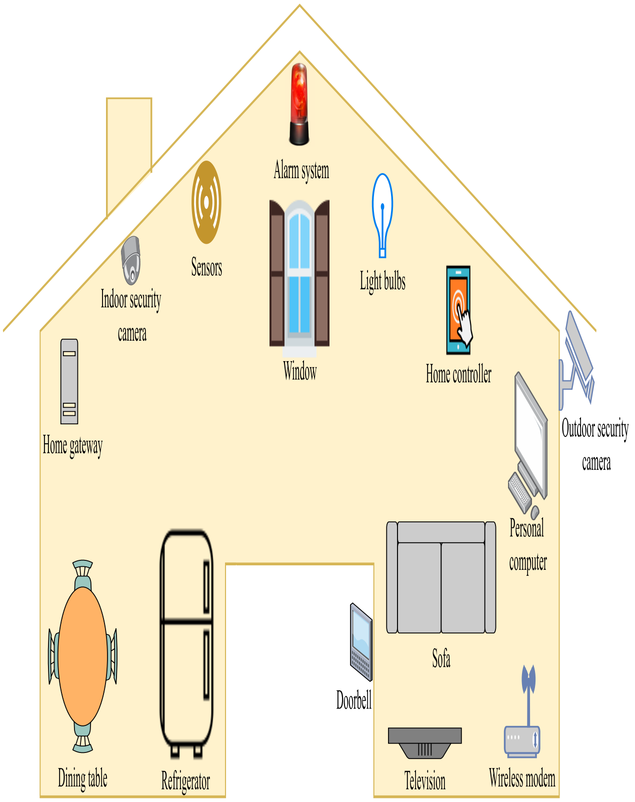

FIGURE 1. SHome environment.

by a device (in this case they are both requestors), and a service response may also be initiated by a user or by a device (responders).

Based on their capabilities, the devices can be classified into two groups: (i) constrained devices, and (ii) non-constrained devices. The Internet Engineering Task Force (IETF) classifies the first group (i.e., constrained devices) into further three groups: class-0 (C0), class-1 (C1), and class-2 (C2). The details for the classification are shown in Table 1 [47]. Yoon *et al.* [48] added an additional class, the fourth class (i.e., class-2+ (C2+)), to capture the devices that have more capabilities such as a personal computer. Although the C2+ devices are more capable compared with other devices, they may still have some constraints (e.g., limited battery life in a smart phone) [49]. The non-constrained devices are regarded as the C2+ devices in our work. They have more capabilities compared with the constrained devices (i.e., C0, C1, and C2 devices). The first class (i.e., C0) devices have the lowest capabilities. Hence, these devices depend on the help of capable devices (e.g., a smart home gateway) to facilitate communication. C1 devices can communicate directly with other devices (i.e., without going through the gateway) and this is done with the help of some lightweight protocols (e.g., the Constrained Application Protocol (CoAP)). C2 devices are more capable and can communicate using conventional protocols, but they may benefit from using lightweight protocols. C2+ devices typically use existing protocols designed for conventional devices unchanged. Some examples of the different classes of devices are given in paper [48].

TABLE 1. SHome devices (KiB = 1024 bytes).

Devices		Data size (KiB)	Code size (KiB)	Example
Constrained	C_0	<10	<100	Low-power sensor
	C_1	∼10	∼100	8-bit thermostat
	C_2	∼50	∼250	IP Camera
Non-constrained	C_{2+}	>50	>250	Smart phone

To govern controlled accesses to resources, which are typically accessed via the use of devices in an IoT environment, different devices may be assigned with different roles, each with a distinctive level of privilege. For simplicity without losing generality, in this work, we define three roles, i.e., an admin, a host, and a guest. The admin role has the highest privilege level and with this role, a user can initiate and control the SHome setup and management. The host role has the next level of privilege, and with this role, a host can use and manage the SHome. The third role, guest, has the lowest privilege level, and it is typically assigned to non-resident users or residents whose access we wish to limit. This role allows a user to only use the devices from inside the SHome.

B. ACCESS DOMAINS

Just like the case in our current homes, an SHome is expected to be equipped with a physical access control system, such as a smart lock. An entity is not expected to get inside an SHome without being permitted by the owner of the SHome. Being given the permission indicates a level of trust the owner has on the entity concerned. To capture this level of trust, we define two access domains, an SHome domain, and an out home (OHome) domain. The SHome domain covers the entire SHome, including all the resources in the SHome (devices, services and any data in the SHome). The OHome domain, on the other hand, covers all the entities that are located outside the SHome. The two domains are separated by a smart home gateway (SHG). The SHG deals with resource coordination, interconnection, and interoperability within the SHome domain or cross domain.

An access of SHome resources (services, devices, and data) may be performed inside the SHome, in which case, it is called intra-domain access, or from outside the SHome, in which case, it is called inter-domain access.

C. INTERACTIONS

The interactions among the entities in an SHome may be for a number of purposes. These purposes can be summarized as follows:

1) Connection establishment: A requestor logs in to the SHome network.

2) Negotiation: The requestor negotiates with the SHome authentication server (AS).
    a) Access request: The requestor sends a request to the AS to access an SHome resource. The request may include information such as the identities of the entities involved (e.g., the requestor, AS, and target resource).
    b) Authorization status: AS verifies that the requestor is authorized to access the resource.
    c) Level of assurance status: AS obtains the level of assurance required (RLoA) to access the resource. Then, it compares it with the requestor level of assurance derived (DLoA) during connection.
    d) Decision: AS analyses the obtained data. If additional verification is needed, it sends a list of suitable authentication methods to the requestor.
    e) Confirmation: The requestor may choose to accept or reject these methods.
3) Authentication: The requestor verifies its identity to the AS to obtain access credentials.
4) Access: The requestor uses his access credentials to access the target resource.

The interactions among the SHome entities can be classified into two generic categories: intra-domain interactions and inter-domain interactions. Intra-domain interactions are initiated by requestors that are located inside the SHome, i.e., both the requestors and responders are within the SHome domain. Inter-domain interactions are initiated by requestors that are outside the SHome, i.e., the requestors are in the OHome domain whereas the responders are in the SHome domain. Intra-domain interactions are typically facilitated by one or more internal networks, whereas the inter-domain interactions involve the use of external networks, such as the Internet, in addition to the internal networks. For clarity, hereafter we refer to the internal networks as the SHome network.

Depending on the types of entities involved, Schwartz [50] has classified IoT interactions into three modes of interactions: (i) human-to-machine, (ii) machine-to-human, and (iii) machine-to-machine interactions. However, this classification method does not consider the mode of communications, namely unicast or multicast, during an interaction. Hence, we propose to classify the interactions based on both the types of entities involved and the mode of communications used, and this leads to the following five types (or modes):

1) User-to-device interaction: if a user initiates communication with a device.
2) Device-to-user interaction: if a device initiates communication with a user.
3) Device-to-device interaction: if a device initiates communication with another device.
4) Device-to-multiDevice interactions: if one device initiates communications with multiple devices.
5) MultiDevice-to-device interactions: if multiple devices initiate communications with a device.

In real-life, a single access request may trigger multiple interactions of multiple types (i.e., user-to-device, device-to-user, device-to-device, device-to-multiDevice, and/or multiDevice-to-device). The design of an authentication solution should take into account all these types of interactions.

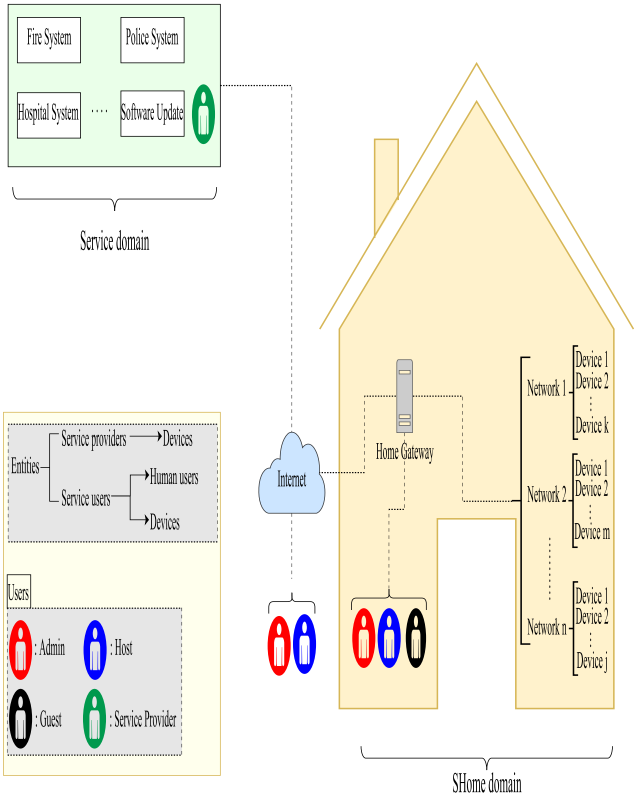

FIGURE 2. The G-SH model.

IV. THREAT ANALYSIS

Based on the G-SH model, this section identifies the threats which may be countered by an authentication service.

A. IMPERSONATION

Impersonation is the identity theft of another entity. In the SHome environment, this attack may be launched by legitimate or illegitimate entities. For example, a deceitful guest may masquerade itself as the SHome host to gain a higher privilege level. A service provider (i.e., an entity located in the service domain) may masquerade itself as the SHome admin to gather information about SHome entities. This information may later be used to launch targeted advertising. An adversary may try to impersonate the AS to deceive legitimate requestors into revealing their confidential information, e.g., login credentials. The aim of an impersonation attack could be to gain unauthorized access to SHome resources or to mount attacks on SHome devices.

B. EAVESDROPPING

Eavesdropping is monitoring communication channels to intercept messages or data or gather any information useful to the adversaries. Messages communicated in an SHome network may contain valuable items (e.g., credentials). If an adversary was able to intercept these items, s/he might be able to use them to impersonate legitimate entities. The adversary may also use the information gathered to track users' movements to commit other crimes (e.g., burglaries). A service provider may use the information to track entities without their consent to target them commercially. The typical aim of eavesdropping attacks is to spy on the SHome.

C. REPLAY

Replay is the re-transmission of a previously captured message. For instance, an adversary may intercept a message coming from a legitimate requestor, then, resend it to the AS to impersonate the requestor. An adversary may also mount DoS (Denial of Service) attacks on SHome entities or service providers by replaying intercepted messages, making them too busy to serve legitimate requestors.

D. DENIAL OF SERVICE

The objective of the SHome (automating household services) cannot be achieved if the services are inaccessible. DoS attacks may also be mounted by flooding the SHome network with a large number of requests draining up the bandwidth resource. Similarly, a service provider may use the same approach with a competitor (i.e., another service provider) to eliminate competition and control the market.

V. SECURITY REQUIREMENTS

Based on the threat analysis, we specify a set of desirable security requirements for an effective authentication service in an SHome. The requirements are as follows.

A. ENTITY AUTHENTICATION

Entity authentication provides identity verification. To counter impersonation attacks, SHome resources should be able to verify that the requestors are who they claim to be, and vice versa. Therefore, a secure entity authentication service should support mutual authentication between:

- Requestor and AS: to ensure that access credentials are only granted to requestors authorized by the AS.
- Requestor and target resource: to ensure that the requestor is authorized by the AS to access the resource.

To protect the different modes of interactions, we specify further four authentication requirements and these are:

1) User-to-device authentication: to verify the identity of a user to a device.
2) Device-to-device authentication: to verify the identity of a device to another device.
3) Device-to-multiDevice authentication: to verify the identity of a device to multiple devices.
4) MultiDevice-to-device authentication: to verify the identity of multiple devices to a device.

B. MESSAGE FRESHNESS

Message freshness assures that a message has been generated fresh (not used before). Since a considerable amount of data communicated in an SHome network comes from sensors (i.e., real-time data), it is important to ensure message freshness. Message freshness is achieved by using freshness identifiers, such as timestamps and nonces [48].

C. MESSAGE INTEGRITY

Message integrity assures that a message has not been altered by unauthorized parties during transmission. This is typically achieved by using security controls, such as hash function, hashed message authentication code (HMAC) and digital signature, to protect message integrity [51].

D. CONFIDENTIALITY

Confidentiality protects secret message components, e.g., access credentials, from being disclosed to unauthorized parties. Confidentiality is typically achieved by encrypting secret components using symmetric and/or asymmetric cryptosystems [52].

E. AUTHORIZATION

Authorization controls access rights. The AS and target resources should verify each requestor's access rights (i.e., permissions) in each access request before granting the access. A number of attributes (e.g., the requestor role, DLoA, and responder RLoA) may affect authorization decisions. In many cases, an attribute-based access control (ABAC) mechanism is more suitable for IoT environments in comparison with other access control mechanisms, e.g., role-based access control (RBAC). This is because it provides dynamic and context-aware authorization, where access is granted or denied based on a number of attributes [53], e.g., RLoA.

Authorization should be considered during the authentication process to reduce unnecessary overhead and strengthen the system against security threats, e.g., impersonation and unauthorized access attacks.

F. AVAILABILITY

Availability refers to the accessibility of authentication solutions used to protect SHome resources. These solutions should always be available to legitimate requestors. This is typically achieved by using a number of security controls, e.g., encryption, to protect the solutions against security threats, e.g., DoS attacks, and ensure availability.

VI. AUTHENTICATION IN IoT ENVIRONMENTS: A COMPREHENSIVE LITERATURE SURVEY

A. AUTHENTICATION METHODS AND APPROACHES

There are a large number of authentication methods proposed for IoT applications [9]–[32]. Based on the type of token used during an authentication instance, these methods can be classified into three main categories, i.e., (i) something you know, (ii) something you have, and (iii) something you are [54].

1) SOMETHING YOU KNOW

These are knowledge-based authentication methods. They use a secret known to a requestor to verify the requestor's identity. The secret could be a password, a symmetric key, a PIN (Personal Identification Number), an answer to a security question, etc. These methods are widely used as they are relatively easy and cheaper to implement, but, in comparison with other methods, they are more vulnerable to brute force and shoulder surfing attacks [55]. To thwart brute force attacks, the length and randomness of the secret are often increased. However, the longer and/or the more random a secret is, the harder it is to memorize it. This conflict between security and easiness to memorize a secret limits the use of such methods to protect resources with lower sensitivity levels.

2) SOMETHING YOU HAVE

To address the conflict between security and easiness to memorize a secret, possession-based authentication methods are proposed. With this category of methods, a hardware token, such as a smart card or a radio frequency identification (RFID) tag, is used to store a secret. The security of these methods lies on the owner of a token keeping the token secure. If an adversary is able to obtain such a token, then the adversary would be able to impersonate the legitimate owner of the token. To reduce risks as caused by the theft of such a token, thus strengthening the security of these methods, the access to a hardware token is typically controlled by using another authentication method. For instance, smart cards are commonly used together with a PIN or a passcode [56].

3) SOMETHING YOU ARE

These authentication methods are biometric-based. The methods utilize the unique physiological or behavioral characteristics, e.g., fingerprints or touch dynamics, of a requestor to verify the requestor's identity. Although biometric-based tokens are typically harder to falsify, they suffer from reliability problems. These methods may lead to false positives, identifying an adversary as a legitimate requestor, or false negatives, identifying a legitimate requestor as an adversary [54]. In addition, additional hardware is required to implement this category of authentication methods.

B. ONE-FACTOR AUTHENTICATION VS MULTI-FACTOR AUTHENTICATION

The security level needed to protect a system can be determined by a number of attributes, e.g., requestor location, asset value, and underlying communication channel. Depending on the security level, an authentication solution may use one method, in which case, the process is called one-factor authentication, or a number of methods, in which case, the process is called multi-factor authentication. With regard to one-factor authentication, the security of an authentication instance is dependent on the method used. This means that if this method is compromised, the whole system will be put at risk. With regard to multi-factor authentication, an adversary would have to compromise more than one method to compromise an authentication process. The work factor needed to launch a successful attack on an authentication solution is proportional to the security level provided by the solution [57]. Therefore, the work factor to compromise multi-factor authentication is typically higher than its one-factor counterpart. For example, the work factor to compromise two-factor authentication where the factors are independent is $2^n + 2^k$, where n and k are, respectively, the security levels provided by the two methods used.

A number of technology leaders, e.g., Cisco, Apple, and Amazon, have made two-factor authentication, known as two-step verification, available to their users [58]. Some vendors, e.g., Amazon, have decided to make it optional to enhance usability [59].

C. AUTHENTICATION SOLUTIONS (METHODS, PROTOCOLS AND SYSTEMS)

Putting an authentication method into action requires the design or use of an authentication protocol. An authentication system is the implementation of an authentication protocol. In this section, we examine various authentication methods, protocols or systems (collectively referred to as authentication solutions) published in literature. Existing authentication solutions can be classified into two main categories: (i) ones proposed for non-IoT applications, and (ii) ones for IoT applications, as shown in Figure 3. The first category presents the solutions proposed for non-IoT applications such as electronic authentication. The second category, on the other hand, discusses the solutions proposed specifically for IoT applications such as SHome, smart health, industrial IoT, etc. Unlike the first category solutions, these solutions typically consider the features of IoT environments, e.g., the existence of resource constrained devices, in their design.

1) SOLUTIONS FOR NON-IoT APPLICATIONS

a: USERNAME AND PASSWORD AUTHENTICATION

In username and password authentication, a requestor uses a username and password, i.e., something you know, to verify his identity to a responder, as shown in Figure 4. The security level of this method relies on the password and underlying communication channel. Hence, it may be vulnerable to password and communication channel attacks [60]. Password policies such as minimum length, i.e., the minimum number of characters, and complexity, i.e., the type of characters used, are typically enforced to overcome password related attacks, such as brute force attacks [61]. Passwords are often encrypted during transit to counter communication channel attacks, such as man-in-the-middle attack.

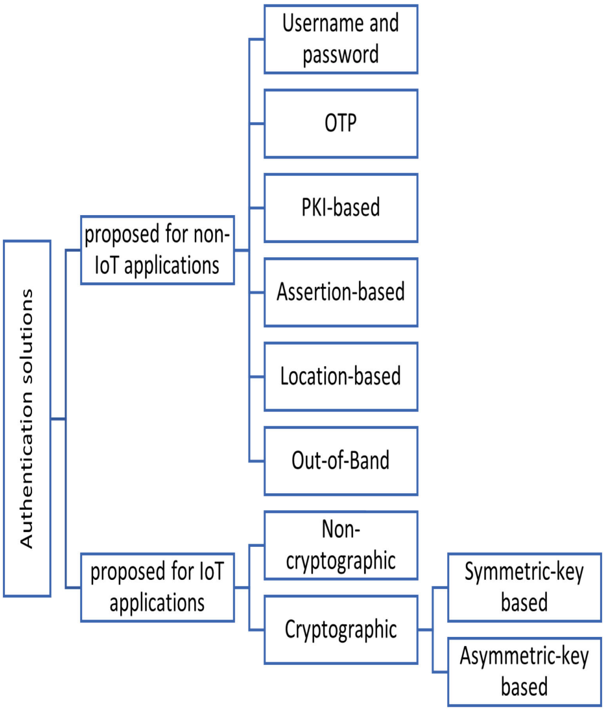

FIGURE 3. Authentication solutions.

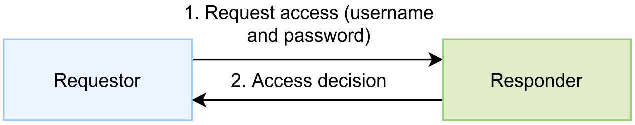

FIGURE 4. Username and password authentication.

b: ONE-TIME PASSWORD AUTHENTICATION

A one-time password (OTP), also known as a dynamic password, is a password that is valid for only one session. Unlike static passwords, used in username and password authentication, the OTP, if implemented correctly, can withstand password leaks, guessing, and replay attacks [62]. This is because each session has a unique password, as shown in Figure 5. The OTP is often delivered through a hard token, i.e., a dedicated OTP device such as Feitian OTP C200 [63], or a soft token, i.e., an OTP application such as Google Authenticator [64]. The security level of the OTP relies on the token used, token access methods, and underlying communication channel.

c: PUBLIC KEY INFRASTRUCTURE (PKI)-BASED AUTHENTICATION

PKI-based authentication uses a PKI token that contains a secret cryptographic key. This key could be stored on a hardware device, e.g., a smart card, in which case, it is called a PKI hardware token, or on a software, e.g., as a file object, in which case, it is called a PKI software token. As shown in Figure 6, the access to these tokens is typically protected by another authentication method, e.g., a password.

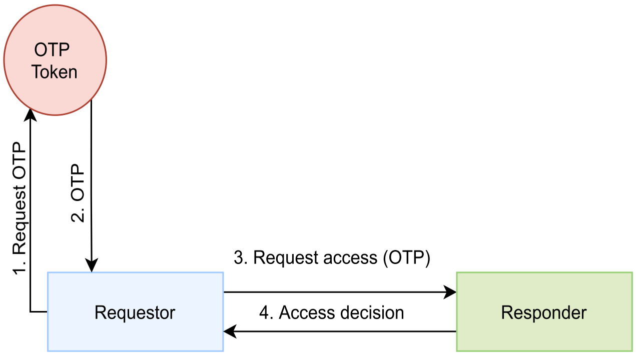

FIGURE 5. An example of OTP authentication.

A requestor uses the PKI token, i.e., something you have, and the password, i.e., something you know, to verify his identity to a responder. Therefore, the PKI-based authentication is a multi-factor process.

When a PKI hardware token is used, a pair of asymmetric keys, i.e., a public and private key, is generated and stored in the token. While the private key never leaves the token, the public key is sent to a certificate authority (CA) for registration. Once registered, the public key certificate, signed by the CA, is stored on the token. In order to compromise the PKI hardware token, an adversary would have to obtain the token, i.e., factor-1, and pass the other authentication method protecting the key stored on the token, i.e., factor-2. Thus, the security level of this method relies on the token and its access methods.

When a PKI software token is used, a private key and its public key certificate are stored in a keystore. An example is the Microsoft Cryptographic Service Provider (CSP) keystore. In order to compromise this method, an adversary would have to access the operating system hosting the keystore, i.e., factor-1, and know the keystore password, i.e., factor-2. Thus, the security level of this method relies on the underlying operating system and the keystore access method.

Although the PKI hardware authentication may provide mobility and higher security compared with the PKI software authentication, it may not be scalable due to the following reasons. The first is the additional cost imposed to acquire hardware tokens. The second reason is the lack of standard solutions. Hardware tokens are manufactured by a number of companies. These companies typically use their own software to manage their tokens. Thus, a number of software applications may need to be installed and managed which complicates the authentication process [65].

d: ASSERTION-BASED AUTHENTICATION

An assertion is a statement from an Identity provider (IdP) to a responder, also referred to as a service provider (SP), containing the results of an authentication instance. The instance is typically initiated by a requestor. The requestor verifies its identity to the IdP to gain access to the responder or a service hosted by the responder. The IdP then creates an assertion.

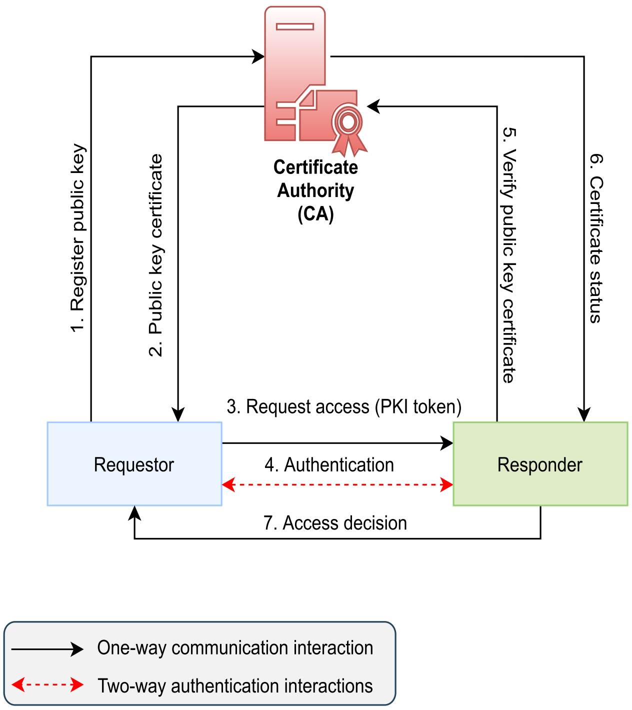
Certificate Authority (CA)
1. Register public key
2. Public key certificate
3. Request access (PKI token)
4. Authentication
5. Verify public key certificate
6. Certificate status
7. Access decision
Requestor
Responder
One-way communication interaction
Two-way authentication interactions

FIGURE 6. An example of PKI-based authentication.

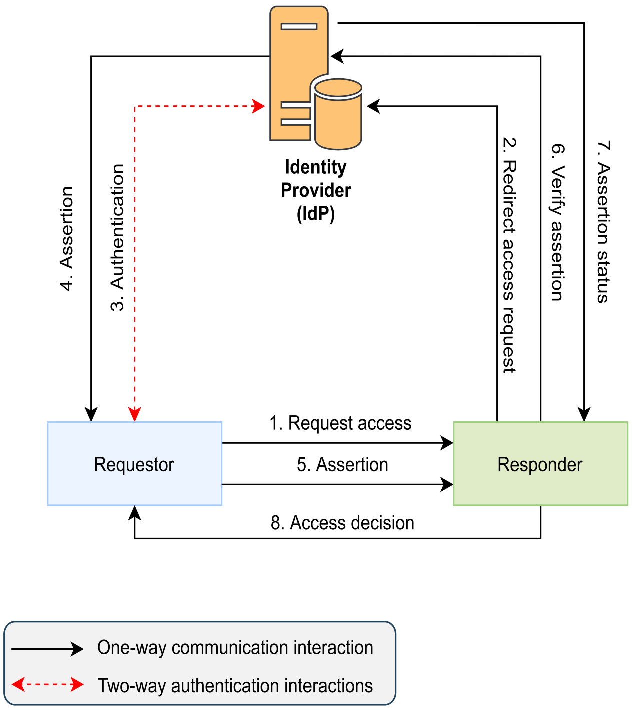

FIGURE 7. An example of assertion-based authentication.

In addition to the authentication result, the assertion may contain additional information, e.g., access privileges. Once the assertion is created, the IdP either sends it directly to the requestor, i.e., push mode assertion, or indirectly through sending a reference to the assertion instead, i.e., pull mode assertion. In the push mode assertion, the responder uses the assertion forwarded by the requestor to verify its identity. In the pull mode assertion, the responder uses the assertion reference sent by the requestor to obtain the assertion from the IdP and verify the requestor's identity. An example of assertion-based authentication is depicted in Figure 7. Hypertext Transfer Protocol (HTTP) Cookies, Security Assertion Markup Language (SAML), and Kerberos are common methods used to implement assertion-based authentication [66].

(i) **Hypertext Transfer Protocol (HTTP) Cookies**

An HTTP cookie, also known as a browser cookie, is a small data file stored by a browser on a requestor device. The file may contain personal data, e.g., the requestor's identity, and/or behavioural data, e.g., browsing history. Based on their lifetime, HTTP cookies could be classified into two categories: session cookies, i.e., cookies that are stored on a temporary memory, and persistent cookies, i.e., cookies that are stored on a long-term memory. Session cookies are typically deleted once the session is closed, whereas persistent cookies can be used in a number of sessions [67]. These cookies could be used to carry authentication information to re-authenticate a requestor to a responder or enable single sign-on (SSO) access within a domain. In the SSO access, the requestor authenticates itself once to obtain an access credential, and then, it uses the same access credential, obtained from the authentication session, to access a number of responders. To implement cross-domain SSO access, other methods, e.g., SAML, are typically used instead of cookies. This is because cookies are not built to be shared across domains [68].

Although HTTP cookies are widely used, they are typically accompanied by a number of security and privacy concerns. One such concern is that they could be misused intentionally, e.g., tracking requestors without their consent, or unintentionally, e.g., exposing private information through negligence [69]. Another concern is that if an adversary is able to obtain such a cookie, then the adversary would be able to impersonate the requestor [70]. The security level of this method relies on the IdP, and the cookie delivery, storage, and access methods.

(ii) **Security Assertion Markup Language (SAML)**

SAML is an Extensible Markup Language (XML)-based framework used to transfer access credentials between trusted entities. A SAML assertion is an XML-based document. The document can carry three types of statements: authentication, authorization, and attribute statements. The first two statements are used to transfer authentication and authorization information, respectively. The latter statement, i.e., attribute statement, is used to transfer additional information related to the requestor [71]. These statements could be used to re-authenticate a requestor to a responder or enable SSO access within a domain or cross-domain [72]. The security level of this method relies on the IdP(s), and the assertion delivery, storage, and access methods.

(iii) **Kerberos**

Kerberos is a widely used symmetric-key based authentication protocol that uses tickets, i.e., encrypted secrets, to provide client-server authentication. In Kerberos, the IdP uses two servers: authentication server (AS) and ticket-granting server (TGS). A requestor authenticates itself to the AS to obtain a ticket-granting ticket. The requestor then uses this ticket to authenticate itself to the TGS to get a service-granting ticket. The service-granting ticket is then used to access the responder [73]. To verify the ownership of the tickets, the requestor generates a fresh authenticator and attach it to each ticket. Kerberos is commonly used over insecure communication channels, such as the Internet. This is because access credentials, i.e., Kerberos tickets and authenticators, are always encrypted [74]. The security level of Kerberos relies on the AS, TGS, and symmetric-key cipher used.

e: LOCATION-BASED AUTHENTICATION

In location-based authentication, the identity of a requestor is verified using his position, as depicted in Figure 8. A number of methods, e.g., Global Positioning System (GPS) coordinates, Internet Protocol (IP) address, or proximity to a certain device, could be used to determine the requestor location. Unlike other authentication methods, discussed in this paper, location-based authentication, if implemented implicitly, does not require the requestor input. Hence, this type of authentication could be invisible to the requestor [75]. Although this may enhance the requestor experience, an adversary could bypass authentication by simply forging his location, e.g., using the GPS coordinates of a trusted location. Additional authentication methods, such as a username and password, are often used, to strengthen location-based authentication. The security level of location-based authentication relies on the location, method used to generate the location-based token, token access methods, and underlying communication channel.

An example is the IP-based authentication. In this method, the responder verifies the identity of the requestor through comparing his IP address to a list of approved IP addresses. To overcome the static nature of IP-based authentication, a range of IP addresses could be specified instead of the list. This method, i.e., specifying a range of approved IP addresses, is often used in organizations to restrict access to their internal networks [65].

f: OUT-OF-BAND AUTHENTICATION

Out-of-Band (OOB) authentication is a multi-factor authentication process. It uses two separate channels, a communication channel, and an out-of-band channel, to authenticate a requestor [76]. The requestor uses the communication channel to verify his identity using an authentication method such as a username and password, i.e., factor-1. If this authentication is successful, the responder sends a temporary access token such as an OTP, i.e., factor-2, through the out-of-band channel. Once the token is received, the requestor uses it to verify his identity to the responder, as shown in Figure 9. OOB authentication is commonly used in financial institutions. This is because it provides a higher assurance level as an adversary would have to compromise two separate channels to bypass authentication. Short Message Service (SMS), phone calls, and emails are common methods used as out-of-band channels [77]. The security level of the OOB authentication relies on the authentication methods used, and underlying communication and out-of-band channels.

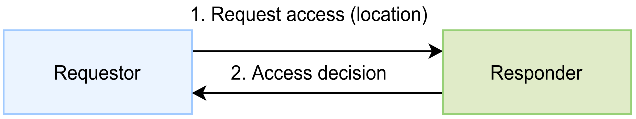

FIGURE 8. Location-based authentication.

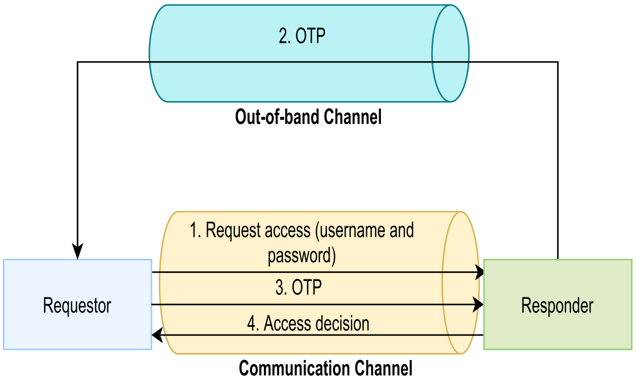

FIGURE 9. An example of OOB authentication.

2) SOLUTIONS DESIGNED FOR IoT

Existing IoT authentication solutions can be classified into two categories: non-cryptographic, and cryptographic solutions. Depending on the cryptographic algorithms used, the latter category, i.e., cryptographic solutions, can be classified into further two subcategories: symmetric-key based, and asymmetric-key based solutions. Next, we review some of the solutions proposed recently. The solutions are then analysed against the security requirements to identify research gaps. A summary of this analysis is given in Table 2.

a: NON-CRYPTOGRAPHIC SOLUTIONS

To reduce authentication overhead imposed by cryptographic operations, a number of non-cryptographic solutions, also known as ultra-lightweight solutions, have been proposed.

Tewari and Gupta [9] proposed a device-to-device Radio Frequency Identification (RFID) based protocol that uses rotation and bitwise exclusive-OR (XOR) operation to achieve mutual authentication. Three entities involved in the protocol: a tag, reader, and database. The reader initiates the authentication process by sending a notification message to the tag. Upon the receipt of the message, the tag sends its identity to the reader. The reader then queries the database to verify the existence of the tag. If the tag exists, the reader generates two random numbers to compute a challenge. The challenge, containing the random numbers, shared key, and tag identity, is then computed using rotation and XOR. Upon the receipt of the challenge, the tag obtains the random numbers to verify the authenticity of the message using the same operations, i.e., rotation and XOR. If verified, the reader has been authenticated. Similarly, the tag then computes a response and sends it to the reader. If verified, the tag has been authenticated to the reader. The main limitation of the protocol is that it could be vulnerable to DoS attacks. This is because the notification message, used to initiate the authentication process, is sent in plaintext over an insecure communication channel. An adversary could flood the responder with many notification messages. In addition, Wang *et al.* [78] show that the protocol is vulnerable against key disclosure attack.

A similar protocol as in [9], was proposed by Fan *et al.* [10]. However, authors attached a random number and timestamp to the notification message to counter replay and DoS attacks. Similar to [9], this protocol could be vulnerable to DoS attacks. This is because the tag does not verify the authenticity of the challenge before computing its response. It only uses the timestamp, received over insecure communication channel, to verify the freshness of the challenge. Hence, an adversary can forge a number of challenges to occupy the tag.

Martinez and Bossuet [11] proposed a device-to-device authentication protocol that uses Physically Unclonable Function (PUF) and XOR to achieve authentication. The requestor initiates the authentication process by sending an access request to the authentication server (AS). The AS replies with a challenge. Upon the receipt of the challenge, the requestor uses it to challenge its PUF to compute two responses. It then XOR the responses to construct and send a response message to the AS. Upon the receipt of the message, the AS computes its own response by retrieving the corresponding information from a secure database. After that, it compares its response to the message received. If verified, the requestor has been authenticated to the AS. The main limitation of the protocol is that the PUF based challenges and responses are transmitted in plaintext. As a result, an adversary may intercept these communications to learn about the devices or launch other attacks, e.g., replay attacks.

To address this limitation, Gu *et al.* [12] proposed a device-to-device PUF based protocol which uses deception techniques, e.g., a fake PUF circuit and random numbers, to prevent the adversary from predicting the PUF responses. Although this approach may enhance the security level of PUF based protocols, it adds additional authentication overhead. The main limitation of the PUF based protocols is that the responder in the authentication process has to be a powerful device, e.g., an authentication server. This is because constrained device may not be able keep track of the PUF responses of each of the devices involved.

The main advantage of non-cryptographic solutions is their low authentication overhead. Although they often use additional hardware, e.g., an RFID tag or a PUF circuit, to enhance their security, their levels of assurance are typically lower than that of cryptographic solutions.

b: CRYPTOGRAPHIC SOLUTIONS

(i) **Symmetric-key based solutions**

To reduce the computation costs imposed by cryptographic operations, most cryptographic authentication solutions, known as lightweight authentication solutions, use symmetric-key based algorithms.

To address user authentication, Amin *et al.* [13] proposed a mobile based user-to-device protocol that uses a username and password, hash and XOR to achieve mutual authentication. The user uses the username and password to verify his identity to a mobile device. If verified, the device generates a random number to construct a request message using hash and XOR. Then, it sends the message to a gateway (GW). Upon the verification of the request, the GW generates a random number and a secret key to construct and send its own request message. Upon the receipt of this message, the target device generates a session key. It then constructs a response message, containing the session key, using the same methods, i.e., hash and XOR, and sends it to the GW. If verified, the GW creates a response message and sends it to the mobile device. Upon the verification of the message, the device sends a confirmation message to the GW. Wu *et al.* [14] show that the protocol is vulnerable against offline guessing attacks. To address this limitation, a similar protocol as in [13], was proposed by Wu *et al.* [14]. However, authors used pseudo identities to prevent guessing attacks.

To provide a higher security level, Wazid *et al.* [15] proposed a multi-factor mobile based user-to-device authentication protocol. In addition to hash and XOR, the protocol uses a username and password, and biometric information to verify the identity of a user. Fotouhi *et al.* [16] show that the protocol is vulnerable against impersonation attacks. To address this limitation, Fotouhi *et al.* [16] proposed a similar multi-factor authentication protocol. However, the protocol uses a username and password, and a registered mobile device to verify the identity of a user.

To provide confidentiality in user authentication, a similar protocol as in [15], was proposed by Liu *et al.* [17]. However, authors used Physical Unclonable Function (PUF) and symmetric encryption to secure the authentication process.

To address device authentication, A similar protocol as in [9], was proposed by Gope *et al.* [18]. However, authors used location, XOR and hash to verify the integrity of messages exchanged and secure authentication. The main limitation of the protocol is that the RFID reader does not verify a received access request before computing and sending its own message to the authentication server. Thus, an adversary could occupy the reader by sending many access requests.

Lara *et al.* [19] proposed a protocol that uses XOR and hash to achieve authentication. Authors attached a timestamp to each protocol message to verify its freshness. Although this approach may strengthen the protocol against replay attacks, it may introduce additional issues. This is because the clocks of some IoT devices may not be synchronized.

A similar protocol as in [11], was proposed by Mahalat *et al.* [20]. In addition to XOR and PUF operations, the protocol uses hash to verify message integrity. The main limitation of the protocol is that it could be vulnerable to DoS attacks. This is due to the reason discussed earlier in [9].

Liang *et al.* [21] proposed an RFID and double PUF based protocol. In addition to RFID and PUF operations, the protocol uses hash, XOR and character padding operation to secure authentication. The main limitation of the protocol is that it requires a requestor to have an RFID tag and PUF circuit. Gope *et al.* [79] show that the protocol is vulnerable against replay and impersonation attacks.

To provide confidentiality in device authentication, Fan *et al.* [22] proposed a device-to-device RFID based protocol that uses XOR, permutation, rotation and symmetric encryption to secure authentication. Adeli *et al.* [80] show that the protocol is vulnerable against a number of attacks, such as replay and impersonation attacks.

Lai *et al.* [23] proposed a multiDevice-to-device authentication protocol. The protocol uses location, message authentication code (MAC) and XOR to authenticate a group of devices to a server. Modiri *et al.* [24] show that the protocol is vulnerable against impersonation attacks. To address this limitation, a similar protocol as in [23], was proposed by Modiri *et al.* [24]. The protocol uses MAC and temporary identities to secure authentication.

(ii) **Asymmetric-key based solutions**

To address user authentication, Chen *et al.* [25] proposed a multi-factor user-to-device authentication protocol. The protocol uses two factors: a smart card, i.e., factor-1, and a username and password, i.e., factor-2. In addition, the protocol uses hash, XOR and asymmetric encryption using the Elliptic Curve Cryptography (ECC) algorithm to secure authentication. Lee *et al.* [81] show that the protocol is vulnerable against a number of attacks, including replay and smart card stolen attack where an adversary is able to impersonate a legitimate user if the user's card is in his possession.

To address this limitation, a similar mobile based user-to-device authentication protocol as in [15], was proposed by Nikravan and Reza [26]. In addition to the operations used in [15], the protocol uses the ECC algorithm to secure authentication. Shamshad *et al.* [82] show that the protocol is vulnerable against impersonation attacks.

To address device authentication, Chatterjee *et al.* [27] proposed a device-to-device PUF based protocol. In addition to the PUF operations, the protocol uses hash, XOR and identity-based encryption (IBE) to secure authentication. To eliminate the need for a Private Key Generator (PKG), i.e., a trusted third party responsible for generating private keys in IBE [83], each device generates a public/private key pair using its PUF. Braeken [28] shows that the that the protocol is vulnerable against replay and impersonation attacks.

To address this limitation, a similar protocol as in [27], was proposed by Braeken [28]. The protocol uses the Elliptic Curve Qu-Vanstone certificate (ECQV) scheme, an implicit certificate scheme based on the ECC algorithm, to secure authentication. The main limitation of the protocol is that it could be vulnerable to DoS attacks. This is due to the same reason discussed in [9].

Naeem *et al.* [29] proposed a device-to-device RFID based protocol. The protocol uses hash and ECC algorithm to secure authentication. Izza *et al.* [30] show the that the protocol is vulnerable against a number of attacks, including replay and impersonation attacks.

To address this limitation, a similar protocol as in [29], was proposed by Izza *et al.* [30]. In addition to the operations used in [29], the protocol uses the Elliptic Curve Digital Signature algorithm with message recovery to secure communication. The main limitation of the protocol is the same as in [28].

Shen *et al.* [31] proposed a device-to-multiDevice protocol to authenticate a personal digital assistant device to a group of sensors. The protocol uses the ECC algorithm and message authentication code (MAC) to secure authentication. Harbi *et al.* [84] show that the protocol is vulnerable against a number of attacks, including replay and impersonation attacks. To address this limitation, an improved version of the protocol was presented by Liu *et al.* [32].

VII. FURTHER DISCUSSIONS

The above related work analysis indicates that there are rooms for improvements in existing authentication solutions when being applied to IoT environments. This is due to the following observations.

Firstly, some special features of an IoT environment, such as the existence of resource constrained devices, the varying sensitivity levels of devices or resources hosted in the environment and automatic M2M communications, are not considered in the design of the existing authentication solutions. The consideration of these features in the design of an authentication solution for an IoT environment may offer a more effective and efficient protection.

Secondly, existing authentication solutions designed for IoT environments are mainly single-factor based and provide a single level protection (or a single level of assurance). They assume that resources or interactions being protected have the same level of sensitivity. This single LoA, one-size-fits-all, approach to authentication may not be proper in some IoT applications. For resources with a higher sensitivity level, a higher level of protection, i.e., a higher assurance level, should be provided. However, a higher assurance level often comes with a higher level of overhead cost, which can be particularly detrimental to devices with constrained capabilities.

Although there are multi-factor and multi LoA authentication solutions in literature and in our daily life activities, e.g., on-line banking. These solutions are mostly focused on user-to-device authentication; they are not readily

TABLE 2. Security analysis of IoT authentication solutions.

The State-of-the-Art			S1				S2	S3	S4	S5	S6
			S1.1	S1.2	S1.3	S1.4					
Non-cryptographic		Tewari and Gupta [9]	x	✓	x	x	x	x	x	x	x
		Fan et al. [10]	x	✓	x	x	o	x	x	x	x
		Martinez and Bossuet [11]	x	✓	x	x	x	x	x	x	x
		Gu et al. [12]	x	✓	x	x	x	x	x	x	✓
Cryptographic	Symmetric-key based	Amin et al. [13]	✓	x	x	x	o	o	o	x	x
		Wu et al. [14]	✓	x	x	x	x	✓	o	x	✓
		Wazid et al. [15]	✓	x	x	x	✓	o	o	✓	x
		Fotouhi et al. [16]	✓	x	x	x	x	o	x	x	✓
		Liu et al. [17]	✓	x	x	x	✓	o	✓	x	✓
		Gope et al. [18]	x	✓	x	x	x	o	x	x	x
		Lara et al. [19]	x	✓	x	x	✓	✓	o	x	✓
		Mahalat et al. [20]	x	✓	x	x	x	o	x	x	x
		Liang et al. [21]	x	✓	x	x	x	x	x	x	x
		Fan et al. [22]	x	✓	x	x	✓	x	o	x	x
		Lai et al. [23]	x	x	x	✓	x	o	x	x	x
		Modiri et al. [24]	x	x	x	✓	✓	o	x	x	✓
	Asymmetric-key based	Chen et al. [25]	✓	x	x	x	✓	✓	o	x	x
		Nikravan and Reza [26]	✓	x	x	x	✓	✓	✓	x	x
		Chatterjee et al. [27]	x	✓	x	x	o	o	x	x	x
		Braeken [28]	x	✓	x	x	✓	o	x	x	x
		Naeem et al. [29]	x	✓	x	x	x	o	o	x	x
		Izza et al. [30]	x	✓	x	x	o	o	o	x	x
		Shen et al. [31]	x	x	✓	x	x	o	✓	x	x
		Liu et al. [32]	x	x	✓	x	x	o	✓	x	✓

✓: supported ; x: not supported; o: partially supported.
S1: entity authentication: S1.1: user-to-device authentication, S1.2: device-to-device authentication, S1.3: device-to-multiDevice authentication, and S1.4: multiDevice-to-device authentication; S2: message freshness; S3: message integrity; S4: confidentiality; S5: authorization; S6: availability.

applicable to device-to-device authentication, particularly for IoT devices that are typically heterogeneous and have varying levels of computational capabilities. Therefore, further research is necessary to examine how to provide multi-level authentication assurance and do so efficiently, particularly for device-to-device interaction or authentication scenarios.

VIII. CONCLUSION

Although SHome technologies have the potential to provide benefits and improve our quality of life, they also introduce new security problems or challenges. In this paper, we have carried out a systematic analysis of security problems and threats in relation to authentication in an SHome environment, and critically analysed state-of-the-art authentication solutions in the context. In doing so, we have examined an SHome environment, and derived a generic model for the environment. Using the model, we then analyzed security threats or attacks in relation to authentication. Based on the analysis, we have specified a list of security requirements for a desirable authentication solution for the SHome environment. Against the requirements, we have critically examined existing authentication solutions proposed for IoT environments, identifying areas for improvements. As part of our future work, we will explore the design and implementation of a multi-level and interaction-based authentication solution to support a more secure and flexible authentication service in IoT.

REFERENCES

[1] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Bus. Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[2] H. Yang, W. Lee, and H. Lee, "IoT smart home adoption: The importance of proper level automation," *J. Sensors*, vol. 2018, pp. 1–11, May 2018.

[3] N. Balta-Ozkan, B. Boteler, and O. Amerighi, "European smart home market development: Public views on technical and economic aspects across the United Kingdom, Germany and Italy," *Energy Res. Social Sci.*, vol. 3, pp. 65–77, Sep. 2014.

[4] M. Alaa, A. A. Zaidan, B. B. Zaidan, M. Talal, and M. L. M. Kiah, "A review of smart home applications based on Internet of Things," *J. Netw. Comput. Appl.*, vol. 97, pp. 48–65, Nov. 2017.

[5] H. Lin and N. W. Bergmann, "IoT privacy and security challenges for smart home environments," *Information*, vol. 7, no. 3, p. 44, 2016.

[6] S. AlJanah and N. Zhang, "An authentication framework for the Internet of Things," presented at the Univ. Manchester Comput. Sci. Res. Symp. (CSRS@UoM), Manchester, U.K., 2019.

[7] M. El-Hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A survey of Internet of Things (IoT) authentication schemes," *Sensors*, vol. 19, no. 5, p. 1141, Mar. 2019.

[8] M. Saadeh, A. Sleit, M. Qatawneh, and W. Almobaideen, "Authentication techniques for the Internet of Things: A survey," in *Proc. Cybersecur. Cyberforensics Conf. (CCC)*, Aug. 2016, pp. 28–34.

[9] A. Tewari and B. B. Gupta, "Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags," *J. Supercomput.*, vol. 73, no. 3, pp. 1085–1102, Mar. 2017.

[10] K. Fan, P. Song, and Y. Yang, "ULMAP: Ultralightweight NFC mutual authentication protocol with pseudonyms in the tag for IoT in 5G," *Mobile Inf. Syst.*, vol. 2017, pp. 1–7, Apr. 2017.

[11] B. Ovilla-Martinez and L. Bossuet, "Restoration protocol: Lightweight and secur devices authentication based on PUF," in *Proc. IFIP/IEEE Int. Conf. Very Large Scale Integr. (VLSI-SoC)*, Oct. 2017, pp. 1–6.

[12] C. Gu, C.-H. Chang, W. Liu, S. Yu, Y. Wang, and M. O'Neill, "A modeling attack resistant deception technique for securing lightweight-PUF-based authentication," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 6, pp. 1183–1196, Jun. 2021.

[13] R. Amin, S. K. H. Islam, G. P. Biswas, M. K. Khan, and N. Kumar, "A robust and anonymous patient monitoring system using wireless medical sensor networks," *Future Gener. Comput. Syst.*, vol. 80, pp. 483–495, Mar. 2018.

[14] F. Wu, X. Li, A. K. Sangaiah, L. Xu, S. Kumari, L. Wu, and J. Shen, "A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks," *Future Gener. Comput. Syst.*, vol. 82, pp. 727–737, May 2018.

[15] M. Wazid, A. K. Das, and A. V. Vasilakos, "Authenticated key management protocol for cloud-assisted body area sensor networks," *J. Netw. Comput. Appl.*, vol. 123, pp. 112–126, Dec. 2018.

[16] M. Fotouhi, M. Bayat, A. K. Das, H. A. N. Far, S. M. Pournaghi, and M. A. Doostari, "A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT," *Comput. Netw.*, vol. 177, Aug. 2020, Art. no. 107333.

[17] Z. Liu, C. Guo, and B. Wang, "A physically secure, lightweight three-factor and anonymous user authentication protocol for IoT," *IEEE Access*, vol. 8, pp. 195914–195928, 2020.

[18] P. Gope, R. Amin, S. K. H. Islam, N. Kumar, and V. K. Bhalla, "Lightweight and privacy-preserving RFID authentication scheme for distributed IoT infrastructure with secure localization services for smart city environment," *Future Gener. Comput. Syst.*, vol. 83, pp. 629–637, Jun. 2018.

[19] E. Lara, L. Aguilar, M. A. Sanchez, and J. A. García, "Lightweight authentication protocol for M2M communications of resource-constrained devices in industrial Internet of Things," *Sensors*, vol. 20, no. 2, p. 501, Jan. 2020.

[20] M. H. Mahalat, S. Saha, A. Mondal, and B. Sen, "A PUF based light weight protocol for secure WiFi authentication of IoT devices," in *Proc. 8th Int. Symp. Embedded Comput. Syst. Design (ISED)*, Dec. 2018, pp. 183–187.

[21] W. Liang, S. Xie, J. Long, K.-C. Li, D. Zhang, and K. Li, "A double PUF-based RFID identity authentication protocol in service-centric Internet of Things environments," *Inf. Sci.*, vol. 503, pp. 129–147, Dec. 2019.

[22] K. Fan, Q. Luo, K. Zhang, and Y. Yang, "Cloud-based lightweight secure RFID mutual authentication protocol in IoT," *Inf. Sci.*, vol. 527, pp. 329–340, Jul. 2020.

[23] C. Lai, R. Lu, D. Zheng, H. Li, and X. Shen, "GLARM: Group-based lightweight authentication scheme for resource-constrained machine to machine communications," *Comput. Netw.*, vol. 99, pp. 66–81, Apr. 2016.

[24] M. M. Modiri, J. Mohajeri, and M. Salmasizadeh, "A novel group-based secure lightweight authentication and key agreement protocol for machine-type communication," *Scientia Iranica*, pp. 1–14, Feb. 2021. [Online]. Available: http://scientiairanica.sharif.edu/article_22225.html, doi: 10.24200/SCI.2021.54832.3936.

[25] Y. Chen, L. López, J.-F. Martínez, and P. Castillejo, "A lightweight privacy protection user authentication and key agreement scheme tailored for the Internet of Things environment: LightPriAuth," *J. Sensors*, vol. 2018, pp. 1–16, Sep. 2018.

[26] M. Nikravan and A. Reza, "A multi-factor user authentication and key agreement protocol based on bilinear pairing for the Internet of Things," *Wireless Pers. Commun.*, vol. 111, no. 1, pp. 463–494, Mar. 2020.

[27] U. Chatterjee, R. S. Chakraborty, and D. Mukhopadhyay, "A PUF-based secure communication protocol for IoT," *ACM Trans. Embed. Comput. Syst.*, vol. 16, no. 3, pp. 1–25, Apr. 2017.

[28] A. Braeken, "PUF based authentication protocol for IoT," *Symmetry*, vol. 10, no. 8, p. 352, 2018.

[29] M. Naeem, S. A. Chaudhry, K. Mahmood, M. Karuppiah, and S. Kumari, "A scalable and secure RFID mutual authentication protocol using ECC for Internet of Things," *Int. J. Commun. Syst.*, vol. 33, no. 13, p. e3906, Sep. 2020.

[30] S. Izza, M. Benssalah, and K. Drouiche, "An enhanced scalable and secure RFID authentication protocol for WBAN within an IoT environment," *J. Inf. Secur. Appl.*, vol. 58, May 2021, Art. no. 102705.

[31] J. Shen, S. Chang, J. Shen, Q. Liu, and X. Sun, "A lightweight multi-layer authentication protocol for wireless body area networks," *Future Gener. Comput. Syst.*, vol. 78, pp. 956–963, Jan. 2018.

[32] X. Liu, C. Jin, and F. Li, "An improved two-layer authentication scheme for wireless body area networks," *J. Med. Syst.*, vol. 42, no. 8, pp. 1–14, Aug. 2018.

[33] K. S. Roy and H. K. Kalita, "A survey on authentication schemes in IoT," in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Dec. 2017, pp. 202–207.

[34] S. Jiao and R. P. Liu, "A survey on physical authentication methods for smart objects in IoT ecosystem," *Internet Things*, vol. 6, Jun. 2019, Art. no. 100043.

[35] M. A. Ferrag, L. A. Maglaras, H. Janicke, J. Jiang, and L. Shu, "Authentication protocols for Internet of Things: A comprehensive survey," *Secur. Commun. Netw.*, vol. 2017, pp. 1–41, Nov. 2017.

[36] Y. Atwady and M. Hammoudeh, "A survey on authentication techniques for the Internet of Things," in *Proc. Int. Conf. Future Netw. Distrib. Syst.*, Jul. 2017, Art. no. 8.

[37] P. Wang, F. Ye, and X. Chen, "A smart home gateway platform for data collection and awareness," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 87–93, Sep. 2018.

[38] S. Yoon, H. Park, and H. S. Yoo, "Security issues on smarthome in IoT environment," in *Computer Science and its Applications*. Berlin, Germany: Springer, 2015, pp. 691–696.

[39] D. Marikyan, S. Papagiannidis, and E. Alamanos, "A systematic review of the smart home literature: A user perspective," *Technol. Forecasting Social Change*, vol. 138, pp. 139–154, Jan. 2019.

[40] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2017, pp. 618–623.

[41] A. Jacobsson, M. Boldt, and B. Carlsson, "A risk analysis of a smart home automation system," *Future Gener. Comput. Syst.*, vol. 56, pp. 719–733, Mar. 2016.

[42] J. Bugeja, A. Jacobsson, and P. Davidsson, "On privacy and security challenges in smart connected homes," in *Proc. Eur. Intell. Secur. Informat. Conf. (EISIC)*, Aug. 2016, pp. 172–175.

[43] C. Lee, L. Zappaterra, K. Choi, and H.-A. Choi, "Securing smart home: Technologies, security challenges, and security requirements," in *Proc. IEEE Conf. Commun. Netw. Secur.*, Oct. 2014, pp. 67–72.

[44] B. L. R. Stojkoska and K. V. Trivodaliev, "A review of Internet of Things for smart home: Challenges and solutions," *J. Cleaner Prod.*, vol. 140, no. 3, pp. 1454–1464, 2017.

[45] O. Horyachyy, "Comparison of wireless communication technologies used in a smart home: Analysis of wireless sensor node based on Arduino in home automation scenario," M.S. thesis, Blekinge Inst. Technol., 2017. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1118965/FULLTEXT02.pdf

[46] D. Meyer, J. Haase, M. Eckert, and B. Klauer, "A threat-model for building and home automation," in *Proc. IEEE 14th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2016, pp. 860–866.

[47] C. Bormann, M. Ersue, and A. Keranen, "Terminology for constrained node networks," Internet Eng. Task Force, USA, Tech. Rep. RFC 7228, 2014, pp. 8–10. [Online]. Available: https://www.rfc-editor.org/rfc/pdfrfc/rfc7228.txt.pdf

[48] S. Yoon, J. Kim, and Y. Jeon, "Security considerations based on classification of IoT device capabilities," in *Proc. 9th Int. Conf. Adv. Service Comput.*, 2017, pp. 1–3.

[49] D. P. Acharjya and M. K. Geetha, *Internet of Things: Novel Advances and Envisioned Applications*, vol. 25. Cham, Switzerland: Springer, 2017.

[50] M. Schwartz, *Internet of Things With Arduino Cookbook*. Birmingham, U.K.: Packt, 2016.

[51] M. A. Mustafa, "Smart grid security: Protecting users' privacy in smart grid applications," Ph.D. dissertation, Dept. Comput. Sci., Univ. Manchester, Manchester, U.K., 2015.

[52] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, "Internet of Things: Security vulnerabilities and challenges," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2015, pp. 180–187.

[53] Y. Zhang, M. Yutaka, M. Sasabe, and S. Kasahara, "Attribute-based access control for smart cities: A smart-contract-driven framework," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6372–6384, Apr. 2021.

[54] P. S. Teh, "Using users' touch dynamics biometrics to enhance authentication on mobile devices," Ph.D. dissertation, Dept. Comput. Sci., Univ. Manchester, Manchester, U.K., 2019.

[55] A. Ali, H. Rafique, T. Arshad, M. A. Alqarni, S. H. Chauhdary, and A. K. Bashir, "A fractal-based authentication technique using Sierpinski triangles in smart devices," *Sensors*, vol. 19, no. 3, p. 678, Feb. 2019.

[56] D. Gibson, "CompTIA security+: Get certified get ahead: SY0-501 study guide," Comput. Technol. Ind. Assoc., USA, 2017, pp. 103-104.

[57] C. Sweeney, "Equivalencies in security," REDCOM Lab., USA, Tech. Rep. 20190716, 2019. [Online]. Available: https://www.redcom.com/wp-content/uploads/2019/08/08-2019-Equivalencies-in-Security.pdf

[58] X. L. Liang, "Two-factor human authentication," B.S. thesis, Tunku Abdul Rahman Univ., Perak, Malaysia, 2018.

[59] *Amazon: Changing Two-Step Verification Settings*. Accessed: Jan. 23, 2020. [Online]. Available: https://www.amazon.sg/gp/help/customer/display.html?nodeId=202073720

[60] C. Doğanay and A. Küpçü, "Comparative survey on single password authentication techniques," in *Proc. Int. Conf. Inf. Secur. Cryptol. (ISCTURKEY)*, Dec. 2020, pp. 5–10.

[61] P. A. Grassi, J. L. Fenton, E. M. Newton, R. A. Perlner, A. R. Regenscheid, W. E. Burr, J. P. Richer, N. B. Lefkovitz, J. M. Danker, Y. Y. Choong, and Y. Y. Greene, "Digital identity guidelines," Nat. Inst. Standards Technol., USA, Tech. Rep. NIST 800-63b, 2017. [Online]. Available: https://pages.nist.gov/800-63-3/sp800-63b.html

[62] D. Zhao and W. Luo, "One-time password authentication scheme based on the negative database," *Eng. Appl. Artif. Intell.*, vol. 62, pp. 396–404, Jun. 2017.

[63] *Feitian: Single Button OTP*. Accessed: Jan. 23, 2020. [Online]. Available: https://www.ftsafe.com/Products/OTP/Single_Button_OTP

[64] *Get Verification Codes With Google Authenticator*. Accessed: Jan. 23, 2020. [Online]. Available: https://support.google.com/accounts/answer/1066447?co=GENIE.Platform%3DAndroid&hl=en

[65] I. Lahmer and N. Zhang, "Towards a virtual domain based authentication on MapReduce," *IEEE Access*, vol. 4, pp. 1658–1675, 2016.

[66] L. Yao, "A structured approach to electronic authentication assurance level derivation," Ph.D. dissertation, Fac. Eng. Phys. Sci., Univ. Manchester, Manchester, U.K., 2010.

[67] P. Pinto, R. Lages, and M. Au-Yong-Oliveira, "Web cookies: Is there a trade-off between website efficiency and user privacy?" in *Proc. World Conf. Inf. Syst. Technol.* Cham, Switzerland: Springer, 2020, pp. 713–722.

[68] M. R. Brannsten, "Federated single sign on in disconnected, intermittent and limited (DIL) networks," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, May 2015, pp. 1–5.

[69] T. Bujlow, V. Carela-Español, B.-R. Lee, and P. Barlet-Ros, "A survey on web tracking: Mechanisms, implications, and defenses," *Proc. IEEE*, vol. 105, no. 8, pp. 1476–1510, Aug. 2017.

[70] H. Kwon, H. Nam, S. Lee, C. Hahn, and J. Hur, "(In-)security of cookies in HTTPS: Cookie theft by removing cookie flags," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1204–1215, 2020.

[71] P. A. Grassi, J. P. Richer, S. K. Squire, J. L. Fenton, E. M. Nadeau, N. B. Lefkovitz, J. M. Danker, Y.-Y. Choong, K. Greene, and M. F. Theofanos, "Digital identity guidelines: Federation and assertions [includes updates as of 03-02-2020]," Nat. Inst. Standards Technol., USA, Tech. Rep. NIST 800-63C, 2020, p. 36.

[72] Y. Wilson and A. Hingnikar, *Solving Identity Management in Modern Applications*. USA: Apress, 2019. [Online]. Available: https://link.springer.com/content/pdf/10.1007/978-1-4842-5095-2.pdf

[73] H. Li, Y. Niu, J. Yi, and H. Li, "Securing offline delivery services by using kerberos authentication," *IEEE Access*, vol. 6, pp. 40735–40746, 2018.

[74] S. William, *Cryptography and Network Security: Principles and Practice*. London, U.K.: Pearson, 2016.

[75] Y.-Y. Choong, K. Greene, and J. Franklin, "Usability and security considerations for public safety mobile authentication," Nat. Inst. Standards Technol., USA, Tech. Rep. NISTIR 8080, 2016, p. 14. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2016/nist.ir.8080.pdf

[76] S. Latvala, "Evaluation of out-of-band authentication channels," M.S. thesis, Dept. Programme Comput., Commun. Inf. Sci., Aalto Univ., Espoo, Finland, 2019.

[77] A. Kaur and K. Mustafa, "Qualitative assessment of authentication measures," in *Proc. 3rd Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, 2016, pp. 694–698.

[78] K.-H. Wang, C.-M. Chen, W. Fang, and T.-Y. Wu, "On the security of a new ultra-lightweight authentication protocol in IoT environment for RFID tags," *J. Supercomput.*, vol. 74, no. 1, pp. 65–70, 2018.

[79] P. Gope, B. Sikdar, and O. Millwood, "A scalable protocol level approach to prevent machine learning attacks on PUF-based authentication mechanisms for internet-of-medical-things," *IEEE Trans. Ind. Informat.*, early access, Jul. 9, 2021, doi: 10.1109/TII.2021.3096048.

[80] M. Adeli, N. Bagheri, S. Sadeghi, and S. Kumari, "χperbp: A cloud-based lightweight mutual authentication protocol," IACR Cryptol. ePrint Arch., 2021. [Online]. Available: https://eprint.iacr.org/2021/144.pdf

[81] J. Lee, S. Yu, M. Kim, Y. Park, and A. K. Das, "On the design of secure and efficient three-factor authentication protocol using honey list for wireless sensor networks," *IEEE Access*, vol. 8, pp. 107046–107062, 2020.

[82] S. Shamshad, K. Mahmood, and S. Kumari, "Comments on 'a multi-factor user authentication and key agreement protocol based on bilinear pairing for the Internet of Things,'" *Wireless Pers. Commun.*, vol. 112, pp. 1–4, Jan. 2020.

[83] L. Jiang, T. Li, X. Li, M. Atiquzzaman, H. Ahmad, and X. Wang, "Anonymous communication via anonymous identity-based encryption and its application in IoT," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–8, Nov. 2018.

[84] Y. Harbi, Z. Aliouat, A. Refoufi, S. Harous, and A. Bentaleb, "Enhanced authentication and key management scheme for securing data transmission in the Internet of Things," *Ad Hoc Netw.*, vol. 94, Nov. 2019, Art. no. 101948.

SALEM ALJANAH received the B.Sc. degree (Hons.) in information systems from Al-Imam Muhammad Ibn Saud Islamic University, Saudi Arabia, and the M.Sc. degree (Dist.) in information systems and technology from the University of Michigan, USA. He is currently pursuing the Ph.D. degree in computer science specializing in Internet of Things (IoT) Security with The University of Manchester, U.K.

His research interests include the IoT, information security, and applied cryptography.

NING ZHANG received the B.Sc. degree (Hons.) from Dalian Maritime University, China, and the Ph.D. degree from the University of Kent, U.K., both in electronics engineering.

Since 2000, she has been with the Department of Computer Science, The University of Manchester, U.K., where she is currently a Senior Lecturer. Her research interests include security in networked and distributed systems, applied cryptography, data privacy, trust, and digital right managements.

SIOK WAH TAY received the B.Sc. degree in security technology from Multimedia University, Malaysia, and the M.Sc. degree in human–computer interaction from the University of Bath, U.K. She is currently pursuing the Ph.D. degree in computer science with The University of Manchester, U.K.

Her research interests include security, the IoT, and human–computer interaction.

• • •