Certificateless-Signcryption-Based Three-Factor User Access Control Scheme for IoT Environment

Shobhan Mandal, Basudeb Bera, Anil Kumar Sutrala, Ashok Kumar Das, *Senior Member, IEEE*, Kim-Kwang Raymond Choo, *Senior Member, IEEE*, and Youngho Park, *Member, IEEE*

***Abstract*—User access control is a crucial requirement in any Internet of Things (IoT) deployment, as it allows one to provide authorization, authentication, and revocation of a registered legitimate user to access real-time information and/or service directly from the IoT devices. To complement the existing literature, we design a new three-factor certificateless-signcryption-based user access control for the IoT environment (CSUAC-IoT). Specifically, in our scheme, a user U's password, personal biometrics, and mobile device are used as the three authentication factors. By executing the login and access control phase of CSUAC-IoT, a registered user (U) and a designated smart device (S_i) can authorize and authenticate mutually via the trusted gateway node (GN) in a particular cell of the IoT environment. In our setting, the environment is partitioned into disjoint cells, and each cell will contain a certain number of IoT devices along with a GN. With the established session key between U and S_i, both entities can then communicate securely. In addition, CSUAC-IoT supports new IoT devices deployment, user revocation, and password/biometric update functionality features. We prove the security of CSUAC-IoT under the real-or-random (ROR) model, and demonstrate that it can resist several common attacks found in a typical IoT environment using the AVISPA tool. A comparative analysis also reveals that CSUAC-IoT achieves better tradeoff for security and functionality, and computational and communication costs, in comparison to five other competing approaches.**

***Index Terms*—Automated validation of Internet security protocols and application (AVISPA), Internet of Things (IoT), key agreement, security, signcryption, user access control.**

Manuscript received October 30, 2019; revised December 28, 2019; accepted January 9, 2020. Date of publication January 13, 2020; date of current version April 14, 2020. This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT, and Future Planning under Grant 2017R1A2B1002147; and in part by the Ripple Centre of Excellence Scheme, CoE in Blockchain (Sanction No. IIIT/R&D Office/Internal Projects/001/2019), IIIT Hyderabad, India. The work of Kim-Kwang Raymond Choo was supported by the National Science Foundation (NSF) Centers of Research Excellence in Science and Technology (CREST) under Grant HRD-1736209. *(Corresponding author: Kim-Kwang Raymond Choo.)*

Shobhan Mandal is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India, and also with Huawei Technologies, Bengaluru, India (e-mail: shobhan.mandal@students.iiit.ac.in).

Basudeb Bera and Ashok Kumar Das are with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India (e-mail: basudeb.bera@research.iiit.ac.in; iitkgp.akdas@gmail.com).

Anil Kumar Sutrala is with CA Technologies—A Broadcom Company, Hyderabad 500 032, India (e-mail: anilkumarsutrala@gmail.com).

Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: raymond.choo@fulbrightmail.org).

Youngho Park is with the School of Electronics Engineering, Kyungpook National University, Daegu 41566, South Korea (e-mail: parkyh@knu.ac.kr).

Digital Object Identifier 10.1109/JIOT.2020.2966242

I. Introduction

IN AN Internet of Things (IoT) environment, Internet-connected devices are increasingly *smart* in the sense that most actions (e.g., collecting environmental data, which are then sent to the edge or cloud servers) are undertaken with minimal human intervention. These things are also called IoT devices and smart devices/objects, and can be either physical or virtual. Examples of such devices include cameras, sensors, smartphones, unmanned ground vehicles, and unmanned aerial vehicles (also referred to as drones). In a typical IoT environment, several smart devices are installed or deployed in a certain deployment area (e.g., smart home, smart city, and hospitals) that can sense relevant information from the surrounding area, and the sensed information is then disseminated to their respective gateway nodes (GNs). The smart devices are assigned with their unique identities, such as device ID or IP address. In recent times, there has been a trend to adopt IPv6 over low-power wireless personal area networks (6LoWPANs) in IoT settings [1], in order to deal with the increasing scale of deployment. For example, a study by Gartner Inc. [2] predicted that the number of connected IoT smart devices will be close to 20.4 billion and hardware spending from both cross-industry and vertical-specific IoT devices will be $3 trillion by the year 2020.

There are a number of challenges in setting up an IoT environment, and one particular example is security. For example, communications among various users, smart devices, and GNs typically take place over insecure channels. In other words, there is a risk that the communications can be intercepted, hijacked, deleted, modified (e.g., inserting fabricated messages), and so on. This necessitates the design and implementation of the secure and efficient user access control mechanism in the IoT system, in order to ensure that only authorized registered users are allowed access to the relevant information and/or services. As we will point out later in Section III, designing secure and efficient user access control solutions remain challenging.

In this article, we design a new three-factor certificateless-signcryption-based user access control scheme for an IoT environment (hereafter referred to as CSUAC-IoT). CSUAC-IoT permits a legitimate registered user U to access real-time data/services from a designated IoT smart device S_i, provided that mutual authentication is successful. The mutual authentication is carried out via the trusted GN. The session key established after mutual authentication is

successfully completed will then be used to secure subsequent communications between U and S_i. CSUAC-IoT is also designed to support other functionality features, such as dynamic smart device addition (i.e., a new smart device can be easily added in the existing IoT environment), user revocation (i.e., access can be easily revoked for a legitimate registered user U), and user password/biometric update (which can be performed locally by a registered user U using his/her mobile device at any time any place).

In the next two sections, we will briefly revisit the background materials and related literature. The proposed CSUAC-IoT scheme is presented in Section IV. The security evaluations of CSUAC-IoT are presented in both Sections V and VI. A comparative analysis in terms of demonstrating communication and computational costs, security, and functionality attributes is then presented in Section VII. The conclusion is presented in the last section.

II. Background

In this section, we will introduce the network and threat models used in this article.

Fig. 1 is an example network model in the IoT [3], where a (large) number of IoT smart devices (e.g., sensors) are installed/placed in some group(s). We will refer to such groups as *cells*, where each cell has a GN for a particular IoT application. Depending on the actual IoT applications, the number and types of IoT devices may differ significantly [e.g., a smart grid setting will likely differ from an Internet of Battlefield Things (IoBT) setting]. The trusted authority (TA) is responsible for enrollment of IoT devices and the gateways, and also for registering any users. The sensed data can then be securely disseminated to some trusted/secure servers (e.g., cloud servers) for data analysis through the GNs, and the (raw/processed) data can also be made available to other authorized users. In other words, the setting we consider in this article is where a legitimate registered user (U) will be permitted to access data/service directly from the designated IoT smart devices (S_i) after mutual authentication has been successfully executed between the relevant entities. This will result in the session key establishment between the mutually authenticated entities (U and S_i) via the GN.

We use the Dolev–Yao (DY) threat model [4], where the adversary $\mathcal{A}$ can intercept messages between any communicating entities (in our context, U, GN, and S_i), and also modify, delete, or insert forged messages. The DY model is widely used in the security literature, as noted in a recent survey [5]. The popular Canetti and Krawczyk (CK-adversary) model [6] is also considered in this article. Specifically, a CK-adversary builds on the capabilities of a DY model adversary, where the CK-adversary can also compromise secret credentials, secret keys as well as session states if these are insecurely stored in the memory of the devices during the access control process [6]. Furthermore, the mobile device MD_U of a registered user U can be stolen or lost, and thus, all sensitive credentials can be extracted from the memory of MD_U by the adversary $\mathcal{A}$, for example, using power analysis attacks [7] or mobile/IoT forensics [8]–[10]. These extracted credentials can be further utilized by $\mathcal{A}$ to mount other potential attacks, including impersonation and offline guessing attacks. In addition, it is possible that some IoT smart devices can be physically captured by $\mathcal{A}$ as the devices are installed/placed in the unattended/hostile environment.

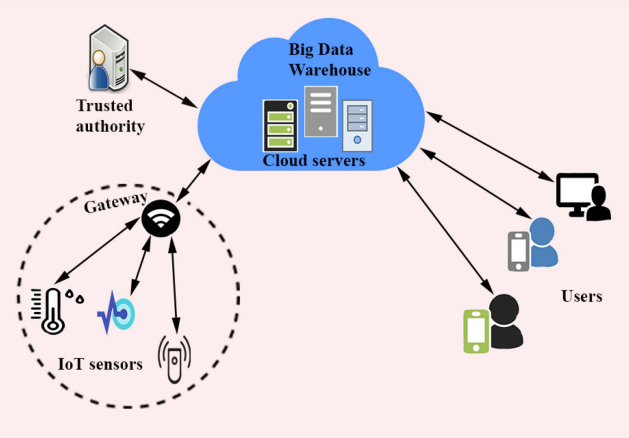

Fig. 1. Network model for IoT environment [3].

III. Related Literature

In this section, we will revisit a number of related user access control mechanisms, designed for both wireless sensor networks (WSNs) and IoT environment (see Table I). For example, in 2016, Kumar *et al.* [11] presented an access control mechanism for WSNs, which supports node or identity privacy. Their mechanism is based on elliptic curve cryptographic techniques. In their scheme, a sensing node authenticates a coordinator node in WSNs, and then, after mutual authentication, both sensing node and coordinator node agree on a common session key that can be used to facilitate future secure communication. Also in the same year, Li *et al.* [12] designed an identity-based access control scheme that employs heterogeneous signcryption. However, the scheme relies on computationally expensive computation operations such as bilinear pairings. In 2017, Li *et al.* [13] also designed another access control solution for authorizing and authenticating a registered user requesting to access real-time information. Their solution also allows a registered user to be revoked. The solution utilizes certificateless signcryption, and provides both public verifiability and ciphertext authenticity. Another scheme that uses certificateless cryptography is that of Luo *et al.* [14]. However, these schemes [12]–[14] are not practical due to the computation overheads.

In 2018, Xue *et al.* [15] designed an access control scheme for smart home systems. Their scheme provides authentication, secure data access, and integrated storage facility. However, the scheme does not support key agreement, anonymity, or untraceability preservation. Moreover, the scheme is also not shown to be secure against ephemeral secret leakage (ESL) attacks under the CK-adversary model.

In 2019, Zeng *et al.* [16] presented an anonymous user authentication (E-AUA) scheme for both users and servers in a multiserver environment. E-AUA utilizes multiple servers in

TABLE I
LIMITATIONS/DRAWBACKS OF EXISTING ACCESS CONTROL SCHEMES DESIGNED FOR THE IOT ENVIRONMENT: A SNAPSHOT

Scheme	Limitations / Drawbacks
Li *et al.* [12]	* Uses computationally expensive computation operations such as bilinear pairings.
Li *et al.* [13]	* Uses computationally expensive computation operations such as bilinear pairings.
Luo *et al.* [14]	* Uses computationally expensive computation operations such as bilinear pairings.
Xue *et al.* [15]	* Does not support key agreement facility, and anonymity & untraceability preservation properties. * The scheme is not sucure against ephemeral secret leakage (ESL) attack under the CK-adversary model.
Zeng *et al.* [16]	* The scheme is computationally expensive due to bilinear pairings operations. * It is vulnerable to offline password guessing attack through the privileged-insider attack. * In addition, an adversary can also derive the server's private key.

order to handle network congestion in a typical mobile IoT environment. E-AUA is, however, computationally expensive as it relies on bilinear pairings operations. In addition, in E-AUA, during the user registration procedure, a user U_i submits the registration message $\{\mathrm{ID}_{U_i}, h_0(\mathrm{ID}_{U_i}, \mathrm{PW}_{U_i}, b_{U_i})\}$ to the registration center (RC), where ID_{U_i} is the identity, PW_{U_i} is the password, and b_{U_i} is the random secret picked by U_i. Now, let us assume that a privileged insider associated with the *RC* is the adversary. The latter knows this information and the extracted information $\{\psi_{U_i}, v_{U_i}, b_{U_i}\}$ from user U_i's lost/stolen smart gateway, where the server's private key sk_A, $\psi_{U_i} = sk_A \oplus h_0(\mathrm{ID}_{U_i}, \mathrm{PW}_{U_i}, b_{U_i})$ and $v_{U_i} = h_1(h_0(\mathrm{ID}_{U_i}, \mathrm{PW}_{U_i}, b_{U_i}), \psi_{U_i})$, and $h_0(\cdot)$ and $h_1(\cdot)$ are cryptographic one-way hash functions. Thus, the adversary can easily guess a password, say PW'_{U_i}, and checks if $h_0(\mathrm{ID}_{U_i}, \mathrm{PW}'_{U_i}, b_{U_i}) = h_0(\mathrm{ID}_{U_i}, \mathrm{PW}_{U_i}, b_{U_i})$. If the condition is valid, the adversary will have correctly guess U_i's password PW_{U_i}. Hence, the scheme is vulnerable to offline password guessing attack involving a privileged insider. Moreover, the adversary will be able to derive the server's private key sk_A using ψ_{U_i} as $sk_A = \psi_{U_i} \oplus h_0(\mathrm{ID}_{U_i}, \mathrm{PW}'_{U_i}, b_{U_i})$.

TABLE II
SUMMARY OF NOTATIONS

Symbol	Remark
TA	Trusted authority
S_i, ID_{S_i}	i^{th} IoT smart device and its identity
GN, ID_{GN}	A gateway node associated with a cell and its identity
U, ID_U	A user and his/her identity
Pw_U, Bio_U	Password and bioemtrics of U, respectively
MD_U	Mobile device of the user U
q	A sufficient large prime
$E_q(a,b)$	A "non-singular elliptic curve: $y^2 = x^3 + ax + b \pmod q$ over a prime finite field $GF(q)$ such that $4a^3 + 27b^2 \neq 0 \pmod q$"
G	A base point in $E_q(a,b)$ of order n
$P+Q$	"Elliptic curve point addition"; $P, Q \in E_q(a,b)$
$k.P$	"Elliptic curve scalar (point) multiplication"; $k \in Z_q^*$, $P \in E_q(a,b)$
$x * y$	Ordinary modular multiplication of x and y that are in Z_q^*
(x_T, Y_T)	Private-public key pair of the TA
(x_G, Y_G)	Distinct private-public key pair of a gateway node GN
(x_{S_i}, Y_{S_i})	Distinct private-public key pair of a smart device S_i
Z_i	Secret key shared between a smart device S_i and its GN
(x_U, Y_U)	Distinct private-public key pair of a user U
(k_U, L_U)	U's partial private key and corresponding public key
ΔT	"Maximum transmission delay" related to a message
$H(\cdot)$	"Collision-resistant cryptographic one-way hash function"
$KH(\cdot)$	"Collision-resistant one-way keyed hash function"
$Gen(\cdot)$, $Rep(\cdot)$	"Fuzzy extractor probabilistic generation and deterministic reproduction" functions, respectively
et	"Error tolerance threshold value used in $Rep(\cdot)$
SK_{US_i}	Established session key between U and S_i
$\|\|$, $\oplus$	"Concatenation" & "bitwise XOR" operations

IV. OUR PROPOSED CSUAC-IOT SCHEME

Our proposed CSUAC-IoT scheme comprises a number of entities, namely, some TA, GNs, users U, and IoT devices S_i. As discussed previously, we assume that a set of IoT devices will be connected with a GN that will form a cell. The GN in a cell can connect with other GNs and also with the TA and cloud servers. The users can access real-time data from the designated IoT devices in a particular cell, as long as the users are registered with the respective TA prior to accessing the data.

A summary of notations is described in Table II. We remark that the current timestamps of various entities (GN, U, and S_i) are used to prevent replay attacks, typical of other approaches presented in (e.g., [17]–[19]).

There are also a number of phases in our proposed scheme, namely, system initialization (see Section IV-A), enrollment (see Section IV-B), user registration (see Section IV-C), login and access control (see Section IV-D), dynamic device addition (see Section IV-E), user revocation (see Section IV-F), and user password/biometrics update (see Section IV-G).

A. System Initialization Phase

The *TA* is responsible for setting up network parameters, as follows.

1) A finite field $GF(q)$ is randomly chosen, where q is a large odd prime (e.g., q of at least 160-bit prime should be used, so that the elliptic curve discrete logarithm problem (ECDLP) is intractable).
2) Next, a nonsingular elliptic curve $E_q(a,b) : y^2 = x^3 + ax + b \pmod q$ is chosen, such that $a, b \in Z_q = \{0, 1, \ldots, q-1\}$ and $4a^3 + 27b^2 \neq 0 \pmod q$.
3) A base point G in $E_q(a,b)$ is also selected, whose order is n and n is as large as q.
4) A random private key $x_T \in \mathbb{Z}_n^*$ is chosen, and the respective public key $Y_T = x_T.G$ is computed.
5) Finally, the parameters $\{G, E_q(a,b), Y_T\}$ are revealed publicly, and TA's private key x_T is only known to the TA.

B. Enrollment Phase

In this phase, the GN and IoT device (S_i) are enrolled with the TA, as follows.

1) When registering a GN, the TA assigns a unique identity $\mathrm{ID_{GN}}$ and a distinct random *private key* x_G, and calculates its corresponding *public key* as $Y_G = x_G.G$. Next, the TA stores its assigned identity $\mathrm{ID_{GN}}$ along with x_G and Y_G in the memory of GN prior to its placement in the IoT environment. The TA declares Y_G publicly.

2) A group of IoT devices are assigned to one GN, and collectively referred to as a *cell*. To register a single device S_i in a particular cell associated with a particular GN, the TA assigns a distinct random private key x_{S_i} and derives the respective public key $Y_{S_i} = x_{S_i}.G$, and also assigns a unique identity ID_{S_i} for S_i. Then, $\{\mathrm{ID}_{S_i}, x_{S_i}\}$ is loaded into S_i's memory before its deployment. Note that Y_{S_i} is declared publicly by the TA.
3) The TA then loads the parameters $\{\mathrm{ID}_{S_i}, Y_{S_i}\}$ of S_i with the respective GN. In addition, a distinct random secret key Z_i is chosen by the TA, which is to facilitate future secure communication between S_i and its corresponding (GN). The TA then places the secret key Z_i into the respective smart device S_i and the GN.
4) The TA securely deletes the generated private keys x_{S_i} for each smart device S_i and x_G for each GN, and the secret key Z_i from its memory.

C. User Registration Phase

A user U registers with the TA, as follows.

1) *Set Secret Value and Public Value:* U first picks an identity ID_U and chooses a random number $x_U \in \mathbb{Z}_n^*$ as a secret value. Next, using this secret value, U calculates the respective public value as $Y_U = x_U.G$.
2) *Partial-Private-Key-Extract:* U submits ID_U and its public value Y_U to the TA securely. The TA calculates the partial private key d_U of U as follows:

$$\mathrm{temp}_{d_U} = H(Y_U||\mathrm{ID}_U||x_T).G = \left(\mathrm{temp}_{d_{Ux}}, \mathrm{temp}_{d_{Uy}}\right)$$
$$\mathrm{verify}_{d_U} = \mathrm{temp}_{d_{Ux}} \oplus \mathrm{temp}_{d_{Uy}}$$
$$d_U = \mathrm{verify}_{d_U} \oplus \mathrm{YYYYMMDD}$$

where YYYYMMDD represents the date of expiration of the public key of user U, and YYYY, MM, and DD are the year, month, and day, respectively. The TA then securely sends d_U to user U, and also sends $verify_{d_U}$ to the GNs with whom user U will establish a connection with the designated smart devices S_i. The TA also stores d_U, Y_U, and ID_U for its further use.
3) *Set Private Key and Public Key:* With the received partial private key d_U securely from the TA, U continues to calculate private key k_U and its respective public key L_U using the secret value x_U and public value Y_U as $k_U = x_U * d_U$ and $L_U = d_U.Y_U$, respectively, where $x*y$ denotes an ordinary modular multiplication in Z_q^*.

Finally, the TA sends the information $\{E_q(a,b), H(\cdot), KH(\cdot), G, x_U, Y_U, k_U, L_U\}$ and a list of identities of IoT smart devices $\{\mathrm{ID}_{S_i}\}$ securely to user U.

After receiving the information securely from the TA, U picks a password Pw_U and imprints his/her biometrics Bio_U at the sensor of the mobile device MD_U. After that MD_U generates the biometric secret key σ_U and public reproduction parameter τ_U corresponding to input Bio_U using the widely used "fuzzy extractor probabilistic generation function $Gen(\cdot)$" [20] as $Gen(Bio_U) = (\sigma_U, \tau_U)$, and calculates $x_U^* = x_U \oplus H(\mathrm{ID}_U||Pw_U||\sigma_U)$, $k_U^* = k_U \oplus H(\sigma_U||\mathrm{ID}_U||Pw_U)$, and $HPB_U = H(Pw_U||L_U||\sigma_U||\mathrm{ID}_U)$. U then stores the information $\{E_q(a,b), H(\cdot), KH(\cdot), G, x_U^*, Y_U, k_U^*, L_U, HPB_U, Gen(\cdot), Rep(\cdot), \tau_U, et\}$ in his/her mobile device MD_U, where $Rep(\cdot)$ represents the"deterministic reproduction function" and et is a predefined "error tolerance threshold value." Note that given a noisy biometric, say Bio'_U of a user U, $Rep(\cdot)$ reconstructs the original biometric secret key σ_U with the help of τ_U and et provided that the hamming distance (Hdist) between registered biometrics Bio_U and current biometrics Bio'_U is less than or equal to et [20], that is, $\sigma_U = Rep(Bio'_U, \tau_U)$ and $Hdist(Bio_U, Bio'_U) \le et$.

Algorithm 1 User Signcryption

Input: $\{Y_G, G, x_U^*, k_U^*, L_U, ID_{S_i}, Y_{S_i}\}$
Output: $\{C_1, T_1, T_{2i}, S_1, S_{2i}, W_{SK}, TS_U, Sign_G, Sign_{S_i}\}$

Input password Pw_U at MD_U.
Imprint biometrics Bio'_U at the sensor of MD_U.
Calculate biometric secret key $\sigma_U = Rep(Bio'_U, \tau_U)$ provided $Hdist(Bio_U, Bio'_U) \le et$.
Calculate $HPB_U^* = H(Pw_U||L_U||\sigma_U||ID_U)$.
if $(HPB_U^* \neq HPB_U)$ **then**
  Terminate the phase immediately.
else
  Compute $x_U = x_U^* \oplus H(ID_U||Pw_U||\sigma_U)$, $k_U = k_U^* \oplus H(\sigma_U||ID_U||Pw_U)$.
  Randomly select integers $v_1, v_2, v_3 \in \mathbb{Z}_n^*$.
  Compute $K_{11} = H(v_1.G)$ shared with the GN.
  Compute $K_{21} = H(v_2.G)$ shared with the IoT smart device S_i.
  Compute $(x_1, y_1) = v_1.G + k_U.Y_G$, $(x_2, y_2) = v_2.G + k_U.Y_{S_i}$.
  Generate the cipher text C_1 using the key k_U as $C_1 = (ID_{S_i}||v_3) \oplus H(x_1||k_U.Y_G)$.
  Using *one-way keyed hash function* $KH(\cdot)$, calculate $r_1 = KH_{y_1}(C_1||K_{11}||L_U||Y_G||TS_U)$, $r_{2i} = KH_{y_2}(K_{21}||L_U||Y_{S_i})$, where TS_U is the current timestamp generated by U.
  Compute $S_1 = \frac{v_1}{r_1 + k_U} \pmod q$ and $S_{2i} = \frac{v_2}{r_{2i} + k_U} \pmod q$.
  Compute $T_1 = r_1.G$ and $T_{2i} = r_{2i}.G$
  Compute $W_{SK} = H(r_1||TS_U) \oplus H(v_3||k_U.Y_G)$.
  Compute the signatures on computed x_1 and x_2 as $Sign_G = H(C_1||T_1||T_{2i}||S_1||S_{2i}||W_{SK}||L_U||TS_U||x_1) * k_U + x_U \pmod q$, $Sign_{S_i} = H(T_{2i}||S_{2i}||L_U||x_2) * x_U + k_U \pmod q$.
  return $\{C_1, T_1, T_{2i}, S_1, S_{2i}, W_{SK}, TS_U, Sign_G, Sign_{S_i}\}$
end if

D. Login and Access Control Phase

In this phase, a registered user, say U, validates himself/herself to his registered GN and also to the accessed smart device S_i for which it belongs to the cell of GN. Once this is completed, U will be able to come to an agreement with the smart device S_i on a session key that can be used for further communications among U and S_i. This phase consists of four processes which are discussed as follows.

1) User Signcryption: In this process, user U performs the signcryption using the parameters available in his/her mobile

Algorithm 2 GN Unsigncryption

Input: Message $m_1 = \langle C_1, T_1, T_{2i}, S_1, S_{2i}, W_{SK}, TS_U, Sign_G, Sign_{S_i}\rangle$ received from the user U, public parameters and its own private key x_G

Output: $\{\overline{m_i}, TS_{GN}, ID_{S_i}, L_{\text{mod}}\}$

1: **if** TS_U retrieved from m_1 satisfies $|TS_U - TS'_U| < \Delta T$, where TS'_U is the timestamp when the unsigncryption is being performed at the GN **then**
2: Compute $temp_1 = S_1.(T_1 + L_U)$.
3: Compute $(x'_1, y'_1) = temp_1 + x_G.L_U$.
4: Verify if $Sign_G.G = H(C_1 \,||T_1\, ||T_{2i}\, ||S_1\, ||S_{2i}\, ||W_{SK}\, ||L_U\, ||TS_U\, ||x'_1).L_U + Y_U$?
5: **if** signature verification is successful **then**
6: Compute $K'_{11} = H(temp_1)$.
7: Compute $r'_1 = KH_{y'_1}(C_1||K'_{11}||L_U||Y_G||TS_U)$.
8: Compute $L_{mod} = L_U \oplus H(Z_i||TS_{GN})$, where TS_G is the current timestamp generated y the GN.
9: Extract $(ID_{S_i}||v_3) = C_1 \oplus H(x'_1||x_G.L_U)$ using decryption algorithm with the help of the key x_G.
10: Using v_3, further retrieve $H(r'_1||TS_U)' = W_{SK} \oplus H(v_3||x_G.L_U)$, which is same as $H(r_1\, ||TS_U)$.
11: **if** ID_{S_i} exists in the database of the GN **then**
12: Retrieve the symmetric key Z_i corresponding to the ID_{S_i}.
13: Compute $\overline{m_i} = H(r'_1\, ||TS_U)' \oplus H(Z_i\, ||TS_{GN}\, ||L_U)$.
14: **end if**
15: **end if**
16: **end if**
17: **return** $\{\overline{m_i}, TS_{GN}, ID_{S_i}, L_{mod}\}$

Algorithm 3 Smart Device Unsigncryption and Session Key Establishment

Input: Message $m_2 = \langle T_{2i}, S_{2i}, \overline{m_i}, L_{\text{mod}}, TS_{GN}, Sign_{S_i}\rangle$, public parameters and stored information $\{Z_i, ID_{S_i}, x_{S_i}\}$

Output: $\{VSK_{S_iU}, TS_{S_i}, SK_{S_iU}\}$

1: **if** TS_{GN} retrieved from m_2 satisfies $|TS_{GN} - TS'_{GN}| < \Delta T$, where TS'_{GN} is the timestamp when the unsigncryption is being performed at the smart device S_i **then**
2: Compute $L_U = L_{\text{mod}} \oplus H(Z_i||TS_{GN})$.
3: Compute $temp_2 = S_{2i}.(T_{2i} + L_U)$.
4: Compute $(x'_2, y'_2) = temp_2 + x_{Si}.L_U$.
5: Verify if $Sign_{S_i}.G = H(T_{2i}||S_{2i}||L_U||x'_2).Y_U + L_U$?
6: **if** signature verification is successful **then**
7: Compute $K_{21} = H(temp_2)$.
8: Compute $r_{2i} = KH_{y'_2}(K_{21}||L_U||Y_{Si})$.
9: Extract $H(r'_1\, ||TS_U)' = \overline{m_i} \oplus H(Z_i\, ||TS_{GN}\, ||L_U)$.
10: Compute the session-key SK_{S_iU} and its verifier VSK_{S_iU} as
$SK_{S_iU} = H((r_{2i}+1)||H(r'_1||TS_U)'||x'_2||y'_2)$,
$VSK_{S_iU} = H(SK_{S_iU}||TS_{S_i})$, where TS_{S_i} is the current timestamp generated by S_i.
11: **end if**
12: **end if**
13: **return** $\{VSK_{S_iU}, TS_{S_i}, SK_{S_iU}\}$

device MD_U by inputting his/her password Pw_U and imprinting biometrics Bio'_U at the sensor of the mobile device MD_U such that U can be authenticated by both the GN and the accessed smart device S_i. U selects a smart device with identity ID_{S_i} from which he/she wishes to access the real-time information in the IoT environment.

For this purpose, U uses the user signcryption in Algorithm 1. U has the access of the parameters $\{Y_G, G, Y_{S_i}, x^*_U, k^*_U, L_U, \text{ID}_{S_i}\}$. The output produced by the algorithm contains $\{C_1, T_1, T_{2i}, S_1, S_{2i}, W_{SK}, TS_U, \text{Sign}_G, \text{Sign}_{S_i}\}$. The signature Sign_G is generated for the GN, while the signature Sign_{S_i} is for S_i. Next, U constitutes the message $m_1 = \langle C_1, T_1, T_{2i}, S_1, S_{2i}, W_{SK}, TS_U, \text{Sign}_G, \text{Sign}_{S_i}\rangle$, which is sent to the intended GN via public channel.

2) Gateway Node Unsigncryption: After reception of message m_1 from user U at time TS'_U, the GN verifies the signcryption which was performed by user U. If it is valid, the GN creates a message and sends it to the accessed smart device S_i, otherwise, it terminates further processing.

To accomplish the above task, the GN executes the gateway unsigncryption provided in Algorithm 2. We observe that $L_U = d_U.Y_U = d_U.(x_U.G) = (x_U * d_U).G = k_U.G$, $\text{temp}_1 = S_1.(T_1 + L_U) = (v_1/r_1 + k_U)[r_1.G + k_U.G] = (v_1/r_1 + k_U)[(r_1 + k_U).G] = v_1.G$, and $x_G.L_U = x_G.(d_U.Y_U) = x_G.(d_U.(x_U.G)) = (x_G * d_U).(x_G.G) = k_U.Y_G$. Thus, $(x'_1, y'_1) = \text{temp}_1 + x_G.L_U = v_1.G + k_U.Y_G = (x_1, y_1)$. In this process, the signature verification condition is $\text{Sign}_G.G = H(C_1||T_1||T_{2i}||S_1||S_{2i}||W_{SK}||L_U||TS_U||x'_1).L_U + Y_U$. Note that $\text{Sign}_G.G = (H(C_1||T_1||T_{2i}||S_1||S_{2i}||W_{SK}||L_U||TS_U||x_1) * k_U).G + x_U.G = H(C_1||T_1||T_{2i}||S_1||S_{2i}||W_{SK}||L_U||TS_U||x_1).(k_U.G) + Y_U = H(C_1||T_1||T_{2i}||S_1||S_{2i}||W_{SK}||L_U||TS_U||x_1).L_U + Y_U = H(C_1||T_1||T_{2i}||S_1||S_{2i}||W_{SK}||L_U||TS_U||x'_1).L_U + Y_U$. If the value of $\overline{m_i}$ comes as *NULL*, it means that the algorithm could not verify at one of the *if* conditions. Otherwise, the GN creates a message $m_2 = \langle T_{2i}, S_{2i}, \overline{m_i}, L_{\text{mod}}, TS_{\text{GN}}, \text{Sign}_{S_i}\rangle$, which is sent to the intended smart device S_i over public channel.

3) Smart Device Unsigncryption and Session Key Establishment: Upon reception of the message $m_2 = \langle T_{2i}, S_{2i}, \overline{m_i}, L_{\text{mod}}, TS_{\text{GN}}, \text{Sign}_{S_i}\rangle$ from the respective GN of the accessed smart device S_i, S_i executes the unsigncryption and session key establishment process in Algorithm 3.

It is worth noticing that $\text{temp}_2 = S_{2i}.(T_{2i} + L_U) = (v_2/[r_{2i} + k_U])[r_{2i}.G + d_U.Y_U] = (v_2/[r_{2i} + k_U])[r_{2i}.G + d_U.(x_U.G)] = (v_2/[r_{2i} + k_U])[r_{2i}.G + (d_U * x_U).G] = (v_2/[r_{2i} + k_U])[r_{2i}.G + k_U.G] = v_2.G$ and $(x'_2, y'_2) = \text{temp}_2 + x_{Si}.L_U = v_2.G + k_U.Y_{S_i} = (x_2, y_2)$. Furthermore, $\text{Sign}_{S_i}.G = H(T_{2i}||S_{2i}||L_U||x_2).(x_U.G) + k_U.G = H(T_{2i}||S_{2i}||L_U||x'_2).Y_U + L_U$. On successful verification of the signature Sign_{S_i} by S_i in Algorithm 3, S_i proceeds to calculate the session key $SK_{S_iU} = H((r_{2i} + 1)||H(r_1||TS_U)'||x'_2||y'_2)$, and its verifier $VSK_{S_iU} = H(SK_{S_iU}||TS_{S_i})$. Next, S_i creates a message $m_3 = \langle VSK_{S_iU},\ TS_{S_i}\rangle$, which is sent to user U who requested its access, via the open channel.

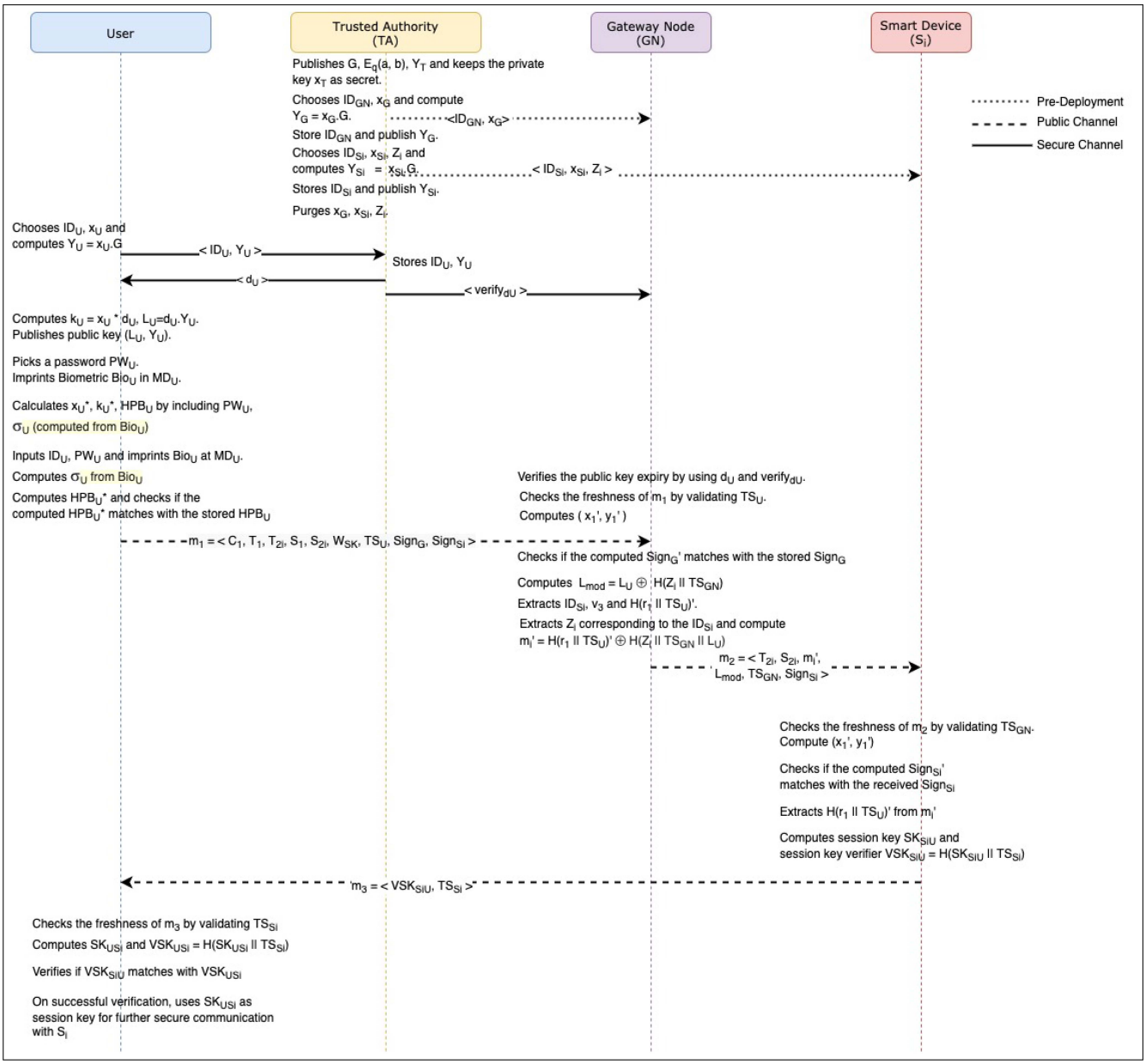

Fig. 2. Overview of the login and user access control phase.

4) Session Key Verification: Upon reception of the message $m_3 = \langle VSK_{S_iU}, TS_{S_i} \rangle$, user U executes the session key verification in Algorithm 4. Once the session key $SK_{US_i} = H((r_{2i}+1)||H(r_1||TS_U)||x_2||y_2)$ is calculated, the verification of SK_{US_i} is done through the verification condition $VSK_{US_i} = VSK_{S_iU}$. Once the condition is successful, U stores the session key SK_{US_i} for secret communication with S_i. Similarly, S_i stores the same session key $SK_{S_iU} (= SK_{US_i})$ for secret communication with U. The overall process is also briefed in Fig. 2.

E. Dynamic Smart Device Addition Phase

When a new smart device S_n needs to be added in the existing IoT environment residing in a particular cell containing the GN, the TA assigns a unique identity ID_{S_n} and a unique random private key x_{S_n} to compute the corresponding public key $Y_{S_n} = x_{S_n}.G$ for S_n. Moreover, a unique secret key Z_n is generated by the TA which is shared between S_n and the GN for encryption/decryption purpose. Next, ID_{S_n}, x_{S_n}, and Z_n are loaded by the TA into S_n's memory before its placement in the network. In addition, the information ID_{S_n} and Z_n are sent securely to the GN by the TA. The public key Y_{S_n} is declared in public by the TA. The TA also removes x_{S_n} and Z_n from its memory.

After the above process is completed, the smart device S_n is placed in the respective cell, and the intended users can start accessing the real-time data from S_n by executing the login and user access control phase discussed in Section IV-D.

F. User Revocation Phase

In this phase, the access is revoked for a legal registered user U by the TA as follows. Suppose user U is attached with

Algorithm 4 Session Key Verfication

Input: Message $m_3 = \langle VSK_{S_iU}, TS_{S_i}\rangle$, $H(r_1 \,||TS_U)$, r_{2i}, and public parameters

Output: Session key SK_{US_i} $(= SK_{S_iU})$ shared between U and S_i

1: **if** TS_{S_i} retrieved from m_3 satisfies $|TS_{S_i} - TS'_{S_i}| < \Delta T$, where TS'_{S_i} is the timestamp when the verification process is done by U **then**
2: Using precalculated r_{2i}, $H(r_1||TS_U)$, x_2 and y_2 during the signcryption process, compute $SK_{US_i} = H((r_{2i}+1) \,||H(r_1 \,||TS_U) \,||x_2 \,||y_2)$.
3: Using SK_{US_i} and TS_{S_i}, compute $VSK_{US_i} = H(SK_{US_i}||TS_{S_i})$.
4: **if** $VSK_{US_i} = VSK_{S_iU}$ **then**
5: Session key SK_{US_i} $(= SK_{S_iU})$ established between U and S_i is verified successfully, and store SK_{US_i} for secret communication with S_i.
6: **end if**
7: **end if**

the smart devices (S_i). The TA first computes

$$\text{temp}_{d_U} = H(Y_U||\text{ID}_U||x_T).G = \left(\text{temp}_{d_{Ux}}, \text{temp}_{d_{Uy}}\right)$$
$$\text{verify}_{d_U} = \text{temp}_{d_{Ux}} \oplus \text{temp}_{d_{Uy}}$$

using the public key Y_U and identity ID_U of user U, and its own private key x_T. Next, the TA applies its computed $verify_{d_U}$ and its stored d_U associated with U in order to retrieve the U's date of expiration as

$$\text{YYYYMMDD} = \text{verify}_{d_U} \oplus d_U.$$

By checking this against the current date, the TA can easily verify if the U's registration is still valid or not. In case, U has to be revoked before the date of expiration, the TA sends a list of users to the GN containing such blacklisted users who need to be cross-referenced by the GN before their messages are forwarded to the GN's cell.

G. User Password/Biometrics Update Phase

This phase permits a registered user U to update his/her old password and/or biometrics information with the new password/biometrics if he/she wishes to do so for achieving the maximum security in the system. This phase is performed by U completely locally without further contacting the TA at any time and any place with the help of MD_U.

The following steps are needed to execute the desired task.

Step 1: U inputs old password Pw_U^{old} and new updated password Pw_U^{new} at MD_U. In addition, U also imprints old biometrics Bio_U^{old} and new biometrics Bio_U^{new} at the sensor of MD_U. At this point, it is worth noticing that biometrics of the user U are not generally changed over time. However, if U wishes to update old biometrics by new one, he/she is permitted to so. In case, old biometrics Bio_U^{old} is not updated, Bio_U^{new} will be considered as Bio_U^{old}.

Step 2: U calculates biometric secret key $\sigma_U^{\text{old}} = Rep(Bio_U^{\text{old}}, \tau_U)$ provided $Hdist(Bio_U, Bio_U^{\text{old}}) \leq et$. Next, U calculates $HPB_U^{\text{old}} = H(Pw_U^{\text{old}}||L_U||\sigma_U^{\text{old}}||\text{ID}_U)$ and if $HPB_U^{\text{old}} = HPB_U$, U proceeds to the next step. Otherwise, this phase is ended here.

Step 3: U calculates $x_U = x_U^* \oplus H(\text{ID}_U||Pw_U^{\text{old}}||\sigma_U^{\text{old}})$, $k_U = k_U^* \oplus H(\sigma_U^{\text{old}}||\text{ID}_U||Pw_U^{\text{old}})$, $Gen(Bio_U^{\text{new}}) = (\sigma_U^{\text{new}}, \tau_U^{\text{new}})$, $x_U^{\text{new}} = x_U \oplus H(\text{ID}_U||Pw_U^{\text{new}}||\sigma_U^{\text{new}})$, $k_U^{\text{new}} = k_U \oplus H(\sigma_U^{\text{new}}||\text{ID}_U||Pw_U^{\text{new}})$, and $HPB_U^{\text{new}} = H(Pw_U^{\text{new}}||L_U||\sigma_U^{\text{new}}||\text{ID}_U)$. Finally, U replaces x_U^*, HPB_U, k_U^*, and τ_U by $x_U^{\text{new}}, HPB_U^{\text{new}}, k_U^{\text{new}}$, and τ_U^{new}, respectively. At the end of this phase, U's mobile device MD_U holds $\{E_q(a,b), H(\cdot), KH(\cdot), G, x_U^{\text{new}}, Y_U, k_U^{\text{new}}, L_U, HPB_U^{\text{new}}, Gen(\cdot), Rep(\cdot), \tau_U^{\text{new}}, et\}$.

V. Security Analysis

Both formal and informal (nonmathematical) security analysis have been carried out for the proposed scheme to show its resilience against various attacks needed in an IoT environment. In the following, we consider two computational problems that are useful in security analysis.

Definition 1: A "collision-resistant cryptographic one-way hash function" $H : \{0,1\}^* \rightarrow \{0,1\}^{l_h}$ is as a "deterministic function that takes an arbitrary length string and outputs a fixed length string of l_h bits." If $Adv_{\mathcal{A}}^{\text{Hash}}(rt)$ is the "advantage of an adversary $\mathcal{A}$ in finding a hash collision in $H(\cdot)$," $Adv_{\mathcal{A}}^{\text{Hash}}(rt) = Pr[(is_1, is_2) \in_r \mathcal{A} : is_1 \neq is_2, H(is_1) = H(is_2)]$, where $Pr[E]$ denotes the "probability of a random event E," and the input strings pair $(is_1, is_2) \in_r \mathcal{A}$ indicates that the strings is_1 and is_2 are randomly chosen by $\mathcal{A}$. An (η, rt)-adversary $\mathcal{A}$ attacking the "collision resistance of $H(\cdot)$" is defined as follows: $Adv_{\mathcal{A}}^{\text{Hash}}(rt) \leq \eta$ and $\mathcal{A}$'s runtime will be at most rt.

Definition 2: Suppose $E_q(a,b)$ is a nonsingular elliptic curve over a finite field $GF(q) : y^2 = x^3 + ax + b \pmod q$ and $G \in E_q(a,b)$ is a point. The elliptic curve decisional Diffie–Hellman problem (ECDDHP) is that given a quadruple $(G, u.G, v.G, w.G)$, derive whether $w = u * v$ or a "uniform random value," where $u, v, w \in Z_q^*$ and $Z_q^* = \{1, 2, \ldots, q-1\}$.

A. Formal Security Under ROR Model

The broadly applied real-or-random (ROR) model [21] has been considered for the formal security analysis of the proposed scheme. Under the ROR model, an active adversary, say $\mathcal{A}$ has access to various queries (also known as random oracles) which are tabulated in Table III. Recently, the ROR model-based formal security analysis becomes popular in the research community in proving the semantic security of an authentication key agreement (AKE) scheme [17]–[19], [22]–[24]. For detailed treatment on the ROR model, the readers can refer to [21].

The following components are associated with the ROR model.

Participants: Various participants involved in the proposed scheme: 1) user (U); 2) GN; and 3) smart device (S_i). The instances j_1, j_2, and j_3 of U, GN, and S_i are considered as $\Pi_U^{j_1}$, $\Pi_{\text{GN}}^{j_2}$, and $\Pi_{S_i}^{j_3}$, which are known as "random oracles."

Accepted State: An instance Π^j is in the "accepted state," if it switches to an accept state after reception of the last authorized protocol message. If we arrange all the sent and

TABLE III
VARIOUS QUERIES AND THEIR GOALS

Query	Purpose
$Send(\Pi^j, Msg)$	This query permits $\mathcal{A}$ to dispatch a message Msg to Π^j, and Π^j also receives a response for the message Msg
$Execute(\Pi_U^{j_1}, \Pi_{GN}^{j_2}, \Pi_{S_i}^{j_3})$	It permits $\mathcal{A}$ to intercept the messages communicated among U, GN and S_i
$CorruptMDevice(\Pi_U^{j_1})$	This query allows $\mathcal{A}$ to get a registered user U's password Pw_U and biometric secret key σ_U from "U's stolen or lost mobile device MD_U"
$Reveal(\Pi^j)$	Using this query, the current session key SK_{US_i} $(= SK_{S_iU})$ between Π^j and its corresponding partner is revealed to the adversary $\mathcal{A}$
$Test(\Pi^j)$	This query permits $\mathcal{A}$ to appeal Π^j for SK_{US_i} $(= SK_{S_iU})$ and Π^j gives a "random outcome of a flipped unbiased coin, say c"

received messages in sequence, the "session identification *sid* of Π^j for the current session" is created.

Partnering: Two instances (Π^{j_1} and Π^{j_2}) are called the partners to each other, if the following three settings are satisfied.

1) Π^{j_1} and Π^{j_2} will be in accepted states.
2) Π^{j_1} and Π^{j_2} will have the same *sid* and they will "mutually authenticate each other."
3) Π^{j_1} and Π^{j_2} will be "mutual partners of each other."

Freshness: $\Pi_U^{j_1}$ or $\Pi_{S_i}^{j_3}$ is said to be fresh if the generated session key $SK_{US_i}(= SK_{S_iU})$ between U and S_i is not known to $\mathcal{A}$ with the help of the Reveal(Π^j) query defined in Table III.

The "semantic security" of the proposed scheme is defined in Definition 3 and also proved in Theorem 1.

Definition 3 (Semantic Security): Let $Adv_{\mathcal{A}}^{\text{CSUAC-IoT}}(t)$ be the "advantage of an adversary $\mathcal{A}$ running in polynomial time t in breaking the semantic security of the proposed CSUAC-IoT to derive the session key $SK_{US_i}(= SK_{S_iU})$ between a user U and an IoT smart device S_i. Then, $Adv_{\mathcal{A}}^{\text{CSUAC-IoT}}(t) = |2Pr[c' = c] - 1|$, where c and c' are the "correct" and "guessed" bits, respectively.

Theorem 1: If there exists an adversary $\mathcal{A}$ running in polynomial time *pt* to derive the established session key $SK_{US_i}(= SK_{S_iU})$ between a user U and an IoT smart device S_i in the proposed scheme, CSUAC-IoT, and q_h, q_s, |Hash|, l_b, and $Adv_{\mathcal{A}}^{\text{ECDDHP}}(t)$ represent "the number of hash queries, the number of *Send* queries, the range space of a one-way collision-resistant hash function $H(\cdot)$, the number of bits in biometrics secret key σ_U, and the advantage of breaking the ECDDHP," respectively, then

$$\text{Adv}_{\mathcal{A}}^{\text{CSUAC-IoT}}(pt) \le \frac{q_h^2}{|\text{Hash}|} + 2\Big(\max\Big\{C'.q_s^{s'}, \frac{q_s}{2^{l_b}}\Big\} + \text{Adv}_{\mathcal{A}}^{\text{ECDDHP}}(pt)\Big)$$

where the details of the Zipf's parameters C' and s' are provided in [25].

Proof: We adopt the similar proof of this theorem as that is also presented in other recently designed AKE protocols [19], [22]–[24], [26]. ■

In the proposed CSUAC-IoT, we have used the user-chosen passwords. For this purpose, we adopt the Zipf's law suggested by Wang *et al.* [25], which is remarkably different from the "uniform distribution for user-chosen passwords." Typically, the size of the password dictionary is much constrained in the perception that the entire space of passwords is not fully utilized by the users, and it is rather a small space of the permissible characters space [25]. Zipf's law is also applied in other existing AKE protocols in recent years [18], [27].

We define the following games, say $G_j^{\mathcal{A}}$ for the adversary $\mathcal{A}$, $j = 0, 1, 2, 3$, whose detailed descriptions are provided as follows. If $Succ_{G_j}^{\mathcal{A}}$ is "an event wherein $\mathcal{A}$ can guess the random bit c in the game $G_j^{\mathcal{A}}$ correctly," then $\mathcal{A}$'s advantage in winning the game $G_j^{\mathcal{A}}$ in the proposed CSUAC-IoT becomes $Adv_{\mathcal{A},G_j}^{\text{CSUAC-IoT}} = Pr[Succ_{G_j}^{\mathcal{A}}]$.

Game $\mathbf{G_0^{\mathcal{A}}}$*:* This game corresponds to the "actual attack" that is performed by the adversary $\mathcal{A}$ against the proposed CSUAC-IoT with respect to the ROR model. Since the bit c is selected randomly by $\mathcal{A}$ prior to starting of the game $G_0^{\mathcal{A}}$, it follows from the semantic security defined in Definition 3 that:

$$\text{Adv}_{\mathcal{A}}^{\text{CSUAC-IoT}}(pt) = \Big|2\text{Adv}_{\mathcal{A},G_0}^{\text{CSUAC-IoT}} - 1\Big|. \quad (1)$$

Game $\mathbf{G_1^{\mathcal{A}}}$*:* Under this game, an eavesdropping attack has been performed by the adversary $\mathcal{A}$ that utilizes the *Execute* query shown in Table III. With the help of the *Execute*($\Pi_U^{j_1}$, $\Pi_{\text{GN}}^{j_2}$, $\Pi_{S_i}^{j_3}$) query, assume that $\mathcal{A}$ intercepts all the communicated messages among the entities U, GN, and S_i, which are $m_1 = \langle C_1, T_1, T_{2i}, S_1, S_{2i}, W_{SK}, TS_U, \text{Sign}_G, \text{Sign}_{S_i}\rangle$, $m_2 = \langle T_{2i}, S_{2i}, \overline{m_i}, L_{\text{mod}}, TS_{\text{GN}}, \text{Sign}_{S_i}\rangle$, and $m_3 = \langle VSK_{S_iU}, TS_{S_i}\rangle$ during the "login and access control phase." After this game is ended, $\mathcal{A}$ needs to execute the *Reveal* and *Test* queries shown in Table III to validate whether the derived session key $SK_{US_i} = H((r_{2i}+1)||H(r_1||TS_U)||x_2||y_2) = H((r_{2i}+1)||H(r_1||TS_U)'||x_2'||y_2') = SK_{S_iU}$ is an actual key or just a random number. It is clear that only eavesdropping of the messages $m_l (l = 1, 2, 3)$ does not at all increase the probability in calculating the session key $SK_{US_i}(= SK_{S_iU})$. Since both games $G_0^{\mathcal{A}}$ and $G_1^{\mathcal{A}}$ are indistinguishable, we have

$$\text{Adv}_{\mathcal{A},G_1}^{\text{CSUAC-IoT}} = \text{Adv}_{\mathcal{A},G_0}^{\text{CSUAC-IoT}}. \quad (2)$$

Game $\mathbf{G_2^{\mathcal{A}}}$*:* This game corresponds to an active attack, wherein we include the simulation of *Send* and Hash queries. Now, the eavesdropped messages m_l $(l = 1, 2, 3)$ among the participants U, GN, and S_i do not lead to any hash collision because the components attached in these messages are safeguarded by the "collision-resistant one-way hash function $H(\cdot)$ (defined in Definition 1) and elliptic curve points, and also various generated random numbers." To derive r_1 and r_{2i} from the respective T_1 and T_{2i} in m_1 and m_2, the adversary $\mathcal{A}$ needs to solve the "ECDLP." Moreover, to calculate r_1 and r_{2i} from S_1 and S_{2i} attached in m_1 and m_2, $\mathcal{A}$ needs to solve

the ECDDHP (defined in Definition 2). It is worth observing that both games $G_1^{\mathcal{A}}$ and $G_2^{\mathcal{A}}$ are indistinguishable except the inclusion of the simulation of the *Send* and Hash queries, and solving the ECDDHP. The birthday paradox result and $Adv_{\mathcal{A}}^{\text{ECDDHP}}(et)$ in solving the ECDDHP give the following:

$$\left|\text{Adv}_{\mathcal{A},G_1}^{\text{CSUAC-IoT}} - \text{Adv}_{\mathcal{A},G_2}^{\text{CSUAC-IoT}}\right| \leq \frac{q_h^2}{2|\text{Hash}|} + \text{Adv}_{\mathcal{A}}^{\text{ECDDHP}}(pt). \quad (3)$$

Game $\mathbf{G_3^{\mathcal{A}}}$*:* This game includes the *CorruptMDevice* query as described in Table III. The adversary $\mathcal{A}$ can obtain the credentials $\{E_q(a,b),\ H(\cdot),\ KH(\cdot),\ G, x_U^*,\ Y_U,\ k_U^*,\ L_U,\ HPB_U,\ Gen(\cdot),\ Rep(\cdot),\ et\}$ through execution of the *CorruptMDevice* query. To derive the private key x_U and partial private key of k_U of a registered user U, $\mathcal{A}$ requires to guess the password Pw_U and biometrics secret key σ_U concurrently. However, $\mathcal{A}$'s probability of guessing the correct σ_U is roughly $(1/2^{l_b})$ [28]. Since $\mathcal{A}$ can apply the Zipf's law on user-selected passwords [25] for guessing the passwords, using the "trawling guessing attacks," $\mathcal{A}$'s advantage of $\mathcal{A}$ becomes $(1/2)$ in the case when $q_s = 10^7$ or 10^8 [25]. For the "targeted guessing attacks where $\mathcal{A}$ can use the target user's personal information," $\mathcal{A}$'s advantage becomes over $(1/2)$ if $q_s \leq 10^6$ [25]. In a typical application, several attempts of wrong password efforts are restricted in the system, say q_s *Send* queries are permitted. If we ignore the guessing attacks for both user U's password and biometrics, the games $G_2^{\mathcal{A}}$ and $G_3^{\mathcal{A}}$ become indistinguishable. Therefore, we obtain the following relationship:

$$\left|\text{Adv}_{\mathcal{A},G_2}^{\text{CSUAC-IoT}} - \text{Adv}_{\mathcal{A},G_3}^{\text{CSUAC-IoT}}\right| \leq \max\left\{C'.q_s^{s'}, \frac{q_s}{2^{l_b}}\right\}. \quad (4)$$

Now, all the queries are made by the adversary $\mathcal{A}$ and it is only remaining for $\mathcal{A}$ to guess a bit c to win the game $G_3^{\mathcal{A}}$. It then follows that:

$$\text{Adv}_{\mathcal{A},G_3}^{\text{CSUAC-IoT}} = \frac{1}{2}. \quad (5)$$

Next, (1), (2), and (5) together produce the following relationship:

$$\begin{aligned}\frac{1}{2}\cdot Adv_{\mathcal{A}}^{\text{CSUAC-IoT}}(pt) &= |Adv_{\mathcal{A},G_0}^{\text{CSUAC-IoT}} - \frac{1}{2}| \\ &= \left|\text{Adv}_{\mathcal{A},G_1}^{\text{CSUAC-IoT}} - \frac{1}{2}\right| \\ &= \left|\text{Adv}_{\mathcal{A},G_1}^{\text{CSUAC-IoT}} - \text{Adv}_{\mathcal{A},G_3}^{\text{CSUAC-IoT}}\right|. \end{aligned} \quad (6)$$

After that, if we apply the "triangular inequality" and also (3), (4), and (6), we arrive at the following relationship:

$$\begin{aligned}&(1/2)\cdot \text{Adv}_{\mathcal{A}}^{\text{CSUAC-IoT}}(pt) \\ &= \left|\text{Adv}_{\mathcal{A},G_1}^{\text{CSUAC-IoT}} - \text{Adv}_{\mathcal{A},G_3}^{\text{CSUAC-IoT}}\right| \\ &\leq \left|\text{Adv}_{\mathcal{A},G_1}^{\text{CSUAC-IoT}} - \text{Adv}_{\mathcal{A},G_2}^{\text{CSUAC-IoT}}\right| \\ &\quad + \left|\text{Adv}_{\mathcal{A},G_2}^{\text{CSUAC-IoT}} - \text{Adv}_{\mathcal{A},G_3}^{\text{CSUAC-IoT}}\right| \\ &\leq \frac{q_h^2}{2|\text{Hash}|} + \text{Adv}_{\mathcal{A}}^{\text{ECDDHP}}(pt) + \max\left\{C'.q_s^{s'}, \frac{q_s}{2^{l_b}}\right\}. \end{aligned} \quad (7)$$

Simplification of (7) by multiplying its both sides by a factor of 2 produces the final desired result

$$\text{Adv}_{\mathcal{A}}^{\text{CSUAC-IoT}}(pt) \leq \frac{q_h^2}{|\text{Hash}|} + 2\left(\max\left\{C'.q_s^{s'}, \frac{q_s}{2^{l_b}}\right\} + \text{Adv}_{\mathcal{A}}^{\text{ECDDHP}}(pt)\right).$$

B. Informal Security Analysis

This section does nonmathematical security analysis on the proposed scheme to exhibit its resilience against the following attacks.

1) Impersonation Attacks: In this section, we mainly consider the following three scenarios under which an adversary, $\mathcal{A}$ can impersonate as a valid entity in the network.

1) *User Impersonation Attack:* Suppose $\mathcal{A}$ acts as a registered user U and wants to send an authorized message to the GN. To succeed such a purpose, assume that $\mathcal{A}$ chooses random secrets v_1', v_2', $v_3' \in \mathbb{Z}_n^*$, generates a new timestamp TS_U', and calculates $K_{11}' = H(v_1'.G)$ and $K_{21}' = H(v_2'.G)$. However, without having user U's partial private key k_U, it is "computationally infeasible problem" for $\mathcal{A}$ to calculate $(x_1', y_1') = v_1'.G + k_U.Y_G$, and $(x_2', y_2') = v_2'.G + k_U.Y_{Si}$. This makes computationally infeasible for $\mathcal{A}$ to calculate $C_1' = (\text{ID}_{S_i}||v_3') \oplus H(x_1'||k_U.Y_G)$, $r_1' = KH_{y_1'}(C_1'||K_{11}'||L_U||Y_G||TS_U')$, $r_{2i}' = KH_{y_2'}(K_{21}'||L_U||Y_{Si})$, $S_1' = (v_1'/[r_1' + k_U])\ (\text{mod}\ q)$ and $S_{2i}' = (v_2'/[r_{2i}' + k_U])\ (\text{mod}\ q)$, $T_1' = r_1'.G$ and $T_{2i}' = r_{2i}'.G$ and $W_{SK}' = H(r_1'||TS_U') \oplus H(v_3'||k_U.Y_G)$. Furthermore, to generate legal signatures Sign_G and Sign_{S_i}, $\mathcal{A}$ requires the private keys x_U and k_U, which are only available to the registered user U. As a result, $\mathcal{A}$ will not be able to create valid message $m_1' = \langle C_1', T_1', T_{2i}', S_1', S_{2i}', W_{SK}', TS_U', \text{Sign}_G, \text{Sign}_{S_i}\rangle$ on behalf of U. Hence, the proposed CSUAC-IoT is resilient against the "user impersonation attack."
2) *GN Impersonation Attack:* In order to impersonate the GN, assume that $\mathcal{A}$ intercepts the message $m_1 = \langle C_1, T_1, T_{2i}, S_1, S_{2i}, W_{SK}, TS_U, \text{Sign}_G, \text{Sign}_{S_i}\rangle$, generates a new timestamp TS_{GN}', obtains the identity ID_{S_i} of an accessed IoT smart device S_i, and tries to calculate $\text{temp}_1 = S_1.(T_1 + L_U)$ and $(x_1', y_1') = \text{temp}_1 + x_G.L_U$. However, without the private key x_G of the GN, it becomes "computationally infeasible task" for $\mathcal{A}$ to calculate (x_1', y_1') and hence, the signature Sign_G verification is not possible. Furthermore, to generate and send a valid message of the form $m_2' = \langle T_{2i}, S_{2i}, \overline{m_i'}, L_{\text{mod}}, TS_{\text{GN}}', \text{Sign}_{S_i}\rangle$, $\mathcal{A}$ requires the shared secret Z_i between the GN and S_i for calculating both L_{mod} and $\overline{m_i'} = H(r_1'||TS_U)' \oplus H(Z_i\ ||TS_{\text{GN}}'||L_U)$. This means that creating the valid message m_2' on behalf of the GN, it will be computationally infeasible task for the adversary $\mathcal{A}$. Thus, the "GN impersonation attack" is not possible in the proposed CSUAC-IoT.
3) *Smart Device Impersonation Attack:* To impersonate a valid smart device S_i, the adversary $\mathcal{A}$ needs to create a legal message of the form $m_3' = \langle VSK_{S_iU}', TS_{S_i}'\rangle$. For this goal, assume $\mathcal{A}$ generates a new timestamp

TS'_{S_i} and tries to calculate $VSK'_{S_iU} = H(SK'_{S_iU} ||TS'_{S_i})$ which further needs to calculate the valid session key shared with user U as $SK'_{S_iU} = H((r_{2i} + 1) ||H(r_1 ||TS_U)' ||x'_2 ||y'_2)$. Since the secrets x'_2 and y'_2 require the private key x_{S_i} of the smart device S_i and the shared secret key Z_i. Hence, generation of valid message m'_3 is computationally infeasible for the adversary $\mathcal{A}$ and the proposed CSUAC-IoT is resilient against the "smart device impersonation attack."

2) Replay Attack: The messages $m_1 = \{C_1, T_1, T_{2i}, S_1, S_{2i}, W_{SK}, TS_U, \text{Sign}_G, \text{Sign}_{S_i}\}$, $m_2 = \{T_{2i}, S_{2i}, \overline{m_i}, L_{\text{mod}}, TS_{\text{GN}}, \text{Sign}_{S_i}\}$, and $m_3 = \{VSK_{S_iU}, TS_{S_i}\}$ flowing over insecure channel among U, GN, and S_i contain the timestamps at which the messages were generated. The timestamps are verified by the receiver for integrity and freshness. Hence, the adversary $\mathcal{A}$ cannot gain advantage by replaying the messages in CSUAC-IoT, and the replay attack protection is assured.

3) Man-in-the-Middle Attack: In this attack, an adversary $\mathcal{A}$ captures the login and access control request message $m_1 = \{C_1, T_1, T_{2i}, S_1, S_{2i}, W_{SK}, TS_U, \text{Sign}_G, \text{Sign}_{S_i}\}$ from the public channel, and tries to tamper the request and generate another valid message m_1. $\mathcal{A}$ cannot generate different values for C_1, T_1, T_{2i}, S_1, S_{2i}, and W_{SK} to produce valid Sign_G and Sign_{S_i} because of the precomputed secret values x_U and k_U at the user U's side. $\mathcal{A}$ also fails to generate a valid message $m_2 = \{T_{2i}, S_{2i}, \overline{m_i}, L_{\text{mod}}, TS_{\text{GN}}, \text{Sign}_{S_i}\}$ because of the preshared secret key Z_i and the verified r_1 which is received from U. Similarly, $\mathcal{A}$ cannot tamper the session key verifier in $m_3 = \{VSK_{S_iU}, TS_{S_i}\}$ because of the shared parameters r_{2i}, r_1, x_1, and y_1 between U and S_i. Hence, CSUAC-IoT is not prone to the man-in-the-middle attack.

4) Guessing Attacks: We consider the following cases.

Online Password Guessing Attack: During the life cycle of the proposed CSUAC-IoT, a user U's identity ID_U, password Pw_U, and the biometrics Bio_U are never included in the message parameters which are transmitted over a public channel. Hence, $\mathcal{A}$ cannot gain knowledge about the identity and password of U by performing online guessing analysis on the messages m_1, m_2, and m_3.

Offline Password Guessing Attack: Let us consider that $\mathcal{A}$ is in the possession of a registered user U's mobile device MD_U. We assume that the mobile device MD_U is not tamper resistant and anyone who is in the possession of MD_U can extract the information $\{E_q(a, b), H(\cdot), KH(\cdot), G, x^*_U, Y_U, k^*_U, L_U, HPB_U, Gen(\cdot), Rep(\cdot), \tau_U, et\}$ stored in it (as stated in the threat model in Section II. To correctly guess the password Pw_U of user U, $\mathcal{A}$ must achieve one of the following.

1) For a successful guess on Pw_U using $x^*_U = x_U \oplus H(\text{ID}_U||Pw_U||\sigma_U)$, $\mathcal{A}$ must succeed in guessing ID_U, x_U, and σ_U simultaneously.
2) To guess a legitimate value for Pw_U using $k^*_U = k_U \oplus H(\sigma_U||\text{ID}_U||Pw_U)$, $\mathcal{A}$ should correctly guess k_U, σ_U, and ID_U at a time.
3) Similarly, to guess a legitimate value for Pw_U using $HPB_U = H(Pw_U||L_U||\sigma_U||\text{ID}_U)$, $\mathcal{A}$ has to guess correct values for ID_U, $L_U = d_U.Y_U$, and σ_U simultaneously.

From the above observations, it is clear that it is not feasible to guess any combination of required parameters simultaneously to obtain correct Pw_U. Hence, CSUAC-IoT is not vulnerable under both online and offline password guessing attacks.

5) Privileged-Insider Attack: A privileged insider at the TA has access to the messages which are sent securely to it during the user registration phase. With this assumption, the privileged insider with malicious behavior can access the user registration request $\{\text{ID}_U, Y_U\}$ which is sent to the TA by the user U via secure channel. It is to be noted that the adversary being the privileged insider cannot compute the partial private key d_U of user U because of the unknown parameter x_T, the private key of TA. Also, the password Pw_U of user U and the biometric secret key σ_U are never submitted to the TA. Assuming that such a privileged insider, acting as an adversary, has the user U's mobile device MD_U, he/she cannot gain knowledge on sensitive attributes such as Pw_U, k_U, x_U, and σ_U that require to know at least two of the unknown parameters x_U, d_U, k_U, Pw_U, and σ_U. Hence, CSUAC-IoT is secure against "privileged-insider attack."

6) Stolen Mobile Device Attack: Let us assume that the MD_U of an authorized user U is stolen or lost by an adversary $\mathcal{A}$. Then, $\mathcal{A}$ has knowledge of the information $\{E_q(a, b), H(\cdot), KH(\cdot), G, x^*_U, Y_U, k^*_U, L_U, HPB_U, Gen(\cdot), Rep(\cdot), \tau_U, et\}$ stored in MD_U. Now, consider the following two scenarios.

1) As described in Section V-B4, it is not possible to guess the password Pw_U of U. On the similar lines, it can be seen that guessing identity ID_U of U is also a hard/infeasbile task for $\mathcal{A}$.
2) Given the one-way nature of the "cryptographic hash function $H(\cdot)$," $\mathcal{A}$ must have knowledge on ID_U, Pw_U, σ_U, and L_U to derive x_U and k_U, and also to successfully tamper HPB_U.

Hence, CSUAC-IoT does not reveal any sensitive information if the mobile device MD_U of a legitimate user is stolen or lost.

7) IoT Smart Device Physical Capture Attack: Due to the tiny nature of IoT smart devices, they are prone to be physically taken into control by the adversary $\mathcal{A}$. Also, the hardware with which IoT devices operate are not tamper resistant and $\mathcal{A}$ can obtain the information stored in the smart device S_i. In this attack, we will analyze the impact of compromising an IoT smart device and the effect it causes on the entire network. During enrollment, each smart device S_i is preloaded with $\{\text{ID}_{S_i}, x_{S_i}\}$, where ID_{S_i} and x_{S_i} are the identity and secret key of S_i, respectively. It is also worth noticing that all the generated credential pairs $\{\text{ID}_{S_i}, x_{S_i}\}$ and also the shared secrets Z_i with the GN for all deployed smart devices S_i are distinct throughout the network. Any adversary who captures a smart device S_i can extract $\{\text{ID}_{S_i}, x_{S_i}\}$ from its memory. However, since the private key x_{S_i} of S_i and the shared secret Z_i are different for each smart device S_i, compromising S_i cannot impact other smart devices $\{S_j | j \neq i\}$ in the network. Thus, CSUAC-IoT does not suffer from the IoT smart device capture attack.

8) Ephemeral Secret Leakage Attack: In this attack, we analyze the intractability of the session key SK_{US_i} $(= SK_{S_iU})$ in case either the long-term secrets or short-term secrets are compromised. During the "login and access control phase," a user U and an IoT smart device S_i establish a shared session

key as $SK_{US_i} = H((r_{2i}+1)||H(r_1||TS_U)||x_2||y_2)$ $(= SK_{S_iU})$ for secure communication between them. Now, consider the following two cases.

Case I: Assume that the adversary $\mathcal{A}$ is in the possession of the long-term key k_U. If the short-term keys v_1, v_2, and v_3 are not revealed to $\mathcal{A}$, he/she cannot succeed in generating a valid session key SK_{US_i} because it is required to compute other secret credentials r_1 and r_{2i} which are derived using the short-term keys v_1 and v_2, respectively.

Case II: Assume that $\mathcal{A}$ gains knowledge on the short-term keys v_1, v_2, and v_3. $\mathcal{A}$ still cannot succeed in generating a valid session key SK_{US_i}, because to compute a valid session key SK_{US_i}, it is required to compute r_1 and r_{2i} which are derived using the long-term key k_U.

Hence, it is necessary to have knowledge of both long-term and short-term credentials to generate a valid session key. Also, it can be seen that the session keys generated for different sessions are independent from each other because of the random values v_1, v_2, and v_3 used in each session. As a result, compromising a session key by revealing the long-term and short-term keys does not compromise the future sessions and also it does not reveal the session keys for the previous sessions as well. This indicates that CSUAC-IoT is secure from the ESL attack, and it also ensures both "forward and backward secrecy" under the CK-adversary model [6].

9) Password Change Attack: Suppose an adversary $\mathcal{A}$ has obtained the stolen of lost mobile device MD_U of an authorized registered user U. Therefore, $\mathcal{A}$ will have the access to the stored credentials $\{E_q(a,b), H(\cdot), KH(\cdot), G, x_U^*, Y_U, k_U^*, L_U, HPB_U, Gen(\cdot), Rep(\cdot), \tau_U, et\}$ using the power analysis attack as stated in [7]. Now, assume that having these extracted credentials, $\mathcal{A}$ tries to change the user U's password Pw_U with newly chosen password, say $Pw_{\mathcal{A}}$. For this goal, $\mathcal{A}$ needs to first guess all the secret credentials ID_U, Pw_U, and Bio_U. Without local biometric and password verification by the MD_U, it is not possible to proceed for changing a new password $Pw_{\mathcal{A}}$. Even if $\mathcal{A}$ tries to update the old password Pw_U with $Pw_{\mathcal{A}}$, he/she requires to retrieve the secrets x_U and k_U from $x_U^* = x_U \oplus H(\mathrm{ID}_U||Pw_U||\sigma_U)$ and $k_U^* = k_U \oplus H(\sigma_U||\mathrm{ID}_U||Pw_U)$. Furthermore, assume that $\mathcal{A}$ imprints his/her own biometrics $Bio_{\mathcal{A}}$ and computes $Gen(Bio_{\mathcal{A}}) = (\sigma_{\mathcal{A}}, \tau_{\mathcal{A}})$. This means that calculating $x_{\mathcal{A}} = x_U \oplus H(\mathrm{ID}_U||Pw_{\mathcal{A}}||\sigma_{\mathcal{A}})$, $k_{\mathcal{A}} = k_U \oplus H(\sigma_{\mathcal{A}}||\mathrm{ID}_U||Pw_{\mathcal{A}})$, and $HPB_{\mathcal{A}} = H(Pw_{\mathcal{A}}||L_U||\sigma_{\mathcal{A}}||\mathrm{ID}_U)$ requires the secrets x_U and k_U, and also the identity ID_U. Hence, it is a computationally infeasible task for $\mathcal{A}$ to mount the password change attack in CSUAC-IoT.

10) Anonymity and Untraceability: Suppose an adversary $\mathcal{A}$ captures and inspects the messages m_1, m_2, and m_3 which are flowing over the public channel among user U, GN, and IoT smart device S_i. Note that the real identity ID_U of user U is never included directly in any of the messages. Therefore, $\mathcal{A}$ cannot relate a particular login request to a particular user U with identity ID_U. Hence, "user anonymity" is assured in CSUAC-IoT.

The message parameters in messages m_1, m_2, and m_3 are dynamically generated, which will differ between any two login and access control requests due to the use of random secrets and current timestamps. Hence, $\mathcal{A}$ cannot easily determine whether two login and access requests belong to the same user or not. This allows us to achieve "untraceability" in CSUAC-IoT.

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/akdas/span/testsuite
  /results/useraccess.if
GOAL
  as specified
BACKEND
  OFMC
STATISTICS
  TIME 144 ms
  parseTime 0 ms
  visitedNodes: 8 nodes
  depth: 3 plies
```

Fig. 3. Analysis of simulation results under OFMC backend.

VI. Formal Security Verification Using AVISPA

There are a number of formal security verification tools, such as automated validation of Internet security protocols and applications (AVISPAs) [29], ProVerif [30], and Scyther [31]. In this article, we use AVISPA due to its popularity among the security community. Specifically, we code CSUAC-IoT using the high-level protocol specification language (HLPSL) [29], which is a "role-oriented language" in which various basic roles (the roles for user U, GN, and IoT smart device S_i in our proposed CSUAC-IoT) and two mandatory roles (session and goal & environment) are defined. The HLPSL2IF translator helps in converting the HLPSL code to the "intermediate format (IF)," and the IF is then passed to one of the available four backends of AVISPA, such as on-the-fly model checker (OFMC), constraint logic-based attack searcher (CL-AtSe), SAT-based model checker (SATMC), and tree automata based on automatic approximations for the analysis of security protocols (TA4SP).

In our implementation, we perform three verifications for the proposed scheme (CSUAC-IoT), namely, executability checking on the nontrivial HLPSL specifications, replay attack validation, and DY model validation. We remark that the executability check is essential to assure whether CSUAC-IoT can reach to a state where a possible attack can occur during the protocol execution. To validate replay attack resilience, both the backends (OFMC and CL-AtSe) check whether legitimate agents can execute the specified protocol by performing a search of a passive intruder. The backends provide the intruder (i) about the knowledge of some normal sessions among the valid agents. Moreover, both OFMC and CL-AtSe backends verify whether any man-in-the-middle attacks can be carried out using the DY threat model. The findings are reported in Fig. 3, which demonstrate that CSUAC-IoT is secure against both replay and man-in-the-middle attacks.

TABLE IV
FUNCTIONALITY AND SECURITY: A COMPARATIVE SUMMARY

Attribute	Luo *et al.* [14]	Li *et al.* [12]	Li *et al.* [13]	Xue *et al.* [15]	Zeng *et al.* [16]	CSUAC-IoT
FA_1	✓	✓	✓	✓	✓	✓
FA_2	✓	✓	✓	✓	✓	✓
FA_3	×	×	×	✓	✓	✓
FA_4	✓	✓	✓	×	✓	✓
FA_5	✓	✓	✓	✓	✓	✓
FA_6	✓	✓	✓	✓	✓	✓
FA_7	×	×	×	×	×	✓
FA_8	×	×	×	×	×	✓
FA_9	✓	✓	✓	✓	×	✓
FA_{10}	✓	✓	×	×	✓	✓
FA_{11}	×	×	×	×	×	✓
FA_{12}	✓	✓	✓	✓	×	✓
FA_{13}	NA	NA	NA	NA	×	✓
FA_{14}	NA	NA	NA	NA	×	✓
FA_{15}	NA	NA	NA	NA	×	✓

FA_1: "replay attack"; FA_2: "man-in-the-middle attack"; FA_3: "mutual authentication"; FA_4: "key agreement"; FA_5: "impersonation attacks"; FA_6: "resilience against (mobile) device physical capture attack"; FA_7: "anonymity and untraceability preservation"; FA_8: "ESL attack under the CK-adversary model"; FA_9: "signcryption-based user access control mechanism"; FA_{10}: "formal security analysis"; FA_{11}: "formal security verification under AVISPA tool"; FA_{12}: "resistant to privileged-insider attack"; FA_{13}: "resistant to offline password guessing attack"; FA_{14}: "resistant to password/biometric change attack"; FA_{15}: "resistant to stolen/lost mobile device attack".
✓: "a scheme is secure or it provides a functionality feature"; ×: "a scheme is insecure or it does not provide a functionality feature"; NA: "not applicable in a scheme".

TABLE V
COMMUNICATION COSTS: A COMPARATIVE SUMMARY

Protocol	No. of messages	Total cost (in bits)
CSUAC-IoT	3	3136
Li *et al.* [12]	2	3488
Li *et al.* [13]	4	5408
Luo *et al.* [14]	2	3040
Xue *et al.* [15]	5	9344
Zeng *et al.* [16]	2	2080

TABLE VI
COMPUTATION COSTS: A COMPARATIVE SUMMARY

Protocol	Total cost	Approx. time (in milliseconds)
CSUAC-IoT	$T_{fe} + 14T_{ecm} + 8T_{eca} + 28T_h$	203.291
Luo *et al.* [14]	$3T_{ecm} + 4T_{bp} + 4T_h + T_{eca} + T_{me}$	173.621
Li *et al.* [12]	$3T_{ecm} + 5T_{bp} + 2T_h + 2T_{eca}$	204.054
Li *et al.* [13]	$9T_{me} + 6T_h + T_{bp} + T_{ecm} + T_{eca}$	66.776
Xue *et al.* [15]	$6T_{ecm} + 2T_{eca} + 6T_{bp} + 3T_{me} + 7T_h + 6T_{senc}/T_{sdec}$	284.345
Zeng *et al.* [16]	$9T_h + 2T_{ecm} + 4T_{eca} + 2T_{senc}/T_{sdec} + 2T_{bp} + 2T_{me}$	97.674

VII. COMPARATIVE ANALYSIS

This section compares the performance of the proposed CSUAC-IoT and the schemes of Li *et al.* [12], [13], Luo *et al.* [14], Xue *et al.* [15], and Zeng *et al.* [16].

A comparative summary of functionality and security is shown in Table IV. We observe that the schemes of Luo *et al.* [14] and Li *et al.* [12] do not support/provide the features FA_3, FA_7, FA_8, and FA_{11}. The features that are not supported in Li *et al.*[13] are FA_3, FA_7, FA_8, FA_{10}, and FA_{11}. The scheme of Xue *et al.* [15] does not support features FA_4, FA_7, FA_8, FA_{10}, and FA_{11}. In addition, the features that are supported in Zeng *et al.* [16] include FA_7–FA_9 and FA_{11}–FA_{15}.

For the analysis of communication costs, it is assumed that identity, random nonce (number), timestamp, hash output (if the SHA-1 hash algorithm is utilized [32]), and "elliptic curve point of the form $P = (P_x, P_y)$, where P_x and P_y are, respectively, x and y coordinates of the point P," need 160, 160, 32, 160, and $(160 + 160) = 320$ bits, respectively. In addition, a message in the existing schemes is taken as 1024 bits in size. We also consider the communication costs incurred during the login and access control phase. In our CSUAC-IoT, three messages $m_1 = \langle C_1, T_1, T_{2i}, S_1, S_{2i}, W_{SK}, TS_U, \text{Sign}_G, \text{Sign}_{S_i}\rangle$, $m_2 = \langle T_{2i}, S_{2i}, \overline{m_i}, L_{mod}, TS_{\text{GN}}, \text{Sign}_{S_i}\rangle$, and $m_3 = \langle VSK_{S_iU}, TS_{S_i}\rangle$ demand $(320+320+320+160+160+160+32+160+160) = 1792$ bits, $(320 + 160 + 160 + 320 + 32 + 160) = 1152$ bits and $(160+32) = 192$ bits, which collectively require 3136 bits. A comparative summary of communication costs is presented in Table V, and we observe that the schemes in [14] and [16] and CSUAC-IoT incur the least communication costs. However, our CSUAC-IoT achieves better security and functionality (see Table IV) compared to the schemes in [14] and [16].

Now, we study the computation costs during the login and access control phase in CSUAC-IoT and the other schemes. We use the following cryptographic operations with the following time (in milliseconds) [17], [33]: $T_h(\approx 0.056$ ms), $T_{ecm}(\approx 13.405$ ms), $T_{eca}(\approx 0.081$ ms), $T_{me}(\approx 2.249$ ms), $T_{bp}(\approx 32.713$ ms), $T_{fe}(\approx T_{ecm})$, and $T_{senc}/T_{sdec}(\approx T_h)$ represent the time require to "execute a one-way cryptographic hash function, an elliptic curve point (scalar) multiplication, an elliptic curve point addition, a modular exponentiation operation, a bilinear pairing, a fuzzy extractor function, and a symmetric encryption/decryption," respectively.

As shown in Table VI, in CSUAC-IoT, a user U, a GN, and an IoT device S_i incur $T_{fe}+ 14T_h+ 6T_{ecm}+ 2T_{eca}$, $4T_{ecm}+ 3T_{eca}+ 7T_h$, and $4T_{ecm}+ 3T_{eca}+ 7T_h$, respectively. Thus, the total computational cost in CSUAC-IoT is $T_{fe}+ 28T_h+ 14T_{ecm} +8T_{eca} \approx 203.291$ ms. The computational costs in the schemes of Luo *et al.* [14], Li *et al.* [12], Xue *et al.* [15], and Zeng *et al.* [16] are $3T_{ecm}+ 4T_{bp}+ 4T_h+ T_{eca} +T_{me} \approx 173.621$ ms, $3T_{ecm}+ 5T_{bp}+ 2T_h+ 2T_{eca} \approx 204.054$ ms,

$9T_{me}+ 6T_h+ T_{bp}+ T_{ecm}+ T_{eca} \approx 66.776$ ms, $6T_{ecm}+ 2T_{eca}+ 6T_{bp}+ 3T_{me} +7T_h +6T_{senc}/T_{sdec} \approx 284.345$ ms, and $9T_h+ 2T_{ecm}+ 4T_{eca} +2T_{senc}/T_{sdec} +2T_{bp} +2T_{me} \approx 97.674$ ms, respectively. In other words, CSUAC-IoT achieves better security and functionality (see Table IV), at the expense of slightly higher computation costs.

VIII. Conclusion

In this article, we presented our user access control scheme designed for an IoT setting. We then demonstrated its security formally in the ROR model and also informally (nonmathematically) showed that it is resilient against several common attacks. We also demonstrated using AVISPA simulation-based formal security verification that our scheme is resilient to both passive and replay and man-in-the-middle attacks. We then evaluated the performance of the proposed scheme with several others.

There are, however, a number of potential directions in which we can extend this article. For example, we intend to identify potential collaborators that can assist in the implementation of our proposed scheme in a real-world setting. In addition, we also intend to explore the utility of blockchain in a future design of the proposed scheme, for example, to achieve properties, such as decentralization, transparency, and immutability [34]–[36].

Acknowledgment

The authors would like to thank the anonymous reviewers and the Associate Editor for their valuable feedback.

References

[1] G. Glissa and A. Meddeb, "6LowPSec: An end-to-end security protocol for 6LoWPAN," *Ad Hoc Netw.*, vol. 82, pp. 100–112, Jan. 2019.

[2] (2017). *Information Matters. The Business of Data and the Internet of Things (IoT)*. Accessed: Oct. 2019. [Online]. Available: http://informationmatters.net/internet-of-things-statistics/

[3] M. Wazid, A. K. Das, R. Hussain, G. Succi, and J. J. P. C. Rodrigues, "Authentication in cloud-driven IoT-based big data environment: Survey and outlook," *J. Syst. Archit.*, vol. 97, pp. 185–196, Aug. 2019.

[4] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.

[5] Q. Do, B. Martini, and K.-K. R. Choo, "The role of the adversary model in applied security research," *Comput. Security*, vol. 81, pp. 156–181, Mar. 2019.

[6] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. Int. Conf. Theory Appl. Cryptograph. Tech. (EUROCRYPT)*, Amsterdam, The Netherlands, 2002, pp. 337–351.

[7] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.

[8] S. Zollner, K.-K. R. Choo, and N.-A. Le-Khac, "An automated live forensic and postmortem analysis tool for bitcoin on windows systems," *IEEE Access*, vol. 7, pp. 158250–158263, 2019.

[9] S. Li, K.-K. R. Choo, Q. Sun, W. J. Buchanan, and J. Cao, "IoT forensics: Amazon echo as a use case," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6487–6497, Aug. 2019.

[10] X. Zhang, K.-K. R. Choo, and N. L. Beebe, "How do I share my IoT forensic experience with the broader community? An automated knowledge sharing IoT forensic platform," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6850–6861, Aug. 2019.

[11] P. Kumar, A. Gurtov, J. Iinatti, M. Sain, and P. H. Ha, "Access control protocol with node privacy in wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 22, pp. 8142–8150, Nov. 2016.

[12] F. Li, Y. Han, and C. Jin, "Practical access control for sensor networks in the context of the Internet of Things," *Comput. Commun.*, vols. 89–90, pp. 154–164, Sep. 2016.

[13] F. Li, J. Hong, and A. A. Omala, "Efficient certificateless access control for industrial Internet of Things," *Future Gener. Comput. Syst.*, vol. 76, pp. 285–292, Nov. 2017.

[14] M. Luo, Y. Luo, Y. Wan, and Z. Wang, "Secure and efficient access control scheme for wireless sensor networks in the cross-domain context of the IoT," *Security Commun. Netw.*, vol. 2018, pp. 1–10, Feb. 2018. [Online]. Available: https://doi.org/10.1155/2018/6140978

[15] J. Xue, C. Xu, and Y. Zhang, "Private blockchain-based secure access control for smart home systems," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 12, pp. 6057–6078, 2018.

[16] X. Zeng, G. Xu, X. Zheng, Y. Xiang, and W. Zhou, "E-AUA: An efficient anonymous user authentication protocol for mobile IoT," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1506–1519, Apr. 2019.

[17] S. Malani, J. Srinivas, A. K. Das, K. Srinathan, and M. Jo, "Certificate-based anonymous device access control scheme for IoT environment," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9762–9773, Dec. 2019, doi: 10.1109/JIOT.2019.2931372.

[18] S. Banerjee *et al.*, "A provably secure and lightweight anonymous user authenticated session key exchange scheme for Internet of Things deployment," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8739–8752, Oct. 2019.

[19] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4900–4913, Dec. 2018.

[20] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. Adv. Cryptol. (EUROCRYPT)*, vol. 3027, 2004, pp. 523–540.

[21] M. Abdalla, P.-A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. 8th Int. Workshop Theory Pract. Public Key Cryptography (PKC)*, vol. 3386, Les Diablerets, Switzerland, 2005, pp. 65–84.

[22] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "TCALAS: Temporal credential-based anonymous lightweight authentication scheme for Internet of drones environment," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6903–6916, Jul. 2019.

[23] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Trans. Dependable Secure Comput.*, early access, 2017, doi: 10.1109/TDSC.2017.2764083.

[24] M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Design and analysis of secure lightweight remote user authentication and key agreement scheme in Internet of drones deployment," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3572–3584, Apr. 2019.

[25] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2776–2791, Nov. 2017.

[26] J. Srinivas, A. K. Das, M. Wazid, and N. Kumar, "Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial Internet of Things," *IEEE Trans. Dependable Secure Comput.*, early access, 2018, doi: 10.1109/TDSC.2018.2857811.

[27] S. Roy, A. K. Das, S. Chatterjee, N. Kumar, S. Chattopadhyay, and J. J. P. C. Rodrigues, "Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 457–468, Jan. 2019.

[28] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.

[29] (2019). *Automated Validation of Internet Security Protocols and Applications*. Accessed: Feb. 2019. [Online]. Available: http://www.avispa-project.org/

[30] M. Abadi, B. Blanchet, and H. Comon-Lundh, "Models and proofs of protocol security: A progress report," in *Proc. 21st Int. Conf. Comput.-Aided Verification (CAV)*, Grenoble, France, 2009, pp. 35–49.

[31] C. J. F. Cremers, "The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols," in *Proc. 20th Int. Conf. Comput.-Aided Verification (CAV)*, vol. 5123, 2008, pp. 414–418.

[32] *Secure Hash Standard, National Institute of Standards and Technology, U.S. Department of Commerce*, NIST Standard FIPS PUB 180-1, Apr. 1995. Accessed: Jan. 2019. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

[33] L. Wu, J. Wang, K.-K. R. Choo, and D. He, "Secure key agreement and key protection for mobile device user authentication," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 319–330, Feb. 2019.
[34] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.
[35] A. Singh, R. M. Parizi, Q. Zhang, K.-K. R. Choo, and A. Dehghantanha, "Blockchain smart contracts formalization: Approaches and challenges to address vulnerabilities," *Comput. Security*, vol. 88, Jan. 2020, Art. no. 101654.
[36] W. Dai, C. Dai, K.-K. R. Choo, C. Cui, D. Zou, and H. Jin, "SDTE: A secure blockchain-based data trading ecosystem," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 725–737, Jan. 2020.

Shobhan Mandal received the M.Tech. degree in computer science and information security from the International Institute of Information Technology Hyderabad, Hyderabad, India, in 2019.

He is currently a Software Developer with Huawei Technologies, Bengaluru, India. His research interests include cryptography and security in IoT-based devices.

Basudeb Bera received the M.Sc. degree in mathematics and computing from IIT (ISM) Dhanbad, Dhanbad, India, in 2014, and the M.Tech. degree in computer science and data processing from IIT Kharagpur, Kharagpur, India, in 2017. He is currently pursuing the Ph.D. degree in computer science and engineering with the Center for Security, Theory and Algorithmic Research, IIIT Hyderabad, Hyderabad, India.

His research interests are cryptography, network security, Internet of Things security, and blockchain.

Anil Kumar Sutrala received the Ph.D. degree in computer science and engineering from IIIT Hyderabad, Hyderabad, India, in 2018, and the M.C.A. degree from the University of Hyderabad, Hyderabad, in 2006.

He is currently working as a Principal Software Engineer with the CA Technologies—A Broadcom Company, Hyderabad. His research interests include cryptography and network security. He has published several journal articles in the above areas.

Ashok Kumar Das (Senior Member, IEEE) received the M.Tech. degree in computer science and data processing, the M.Sc. degree in mathematics, and the Ph.D. degree in computer science and engineering from IIT Kharagpur, Kharagpur, India.

He is currently an Associate Professor with the Center for Security, Theory and Algorithmic Research, IIIT Hyderabad, Hyderabad, India. His current research interests include cryptography and network security. He has authored over 200 papers in international journals and conferences in the above areas, including more than 175 reputed journal papers. Some of his research findings are published in top cited journals, such as the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON SMART GRID, the IEEE INTERNET OF THINGS JOURNAL, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, the IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, the *IEEE Consumer Electronics Magazine*, IEEE ACCESS, and *IEEE Communications Magazine*.

Dr. Das was a recipient of the Institute Silver Medal from IIT Kharagpur. He is on the editorial board of the *KSII Transactions on Internet and Information Systems*, the *International Journal of Internet Technology and Secured Transactions* (Inderscience), and *IET Communications*. He is a Guest Editor of *Computers and Electrical Engineering* (Elsevier) for the special issue on Big Data and IoT in E-healthcare and for *ICT Express* (Elsevier) for the special issue on Blockchain Technologies and Applications for 5G Enabled IoT. He has served as a program committee member in many international conferences. He also severed as one of the Technical Program Committee Chairs of the International Congress on Blockchain and Applications (BLOCKCHAIN'19), Avila, Spain, in June 2019.

Kim-Kwang Raymond Choo (Senior Member, IEEE) received the Ph.D. degree in information security from the Queensland University of Technology, Brisbane, QLD, Australia, in 2006.

He currently holds the Cloud Technology Endowed Professorship with the University of Texas at San Antonio (UTSA), San Antonio, TX, USA.

Dr. Choo was named the Cybersecurity Educator of the Year—APAC (Cybersecurity Excellence Awards are produced in cooperation with the Information Security Community on LinkedIn) in 2016 and his team won the Digital Forensics Research Challenge Organized by the University of Erlangen–Nuremberg, in 2015. He is the recipient of the 2019 IEEE Technical Committee on Scalable Computing Award for Excellence in Scalable Computing (Middle Career Researcher), the 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, British Computer Society's 2019 Wilkes Award Runner-Up, the 2019 EURASIP JWCN Best Paper Award, the Korea Information Processing Society's JIPS Survey Paper Award (Gold) 2019, the IEEE Blockchain 2019 Outstanding Paper Award, the Inscrypt 2019 Best Student Paper Award, the IEEE TrustCom 2018 Best Paper Award, the ESORICS 2015 Best Research Paper Award, the 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, the Fulbright Scholarship in 2009, the 2008 Australia Day Achievement Medallion, and the British Computer Society's Wilkes Award in 2008. He is the Co-Chair of IEEE Multimedia Communications Technical Committee's Digital Rights Management for Multimedia Interest Group. He is a Fellow of the Australian Computer Society.

Youngho Park (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic engineering from the Kyungpook National University, Daegu, South Korea, in 1989, 1991, and 1995, respectively.

He is currently a Professor with the School of Electronics Engineering, Kyungpook National University. He was a Professor with the School of Electronics and Electrical Engineering, Sangju National University, Daegu, South Korea, from 1996 to 2008. He was a Visiting Scholar with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA, from 2003 to 2004. His research interests include computer networks, multimedia, and information security.