Received January 29, 2017, accepted February 14, 2017, date of publication March 3, 2017, date of current version March 28, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2673239

Lightweight Three-Factor Authentication and Key Agreement Protocol for Internet-Integrated Wireless Sensor Networks

QI JIANG[1], SHERALI ZEADALLY[2], JIANFENG MA[1], AND DEBIAO HE[3,4]

[1]School of Cyber Engineering, Xidian University, Xi'an 710071, China
[2]College of Communication and Information, University of Kentucky, Lexington, KY 40506-0224, USA
[3]Software Engineering, Wuhan University, Wuhan 430072, China
[4]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

Corresponding author: Q. Jiang (jiangqixdu@gmail.com)

This work was supported in part by the National Science Foundation of China under Grant 61672413, Grant 61572379, Grant 61501333, Grant U1405255, Grant 61372075, and Grant U1536202, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2016JM6005, in part by Fundamental Research Funds for the Central Universities under Grant JB161501 and Grant JBG161511, in part by the National High Technology Research and Development Program (863 Program) of China under Grant 2015AA016007, and in part by the China 111 Project under Grant B16037.

ABSTRACT Wireless sensor networks (WSNs) will be integrated into the future Internet as one of the components of the Internet of Things, and will become globally addressable by any entity connected to the Internet. Despite the great potential of this integration, it also brings new threats, such as the exposure of sensor nodes to attacks originating from the Internet. In this context, lightweight authentication and key agreement protocols must be in place to enable end-to-end secure communication. Recently, Amin *et al.* proposed a three-factor mutual authentication protocol for WSNs. However, we identified several flaws in their protocol. We found that their protocol suffers from smart card loss attack where the user identity and password can be guessed using offline brute force techniques. Moreover, the protocol suffers from known session-specific temporary information attack, which leads to the disclosure of session keys in other sessions. Furthermore, the protocol is vulnerable to tracking attack and fails to fulfill user untraceability. To address these deficiencies, we present a lightweight and secure user authentication protocol based on the Rabin cryptosystem, which has the characteristic of computational asymmetry. We conduct a formal verification of our proposed protocol using ProVerif in order to demonstrate that our scheme fulfills the required security properties. We also present a comprehensive heuristic security analysis to show that our protocol is secure against all the possible attacks and provides the desired security features. The results we obtained show that our new protocol is a secure and lightweight solution for authentication and key agreement for Internet-integrated WSNs.

INDEX TERMS Authentication, biometrics, key management, privacy, Rabin cryptosystem, smart card, wireless sensor networks.

I. INTRODUCTION

One vision of future Internet is that objects and things with sensing and actuating capabilities will be connected and integrated making up the Internet of Things (IoTs). As Wireless Sensor Network (WSN) is one of the core technologies supporting the sensing capabilities required by future applications, the integration of WSN with the Internet will have an active role in the evolution of the architecture of future Internet [1]–[3]. The Internet Engineering Task Force (IETF) has developed a suite of protocols and open standards for integrating WSN into Internet [4], such as 6LoWPAN [5] and ROLL [6]. As illustrated in Fig. 1, sensor nodes (SNs) can be connected by low rate and low power wireless technologies such as IEEE 802.15.4, and can be further linked to the Internet via a 6LoWPAN gateway. Therefore, sensors will be globally addressable by any entity connected to the Internet thereby enabling the remote access of sensor data.

Despite its great potential, the integration of WSN with the Internet also brings new threats, such as the exposure of resource-constrained SNs and low rate wireless links in

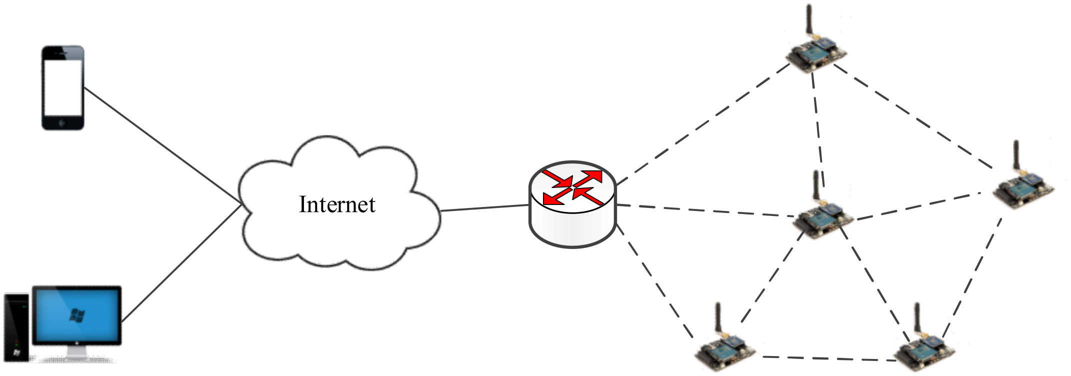


FIGURE 1. Typical architecture of Internet integrated wireless sensor networks.

WSN to attacks emanating from the Internet [3], [7]–[12]. Given its sensitivity and criticality, the sensor data in transit must be protected by an end-to-end (E2E) secure channel between the SN and the entity outside the WSN. The creation of such a channel requires authentication and key agreement mechanisms that allow two remote entities to mutually authenticate and negotiate secret keys that are used to protect the sensor data against various types of active and passive attacks [2], [13], [14]. Note that even if the WSN itself has security measures at a lower layer such as the link layer security services defined by IEEE 802.15.4, the openness of the Internet still requires authentication and key agreement protocols for establishing the E2E secure channel between the two communicating peers [2].

It is not possible to directly utilize Internet-centric security solutions because of the inherent characteristics of WSN (e.g., the limited computational capabilities and power supply of sensors and mobile devices) [2]. As discussed in [15], many attempts (for instance, IPsec [16], IKEv2 [17]) have been made to adapt standard Internet security protocols in this scenario.

However, resource limitation and the large number of SNs hinder the adoption of these solutions. Therefore, it is imperative to enable authentication and key establishment between the SN and the entity outside WSN in a secure and lightweight manner. However, past experiences [18]–[20] have shown that it is not trivial to design such security protocols.

A. RELATED WORKS

In the last decade, various security mechanisms [7], [21]–[25] have been proposed to prevent unauthorized access to the sensor data in transit. Li and Xiong [24] proposed a signcryption scheme to protect the information flow between a sensor and an entity outside the WSN, which fulfills confidentiality, integrity, authentication, and non-repudiation in one step. However, bilinear pairing is used in the scheme, which makes it unsuitable (because of its high computation and processing overheads) for regular SNs.

Astorga et al. [25] proposed the Ladon security protocol which provides E2E authentication and key establishment mechanism for resource-constrained devices. To prevent potential eavesdroppers from tracking users' access patterns, they also presented a privacy-enhanced Ladon protocol by integrating the original protocol with the PrivaKERB user privacy framework for Kerberos [7]. In these protocols, the long keys need to be securely stored and may be compromised.

To improve WSN security, two-factor authentication (2FA) protocols [26] have been introduced in WSN. In such protocols, two different types of security credential are used, i.e., smart card and password, to prove his/her identity. Over a dozen of 2FA schemes for WSNs have been proposed in recent years [18]–[20], [27]–[31]. We briefly review the ones closest to this work.

In 2014, Turkanovic et al. [32] proposed a lightweight 2FA protocol based on hash function for WSNs, which is claimed to be energy efficient and secure. However, Amin and Biswas [33] showed that the protocol of Turkanovic et al. [32] has several security weaknesses, including offline identity guessing attack, offline password guessing attack, impersonation attack, etc. To address these security deficiencies, they proposed a 2FA protocol for multi-gateway WSN. Independently, Farash et al. [34] also revealed that the protocol of Turkanovic et al. [32] is susceptible to smart card loss attacks (SCLA), impersonation attack, session key disclosure, et al. and proposed an improved 2FA protocol. In the same year, Chang and Le [35] demonstrated that the 2FA protocol by Turkanovic et al. [32] is prone to SN spoofing attack, stolen smart card attack, and stolen verifier attack, et al. Then they proposed two 2FA protocols P1 and P2 [37]. P1 is based on hash functions, while P2 employs Elliptic Curve Cryptography (ECC).

Lu et al. [36] analyzed Amin-Biswas's 2FA protocol [33] and identified several security drawbacks in their protocol. Next, they proposed an enhanced 2FA scheme using symmetric key cryptography, which is claimed to be resilient to a variety of attacks. Wu et al. [37] also pointed out that the 2FA protocol of Amin and Biswas [33] is insecure. Most recently, Das et al. [38] showed that both P1 and P2 by Chang and Le [35] are vulnerable to session specific temporary information attack and offline password guessing attack, etc. Amin et al. [39] found that the protocol of Farash et al. [34] cannot provide user anonymity, or withstand SCLA, offline password guessing attack, user impersonation attack, known session-specific temporary information attack (KSSTIA).

To address the vulnerabilities associated with the various 2FA approaches and further enhance the security strength of 2FA protocols, three-factor authentication (3FA) protocols have attracted the attention of researchers [40]–[44]. In the context of WSN, Das [45] presented a 3FA protocol based on symmetric cryptographic primitives. Next, he also proposed two other 3FA protocols [46]–[47]. Unfortunately, Wu et al. [48] demonstrated that all the three protocols are not secure. To address the drawbacks, they proposed an improved 3FA scheme based on ECC. Independently, Li et al. [49] presented a novel 3FA protocol based on the concept of biohashing. However, Das et al. [50] demonstrated that the protocol of Li et al. [49] is vulnerable to privileged insider attack, SN capture attack and cannot provide user anonymity.

Most recently, to remedy these security loopholes in the protocol of Farash et al. [34], Amin et al. [39] presented a new secure 3FA protocol, which is claimed to be secure against all the known security attacks. Additionally, to satisfy the practical requirements, their protocol provides the functionalities of post deployment, identity update, password update, and smart card revocation. However, we observe that the protocol by Amin et al. still has some subtle security weaknesses.

B. OUR RESEARCH CONTRIBUTIONS

Although several 3FA protocols [38], [39], [45]–[49] have been proposed in the literature, all these protocols either fail to provide adequate security protection or suffer from various security vulnerabilities. In this paper, we use the most recent 3FA protocol of Amin et al. [39] as a case study to show the challenge of designing a lightweight authentication protocol suitable for Internet integrated WSN. Then we propose an authentication protocol for Internet integrated WSN which exploits the computational asymmetry feature of Rabin cryptosystem. We summarize our main research contributions as follows:

- First, we analyze the most recent 3FA protocol of Amin et al. [39] and we present its security drawbacks. Specifically, we found that their protocol suffers from Type I SCLA (the secret data obtained from the smart card is enough for an adversary to reveal the user password) and Type II SCLA (the transcripts of an authentication session are needed for an attacker, in addition to the secret parameters in the user's smart card). Specifically, the user identity and password can be exhaustively guessed in an offline manner along with the secrets stored in the stolen smart card and the intercepted authentication messages. Additionally, the protocol suffers from KSSTIA if the temporal parameters in an authentication session are disclosed. Furthermore, the protocol is prone to tracking attack and cannot fulfill user untraceability.
- Second, we present an efficient and secure 3FA protocol based on the Rabin cryptosystem. Unlike other public key-based encryption algorithms such as RSA and ECC, Rabin has the characteristic of computational asymmetry. In this case, the encryption is very efficient while the decryption is relatively heavyweight. This feature is particular well suited for Internet integrated WSN because the mobile device of users is generally resource-constrained while the gateway has no such restriction.
- Third, we conducted a formal verification using ProVerif [51] to demonstrate that our protocol fulfills the required security features. Furthermore, we also present comprehensive heuristic security analysis to demonstrate that our protocol is capable of withstanding all the possible active and passive attacks including the security weaknesses revealed in the protocol of Amin et al., and we show how it provides the desired security features. Additionally, performance analysis shows that our proposed protocol is a practical solution that can provide authentication and key agreement for Internet integrated WSN, while achieving both security and efficiency.

The remainder of this paper is organized as follows. The preliminaries of Biohashing and Rabin cryptosystem are given in Section 2. We review and analyze Amin et al.'s protocol [39] in Section 3 and 4 respectively. In Section 5, we propose a novel 3FA and key agreement protocol for Internet integrated WSN. Section 6 and Section 7 present security and efficiency analyses of the new protocol. Section 8 concludes the paper.

II. PRELIMINARY

A. BIOHASHING

Biometric is widely used to verify the identity of a user. It offers several advantages over traditional authentication methods, i.e., password and smart card. Biometric feature data is closely coupled with each individual and cannot be replaced. As a result, the disclosure of biometric data leads to serious privacy risks. Numerous schemes have been proposed to preserve the privacy of biometric template [52]–[55].

Biohashing [55], [56] is one of the mainstream privacy-preserving biometric schemes. In the enrollment stage, a biohash value $BH(K, B)$ is generated from the biometric template B and a random secret key K. Specifically, a pre-processing is performed on B in order to make the biometric feature invariant to small variations in the input biometric signal. Then, the biohash value $BH(K, B)$ is generated by comparing the inner product of the random vector generated from the user specific secret key K and the feature vector extracted against a predefined threshold. In the verification stage, by following the process used at the enrolment stage, a biohash value $BH(K, B')$ can be generated from the received biometric signal B' and the secret key given by the user. Afterwards, the verification is done by comparing $BH(K, B')$ with the stored value $BH(K, B)$ [52].

B. RABIN CRYPTOSYSTEM

The Rabin cryptosystem [57], [58] is a public key cryptographic primitive based on integer factorization. The scheme includes three algorithms, i.e., key generation, encryption and decryption.

Key generation: We choose two large distinct primes p and q such that $p, q \equiv 3(\text{mod}4)$, and compute $N = pq$. Then N is the public key, and (p, q) is the private key.

Encryption: We encrypt a plaintext m by computing $c = m^2 \bmod N$.

Decryption: We decrypt a ciphertext $m = \sqrt{c} \bmod N$. Specifically, the receiver who knows the private key (p, q) can apply the Chinese remainder theorem to derive the four possible plaintexts $\{m_1, m_2, m_3, m_4\}$. One common technique used to identify the correct plaintext is to add some pre-defined padding in the plaintext or requires the plaintext to conform to some pre-defined format.

If y has a square root x, i.e., there is a solution for $y = x^2 \bmod N$, then y is a quadratic residue mod N. The quadratic residue problem is described as follows: for $y \in QR_n$, where QR_n is the set of all quadratic residues mod N, it is computationally infeasible to find x without knowing p and q due to the hardness of factoring N.

III. REVIEW OF Amin et al.'s PROTOCOL

Amin et al.'s 3FA protocol [39] consists of 9 phases, i.e., system setup, SN registration, user registration, login, authentication, post deployment, identity update, password change, and smartcard revocation. We use the notations listed in Table 1 throughout this paper.

TABLE 1. Notations.

Notation	Description
U_i	User
S_j	SN
SA	System administrator
SCN_i	Smart card number
GWN	Gateway node
PW_i	U_i 's password
ID_i	U_i 's identity
ID_j	S_j 's identity
X_{GWN-S_j} :	The key shared between GWN and S_j
X_{GWN} :	GW 's secret key
X_j	S_j 's secret key
K_i, K_j	Random number chosen by U_i and S_j , respectively
T	Timestamp
fng_i :	U_i 's biometric template
ΔT	Time delay threshold
SK	Session key established in the protocol
$h()$	One-way hash algorithm
$H()$	Bio-hashing algorithm defined in Amin et al.'s protocol [52]
$BK()$	The biometric key extraction algorithm defined in Amin et al.'s protocol
$BH(\cdot,\cdot)$	The two-factor bio-hashing algorithm defined in the new protocol [52]
B_i	Biometric key of fng_i
$\|\|$	Concatenation
$\oplus$	Bitwise XOR operation

A. SYSTEM SETUP

SA selects and computes the system parameters in offline mode.

Step 1: SA chooses a master secret key X_{GWN}.

Step 2: SA selects an identity ID_j and computes the secret key $X_j = h(ID_j \| X_{GWN})$ for each SN $S_j (1 \le j \le m)$.

Step 3: SA randomly generates a number R_{shrd}, which is shared between GWN and S_j. Finally, S_j stores $< ID_j, X_j, R_{shrd} >$ in its memory.

B. SN REGISTRATION

When GWN and $S_j (1 \le j \le m)$ are deployed to form a WSN. Each S_j executes the following procedure to register with GWN.

Step 1: S_j computes $S_1 = ID_j \oplus h(R_{shrd} \| Ts_1)$ and $S_2 = h(ID_j \| X_j \| R_{shrd} \| Ts_1)$, and then sends $< S_1, S_2, T_{s1} >$ to GWN.

Step 2: GWN verifies whether $|T_{GWN} - T_{s1}| \le \Delta T$ holds. If it is false, GWN rejects the request of S_j; otherwise, it computes $ID'_j = S_1 \oplus h(R_{shrd} \| T_{s1})$, $X' = h(ID'_j \| X_{GWN})$, and $S'_2 = h(ID'_j \| X'_j \| R_{shrd} \| T_{s1})$ and checks whether $S'_2 = S_2$ holds. If it is not true, GWN rejects S_j; otherwise, it accepts S_j and stores ID_j into the database. Then GWN sends a confirmation message to S_j.

Step 3: After receiving the confirmation message, S_j deletes R_{shrd} from its memory.

C. USER REGISTRATION

In this phase, U_i executes the following procedure to register with SA.

Step 1: U_i sends the selected identity ID_i and personal credentials to SA through a secure channel.

Step 2: SA checks whether ID_i exists in the database. If it does, SA indicates U_i to select a new identity; otherwise, SA computes $d_i = h(ID_i \| X_{GWN})$ and $L_i = h(SCN_i \| X_{GWN})$, then the smart card storing $< d_i, L_i, SCN_i, BK() >$ is delivered to U_i securely. SA maintains a database storing U_i's ID_i and credentials.

Step 3: U_i inserts the card into a card reader. U_i then enters $< ID_i, PW_i >$ and fingerprint fng_i. The card computes $B_i = BK(H(fng_i))$, $e_i = h(ID_i \| PW_i \| B_i)$, $f_i = d_i \oplus h(ID_i \| PW_i)$, and $g_i = L_i \oplus h(PW_i \oplus ID_i)$. Then the card stores $< B_i, e_i, f_i, g_i, SCN_i, BK() >$ and deletes $< d_i, L_i >$.

D. LOGIN

The following procedure is performed when U_i wishes to access sensor data.

Step 1: U_i inserts the smart card and imprints the fingerprint fng_i. Then, the card computes $B_i^* = BK(H(fng_i))$ and checks whether $B_i^* = B_i$. If $B_i^* \neq B_i$, the card denies U_i's login request; otherwise, U_i continues to enter his/her identity ID_i^* and password PW_i^*, and the card computes $e_i^* = h(ID_i^* \| PW_i^* \| B_i^*)$. The card rejects U_i's login request if $e_i^* \neq e_i$; otherwise, the entered ID_i^* and PW_i^* are valid.

Step 2: The card generates a random number K_i and the timestamp T_1, computes $d_i^* = f_i \oplus h(ID_i^* \| PW_i^*)$, $L_i^* = g_i \oplus h(PW_i^* \oplus ID_i^*)$, $M_1 = ID_i^* \oplus h(L_i^* \| T_1)$, $M_2 = K_i \oplus h(d_i^* \| T_1)$, $M_3 = h(d_i^* \| K_i \| T_1)$ and $SCT_i = SCN_i \oplus h(T_1)$.

Step 3: U_i selects the identity ID_j of the sensor that he/she wishes to access, then the card computes $EID_j = ID_j \oplus h(ID_i \| K_i \| T_1)$and sends $MSG_1 =< M_1, M_2, M_3, T_1, SCT_i, EID_j >$ to GWN.

E. AUTHENTICATION

U_i, GWN, and S_j mutually authenticate each other and negotiate a secret key through the following steps.

Step 1: After receiving MSG_1 from U_i, GWN computes $SCN_i = SCT_i \oplus h(T_1)$, $L_i' = h(SCN_i \| X_{GWN})$, $ID_i' = M_1 \oplus h(L_i' \| T_1)$, $d_i' = h(ID_i' \| X_{GWN})$, $K_i' = M_2 \oplus h(d_i' \| T_1)$, and $M_3' = h(d_i' \| K_i' \| T_1)$. GWN aborts the current session if $M_3' \neq M_3$; otherwise, it computes $M_4 = h(ID_i' \| d_i' \| T_1)$ and then forwards $MSG_1 =< M_4 >$ to U_i.

Step 2: U_i computes $M_4^* = h(ID_i \| d_i^* \| T_1)$. If $M_4^* \neq M_4$, U_i terminates the session; otherwise, he/she calculates $M_5 = h(d_i^* \| ID_i \| K_i \| T_1)$ and transmits $MSG_3 =< M_5 >$ to GWN.

Step 3: GWN calculates $M_5' = h(d_i' \| ID_i' \| K_i' \| T_1)$ and aborts the connection if $M_5' \neq M_5$; otherwise, it proceeds to execute the next procedure.

Step 4: GWN computes $ID_j' = EID_j \oplus h(ID_i \| K_i \| T_1)$, $X_j' = h(ID_j' \| X_{GWN})$, $M_6 = h(ID_i' \| ID_j' \| ID_{GWN} \| X_j' \| K_i' \| T_2)$, $M_7 = ID_i' \oplus h(ID_{GWN} \| X_j' \| T_2)$, $M_8 = K_i' \oplus h(ID_i' \| X_j')$ and then sends $MSG_4 =< ID_{GWN}, M_6, M_7, M_8, T_2 >$ to S_j.

Step 5: S_j checks whether $|T_3 - T_2| \leq \Delta T$ holds, where T_3 is the current timestamp. If it is invalid, S_j immediately terminates the session; otherwise, it computes $ID_i^{**} = M_7 \oplus h(ID_{GWN} \| X_j \| T_2)$, $K_i^{**} = M_8 \oplus h(ID_i^{**} \| X_j)$, and $M_6^{**} = M_7 \oplus h(ID_i^{**} \| ID_j \| ID_{GWN} \| X_j \| K_i^{**} \| T_2)$. S_j aborts the connection if $M_6^{**} \neq M_6$; otherwise, it accepts that U_i and GWN are legitimate. Next, S_j computes $SK_j = h(ID_i^{**} \| ID_j \| K_i^{**} \| K_j)$, $M_9 = h(SK_j \| X_j \| K_j \| T_3)$, and $M_{10} = K_i^{**} \oplus K_j$, where K_j is a random number generated by S_j. Finally, S_j forwards $MSG_5 =< M_9, M_{10}, T_3 >$ to GWN.

Step 6: GWN checks whether $|T_4 - T_3| \leq \Delta T$ holds, where T_4 is the current timestamp. If it is invalid, GWN aborts the session; otherwise, it computes $K_j' = M_{10} \oplus K_i'$, $SK_{GWN} = h(ID_i' \| ID_j \| K_i' \| K_j')$, and $M_9' = h(SK_{GWN} \| X_j' \| K_j' \| T_3)$. GWN rejects the session if $M_9^* \neq M_9$; otherwise, it computes $M_{11} = h(SK_{GWN} \| ID_i' \| d_i \| K_j')$. Finally, GWN sends $MSG_6 =< M_{11}, M_{10} >$ to U_i.

Step 7: U_i computes $K_j^* = M_{10} \oplus K_i$, $SK_i = h(ID_i \| ID_j \| K_i \| K_j^*)$, and $M_{11}^* = h(SK_i \| ID_i \| d_i \| K_j^*)$. U_i rejects the session if $M_{11}^* \neq M_{11}$; otherwise, U_i accepts that GWN and S_j are authentic. The session key $SK_i = SK_j = SK_{GWN}$ are shared among U_i, S_j, and GWN.

F. POST DEPLOYMENT

In this phase, SNs can be deployed after the installation of a WSN. Assume that a new SN S_k is required to be deployed in the target field. SA first chooses the identity ID_k of S_k, computes $X_k = h(ID_k \| X_{GWN})$, and writes $< ID_k, X_k, R_{shrd} >$ into the memory of S_k. Then S_k executes the SN registration phase to register with GWN.

G. IDENTITY UPDATE

In this phase, a registered user securely updates his/her identity as follows.

Step 1: Step 1 of the login phase is performed to verify the legitimacy of U_i. Then U_i inputs a new identity ID_i^{new}, and then the smart card computes $d_i^* = f_i \oplus h(ID_i \| PW_i)$, $L_i^* = g_i \oplus h(PW_i \oplus ID_i)$, $Z_i = h(d_i^* \| ID_i \| T_{id})$, $W_i = ID_i \oplus h(L_i \| T_{id})$, $SCT_i = SCN_i \oplus h(T_{id})$, and $DD_i^* = ID_i^{new} \oplus h(L_i \| d_i \| T_{id})$. The card then sends $< Z_i, W_i, DD_i, SCT_i, T_{id} >$ to GWN.

Step 2: GWN computes $SCT_i = SCN_i \oplus h(T_{id})$, $L_i' = h(SCN_i \| X_{GWN})$, $ID_i' = W_i \oplus h(L_i' \| T_{id})$, $d_i' = h(ID_i \| X_{GWN})$ and $Z_i' = h(d_i' \| ID_i' \| T_{id})$, and then checks whether $Z_i' = Z_i$ holds. If it holds, GWN believes that U_i is authentic. GWN then computes $ID_i^{new} = DD_i \oplus h(L_i' \| d_i' \| T_{id})$, $d_i^{**} = h(ID_i^{new} \| X_{GWN})$, $Y_i = d_i^{**} \oplus d_i'$, $ZZ_i = h(d_i^{**} \| Z_i')$, and sends $< ZZ_i, Y_i >$ to the smart card and updates ID_i^{new} in the database.

Step 3: The smart card calculates $d_i^{**} = Y_i \oplus d_i'$ and $ZZ_i = h(d_i^{**} \| Z_i)$, and checks whether $ZZ_i^* = ZZ_i$ holds. If it holds, the card computes $e_i^{new} = h(ID_i^{new} \| PW_i \| B_i)$, $f_i^{new} = dd_i^{**} \oplus h(ID_i^{new} \| PW_i)$, and $g_i^{new} = L_i \oplus h(ID_i^{new} \oplus PW_i)$. Finally, the card updates $< e_i, f_i, g_i >$ with $< e_i^{new}, f_i^{new}, g_i^{new} >$.

H. PASSWORD CHANGE

In this phase, U_i updates the password PW_i locally as follows.

Step 1: U_i inserts his or her smart card into the card reader and executes Step 1 of the login phase to verify the validity of fingerprint, password, and identity.

Step 2: U_i inputs a new password PW_i^{new}, and the card computes $e_i^{new} = h(ID_i \| PW_i^{new} \| B_i)$, $d_i' = f_i \oplus h(ID_i \| PW_i)$, $f_i^{new} = d_i^{new} \oplus h(ID_i \| PW_i^{new})$, $L_i' = g_i \oplus h(PW_i \| ID_i)$, and $g_i^{new} = L_i' \oplus h(ID_i \oplus PW_i^{new})$.

Step 3: The card updates $< e_i, f_i, g_i >$ with $< e_i^{new}, f_i^{new}, g_i^{new} >$.

I. SMART CARD REVOCATION

If U_i's smart card is stolen or lost, U_i can obtain a new smart card as follows.

Step 1: U_i sends the identity ID_i and his credential to SA through a secure channel. SA first verifies U_i's credential, if it is valid, it computes $d_i^{new} = h(ID_i \| X_{GWN})$ and $L_i^{new} = h(SCN_i^{new} \| X_{GWN})$, where SCN_i^{new} is the new smart card number. Then, the new card storing $< d_i^{new}, L_i^{new}, SCN_i^{new}, BK() >$ is sent to U_i securely. Then, SA updates the database with SCN_i^{new}.

Step 2: U_i attaches the smart card into a card reader, enters ID_i and PW_i, and provides fingerprint fng_i at the biometric capturing device. The card then computes $B_i^{new} = BK(H(fng_i))$, $e_i^{new} = h(ID_i \| PW_i \| B_i^{new})$, $f_i^{new} = d_i^{new} \oplus h(ID_i \| PW_i)$ and $g_i^{new} = L_i^{new} \oplus h(ID_i \oplus PW_i)$. Finally, the smart card stores $< B_i^{new}, e_i^{new}, f_i^{new}, g_i^{new}, SCN_i^{new}, BK() >$ into its memory, and deletes $< d_i^{new}, L_i^{new} >$.

IV. WEAKNESSES OF THE PROTOCOL BY Amin et al.

Before presenting the cryptanalysis of Amin et al.'s protocol, we first briefly present the adversarial model [59]–[61].

1) The adversary A may capture all messages sent or received in the authentication session.

2) A can either (i) obtain the password of a registered user, or (ii) obtain a stolen or lost smart card of the user, and reveal the secret parameters in it by side channel attacks [62], [63], but not both at the same time.

3) A has the capability of enumerating offline all possible candidates in the Cartesian product $D_{id} * D_{pw}$ in polynomial time, where D_{id} and D_{pw} denote the identity space and the password space respectively.

4) A may somehow learn the identity ID_i of the victim when considering security properties (such as mutual authentication and session key security) and attacks (such as impersonation attack and SCLA).

In [39], Amin et al. claimed that their protocol can withstand various attacks even if the smart card is stolen. However, we show that Amin et al.'s protocol is prone to Type I SCLA, Type II SCLA, KSSTIA and tracking attack. Thus, Amin et al.'s protocol is not actually suitable for practical deployment.

A. TYPE I SCLA

In a SCLA, A attempts to guess U_i's identity and password after extracting information from the smart card. It is worth noting that it is widely accepted when designing password-based protocols that the space of D_{pw} is enumerable [64]. SCLAs can be classified into eight types [61]. In this paper, we focus on the attacks involving the extraction of secret information from a lost smart card, and classify them into two types, type I and II. In type I SCLA, the secret data obtained from U_i's card is enough for A to reveal U_i's password. In type II SCLA, the transcripts of an authentication session are needed for A, in addition to the secret parameters in U_i's smart card.

In [39], the authors assumed that the probability of guessing ID_i and PW_i using e_i is negligible. However, Wang et al. [59] pointed out that the identity of a user can be revealed by the attacker when the user's smart card is lost. Thus it is more prudent to take this risk into consideration.

Suppose that the smart card is somehow acquired by A, and then A reveals the parameters $< B_i, e_i, f_i, g_i, SCN_i, BK() >$ from the card. With the secret information $e_i = h(ID_i \| PW_i \| B_i)$ and B_i, A may conduct an offline password guessing attack described below [61].

Step 1: A guesses a possible value ID_i^* of ID_i and a value PW_i^* of PW_i.

Step 2: A calculates $e_i^* = h(ID_i^* \| PW_i^* \| B_i)$ and validates the correctness of ID_i^* and PW_i^* by checking whether $e_i ? = e_i^*$ holds. If it is positive, A has found out the correct pair of identity and password. Otherwise, A repeats the steps (1) and (2) until $e_i = e_i^*$.

Let $|D_{pw}|$ and $|D_{id}|$ denote the dictionary space of D_{pw} and D_{id}, respectively. In practice, the dictionary size is $|D_{pw}| \leq |D_{id}| \leq 10^6$ [61]. The time complexity of the above attack is $O(|D_{pw}| * |D_{id}| * T_H)$, where T_H is the execution time for the Hash operation. Hence, the time needed for A to carry out this attack is linear to $|D_{pw}| * |D_{id}|$.

The root cause of the above attack is that there is a definite password verifier (i.e., e_i) stored in U_i's card. As a result, it can be utilized by A to offline guess U_i's password.

B. TYPE II SCLA

Besides the assumption about the smart card in Type I SCLA, it is widely accepted that A can capture the messages (e.g., $MSG_1 =< M_1, M_2, M_3, T_1, SCT_i, EID_j >$) exchanged between U_i and S in the process of authentication. Then, A can exhaustively guess U_i's password PW_i as follows:

Step 1: A guesses a possible value ID_i^* of ID_i and a value PW_i^* of PW_i.

Step 2: A computes $L_i^* = g_i \oplus h(PW_i^* \oplus ID_i^*)$ and $M_1^* = ID_i^* \oplus h(L_i^* \| T_1)$, where g_i is revealed from U_i's smart card, T_1 is captured from the open channel.

Step 3: A checks whether $M_1^* ? = M_1$. If it is positive, A has found the correct pair of identity and password. Otherwise, A repeats the steps (1) to (3) until $M_1^* = M_1$.

The time complexity of the above attack is $O(|D_{pw}| * |D_{id}| * 2T_H)$, and the time required for A to carry out this attack is linear to $|D_{pw}| * |D_{id}|$.

C. KNOWN-SESSION SPECIFIC TEMPORARY INFORMATION ATTACK

In the authentication phase, if U_i is legitimate, GWN sends $MSG_4 =< ID_{GWN}, M_6, M_7, M_8, T_2 >$ to S_j, where $M_8 = K_i \oplus h(ID_i' \| X_j')$, K_i is a random number chosen by U_i. After verifying the authenticity of GWN, S_j forwards $MSG_5 =< M_9, M_{10}, T_3 >$ to GWN, where $M_{10} = K_i^{**} \oplus K_j$, and K_j is the random number generated by S_j. The session key between U_i and S_j is $SK_j = h(ID_i^{**} \| ID_j \| K_i^{**} \| K_j)$. If K_i is compromised, A can derive the value $h(ID_i' \| X_j') = K_i \oplus M_8$, which is static and remains unchanged in all authentication sessions between U_i and S_j. With this value, A can derive all the previous and future session key between U_i and S_j. A first derives the random numbers $K_i = M_8 \oplus h(ID_i' \| X_j')$ and $K_j = K_i^{**} \oplus M_{10}$, and then computes $SK_j = h(ID_i^{**} \| ID_j \| K_i^{**} \| K_j)$. We note that the disclosure of a random number in one authentication session will compromise all the session keys. Therefore, Amin et al.'s protocol has the problem of KSSTIA.

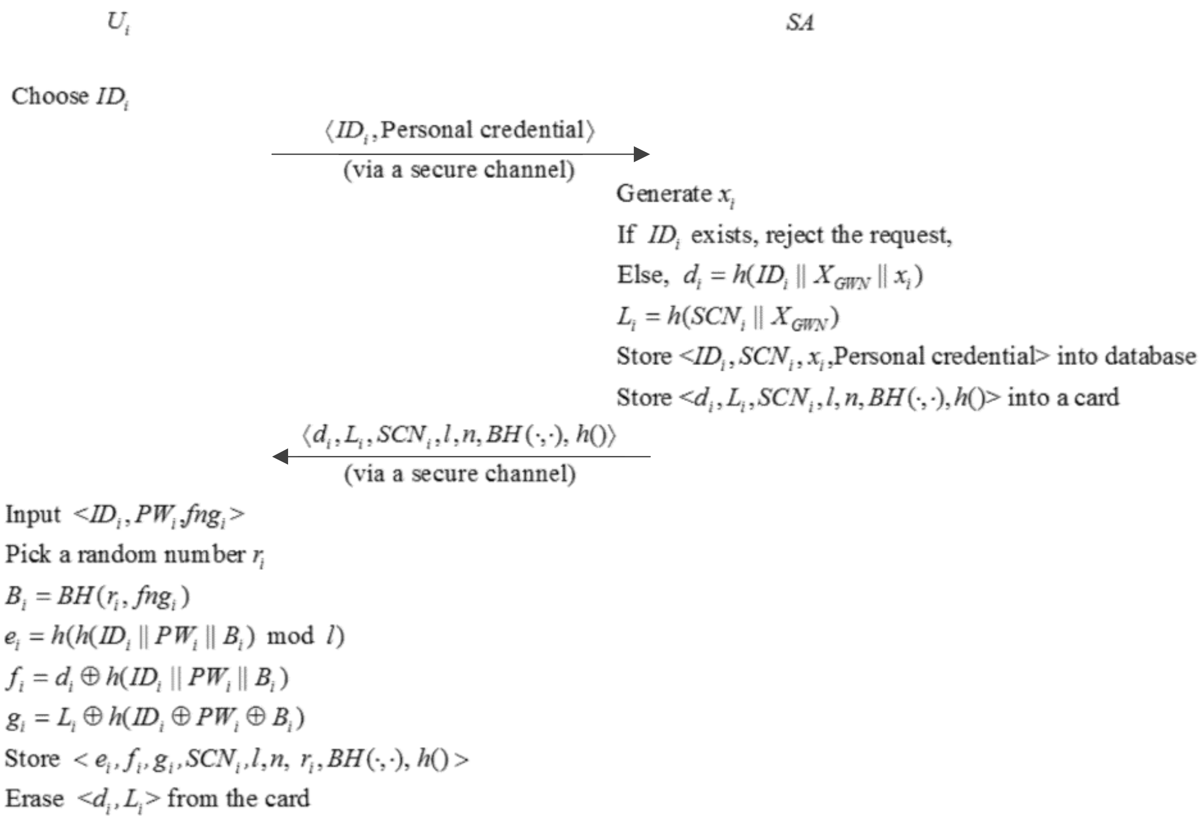

U_i
SA
Choose ID_i
$\langle ID_i$,Personal credential$\rangle$
(via a secure channel)
Generate x_i
If ID_i exists, reject the request,
Else, $d_i = h(ID_i \| X_{GWN} \| x_i)$
$L_i = h(SCN_i \| X_{GWN})$
Store $<ID_i, SCN_i, x_i$,Personal credential$>$ into database
Store $<d_i, L_i, SCN_i, l, n, BH(\cdot,\cdot), h()>$ into a card
$\langle d_i, L_i, SCN_i, l, n, BH(\cdot,\cdot), h()\rangle$
(via a secure channel)
Input $<ID_i, PW_i, fng_i>$
Pick a random number r_i
$B_i = BH(r_i, fng_i)$
$e_i = h(h(ID_i \| PW_i \| B_i) \bmod l)$
$f_i = d_i \oplus h(ID_i \| PW_i \| B_i)$
$g_i = L_i \oplus h(ID_i \oplus PW_i \oplus B_i)$
Store $< e_i, f_i, g_i, SCN_i, l, n, r_i, BH(\cdot,\cdot), h() >$
Erase $<d_i, L_i>$ from the card

FIGURE 2. User registration phase of our proposed protocol.

D. TRACKING ATTACK

When U_i wishes to access sensor data, U_i sends the message $MSG_1 =< M_1, M_2, M_3, T_1, SCT_i, EID_j >$ to initiate the authentication session. Although ID_i is concealed in M_1, and each field in MSG_1 is dynamic, the identity of the smart card SCN_i can be derived as $SCN_i = SCT_i \oplus h(T_1)$. Generally, the value of SCN_i is fixed for a specific user, which is generated by SA in the registration phase, and updated only in the smart card revocation phase. With this fixed value, an adversary A can launch tracking attack as follows.

Suppose that a legal user U_i interacts with GWN. A first captures U_i's message $MSG_1 =< M_1, M_2, M_3, T_1, SCT_i, EID_j >$, then retrieves and stores SCN_i. Then A can threaten the privacy of U_i through two ways [29]. Firstly, if A obtains U_i's identity by accident, then he/she is capable of identifying the user at the instant that U_i interacts with GWN by using the value SCN_i. Moreover, even though A cannot obtain ID_i, he/she is always capable of identifying different authentication sessions of U_i via the value SCN_i derived from each message $MSG_1 =< M_1, M_2, M_3, T_1, SCT_i, EID_j >$. Then, he/she might collect various types of sensitive information related to U_i, such as U_i's traveling routes, sensor access patterns, which may help A to violate user anonymity provided in Amin et al.'s protocol [39]. Therefore, Amin et al.'s protocol [39] is prone to tracking attack and cannot provide untraceability.

V. OUR PROPOSED AUTHENTICATION PROTOCOL

We enhance Amin et al.'s protocol as follows: (1) The public key primitive Rabin cryptosystem is employed to avoid SCLA and tracking attack. (2) The concept of fuzzy verifier [60] is adopted to achieve local password verification. (3) The timestamp mechanism mitigates session specific temporary information attack. Our new protocol also has 9 phases. SN registration and post deployment phase which remain unchanged are omitted here.

A. SYSTEM SETUP

SA selects and computes the system parameters in offline mode.

Step 1: SA first generates two large primes p and q, and computes $N = pq$, and keeps (p, q) as the private key. Then SA selects a master secret key X_{GWN} and an integer $2^4 \leq l \leq 2^8$ as the parameter of fuzzy verifier.

Step 2: SA selects an identity ID_j and computes the secret key $X_j = h(ID_j \| X_{GWN})$ for $S_j (1 \leq j \leq m)$.

Step 3: SA randomly generates a number R_{shrd}, which is shared between GWN and S_j. Finally, S_j stores $< ID_j, X_j, R_{shrd} >$ in its memory.

B. USER REGISTRATION

In this phase, U_i executes the following procedure to register with SA as shown in Fig. 2.

Step 1: U_i sends the selected identity ID_i and personal credentials to SA through a secure channel.

Step 2: SA checks whether ID_i exists in the database. If it does, SA indicates U_i to select a new identity; otherwise, SA generates a random number x_i, computes $d_i = h(ID_i\|X_{GWN}\|x_i)$ and $L_i = h(SCN_i\|X_{GWN})$. Then SA delivers the smart card storing $< d_i, L_i, SCN_i, l, n, BH(\cdot,\cdot), h() >$ to U_i securely. SA maintains a database storing each U_i's parameters <ID_i, SCN_i, x_i,Personal credential>.

Step 3: U_i attaches the card into a card reader. Then he/she enters $< ID_i, PW_i >$ and imprints fng_i. The card selects a random number r_i, computes $B_i = BH(r_i, fng_i)$, $e_i = h(h(ID_i\|PW_i\|B_i) \bmod l)$, $f_i = d_i \oplus h(ID_i\|PW_i\|B_i)$, and $g_i = L_i \oplus h(ID_i \oplus PW_i \oplus B_i)$. Finally, the smart card stores $< e_i, f_i, g_i, SCN_i, l, n, r_i, BH(\cdot,\cdot), h() >$ into its memory and deletes $< d_i, L_i >$.

C. LOGIN

The following procedure is performed when U_i wishes to access sensor data as shown in Fig. 3.

Step 1: U_i attaches the smart card and enters the identity ID_i^*, password PW_i^*, and fingerprint fng_i. Then, the card computes $B_i^* = BH(r_i, fng_i)$ and $\mathrm{e}_i^* = h(h(ID_i^*\|PW_i^*\|B_i^*) \bmod l)$. The card rejects U_i's login request if $e_i^* \neq e_i$.

Step 2: The card generates a random number K_i and a timestamp T_1, calculates $d_i^* = f_i \oplus h(ID_i^*\|PW_i^*\|B_i^*)$, $L_i^* = g_i \oplus h(ID_i^* \oplus PW_i^* \oplus B_i^*)$, $M_1 = (ID_i\|SCN_i\|K_i)^2 \bmod n$, $M_2 = h(d_i^*\|L_i^*\|K_i\|T_1)$.

Step 3: U_i selects the identity ID_j of the sensor that he/she wishes to access, then the card computes $EID_j = ID_j \oplus h(ID_i\|K_i\|T_1)$and sends $MSG_1 =< M_1, M_2, T_1, EID_j >$ to GWN.

D. AUTHENTICATION

To achieve mutual authentication and session key agreement among U_i, GWN, and S_j, the following steps are executed as shown in Fig. 3.

Step 1: After receiving MSG_1 from U_i, GWN decrypts M_1 using p and q to obtain ID_i', SCN_i', K_i', and then retrieves x_i according to ID', and verifies whether SCN_i' matches the value in the entry. If the two values do not match, then GWN rejects the request and aborts; otherwise, GWN computes $L_i' = h(SCN_i'\|X_{GWN})$, $d_i' = h(ID_i'\|X_{GWN}\|x_i)$, $K_i' = M_2 \oplus h(d_i'\|T_1)$, and $M_2' = h(d_i'\|L_i'\|K_i'\|T_1)$. GWN aborts the current session if $M_2' \neq M_2$; otherwise, GWN computes $ID_j' = EID_j \oplus h(ID_i\|K_i\|T_1)$, $X_j' = h(ID_j'\|X_{GWN})$, $M_3 = h(ID_i'\|ID_j'\|ID_{GWN}\|X_j'\|K_i'\|T_2)$, $M_4 = ID_i' \oplus h(ID_{GWN}\|X_j'\|T_2)$, $M_5 = K_i \oplus h(ID_i'\|ID_j'\|X_j'\|T_2)$ and then sends $MSG_2 =< ID_{GWN}, M_3, M_4, M_5, T_2 >$ to S_j.

Step 2: S_j checks whether $|T_3 - T_2| \leq \Delta T$ holds, where T_3 is the current timestamp. If it is invalid, S_j immediately terminates the session; otherwise, it computes $ID_i^{**} = M_4 \oplus h(ID_{GWN}\|X_j\|T_2)$, $K_i^{**} = M_5 \oplus h(ID_i^{**}\|ID_j\|X_j\|T_2)$, and $M_3^{**} = h(ID_i^{**}\|ID_j\|ID_{GWN}\|X_j\|K_i^{**}\|T_2)$. S_j aborts the connection if $M_3^{**} \neq M_3$; otherwise, it accepts that U_i and GWN are legitimate. Next, S_j computes $SK_j = h(ID_i^{**}\|ID_j\|K_i^{**}\|K_j)$, $M_6 = h(SK_j\|X_j\|K_j\|T_3)$, and $M_7 = K_i^{**} \oplus K_j$, where K_j is a random number generated by S_j. Finally, S_j forwards $MSG_3 =< M_6, M_7, T_3 >$ to GWN.

Step 3: GWN checks whether $|T_4 - T_3| \leq \Delta T$ holds, where T_4 is the current timestamp. If it is negative, GWN aborts the session; otherwise it computes $K_j' = M_7 \oplus K_i'$, $SK_{GWN} = h(ID_i'\|ID_j\|K_i'\|K_j')$, and $M_6' = h(SK_{GWN}\|X_j'\|K_i'\|T_3)$. GWN rejects the session if $M_6' \neq M_6$; otherwise, it computes $M_8 = h(SK_{GWN}\|ID_i'\|d_i'\|K_j')$. Finally, GWN sends $MSG_4 =< M_7, M_8 >$ to U_i.

Step 4: U_i computes $K_j^* = M_7 \oplus K_i$, $SK_i = h(ID_i\|ID_j\|K_i\|K_j^*)$, and $M_8' = h(SK_i\|ID_i\|d_i\|K_j^*)$. U_i rejects the session if $M_8^* \neq M_8$; otherwise, U_i accepts that GWN and S_j are authentic. At this point, a session key $SK_i = SK_j = SK_{GWN}$ has been established among U_i, S_j, and GWN.

E. IDENTITY UPDATE

In this phase, a registered user securely updates the identity as follows.

Step 1: U_i attaches the card and enters the identity ID_i^*, password PW_i^*, and fingerprints fng_i. Then, the card computes $B_i^* = BH(r_i, fng_i)$ and $\mathrm{e}_i^* = h(h(ID_i\|PW_i\|B_i) \bmod l)$. The card rejects U_i's login request if $e_i^* \neq e_i$. Then U_i inputs a new identity ID_i^{new}, and then the card generates a timestamp T_{id}, computes $d_i^* = f_i \oplus h(ID_i^*\|PW_i^*\|B_i^*)$, $L_i^* = g_i \oplus h(ID_i^* \oplus PW_i^* \oplus B_i^*)$, $DD_i = (ID_i\|SCN_i\|ID_i^{new})^2 \bmod n$, $Z_i = h(d_i^*\|L_i^*\|ID_i^{new}\|T_{id})$. The card then sends $< Z_i, DD_i, T_{id} >$ to GWN.

Step 2: GWN decrypts DD_i using p and q to obtain ID_i', SCN_i', ID_i^{new}, then retrieves x_i according to ID_i', and verifies whether SCN_i' matches the value in the entry. If the two values do not match, then GWN terminates; otherwise, GWN computes $L_i' = h(SCN_i'\|X_{GWN})$, $d_i' = h(ID_i'\|X_{GWN}\|x_i)$, $Z_i' = h(d_i^*\|L_i^*\|ID_i^{new}\|T_{id})$. GWN aborts the current session if $Z_i' \neq Z_i$; otherwise, GWN computes $d_i^{**} = h(ID_i^{new}\|X_{GWN}\|x_i)$, $Y_i = d_i^{**} \oplus h(d_i'\|T_{id})$, and $ZZ_i = h(d_i^{**}\|d_i'\|L_i'\|ID_i^{new}\|T_{id})$. Then GWN sends $< ZZ_i, Y_i >$ to the card and updates ID_i^{new} in the database.

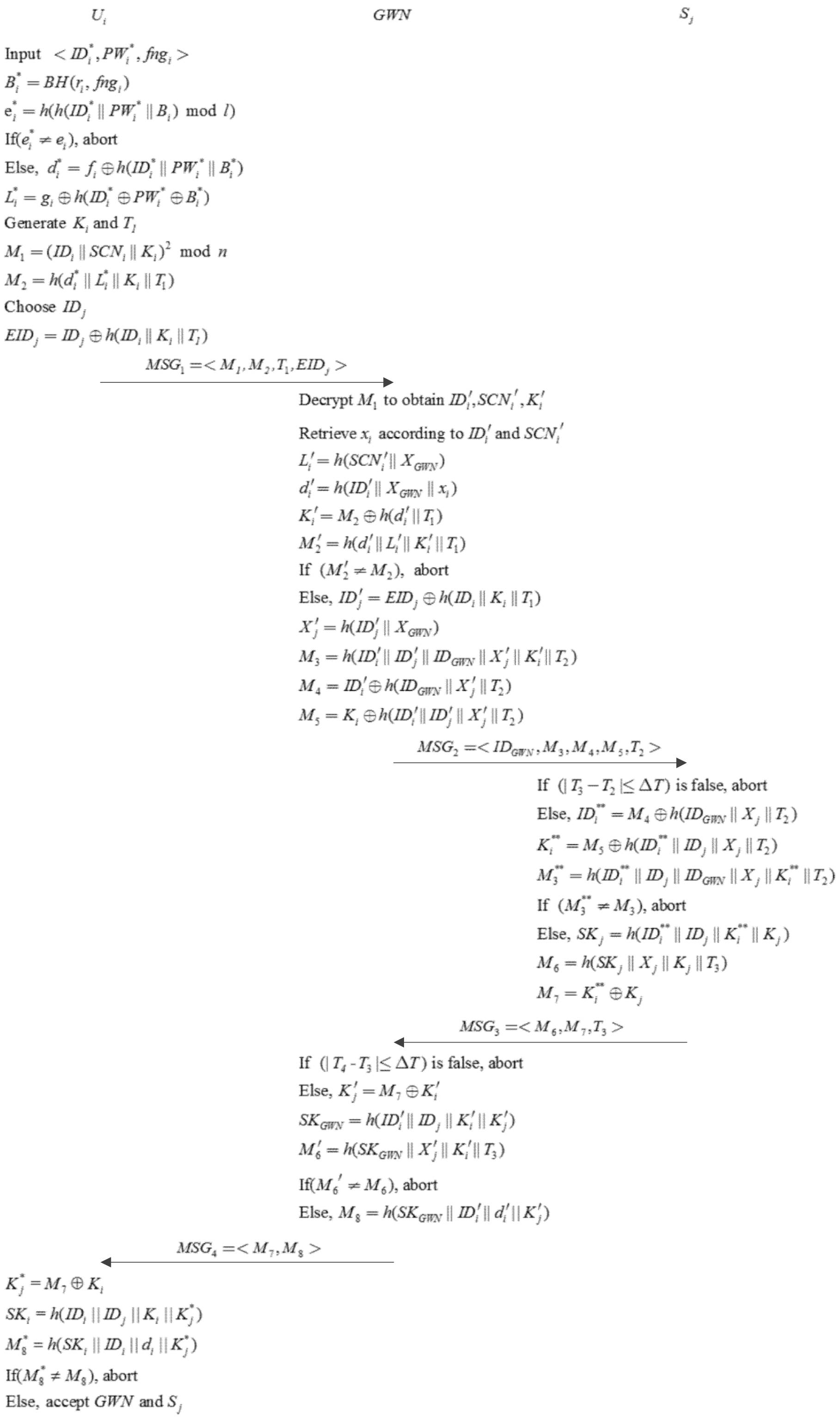
Step 3: The card computes $d_i^{**} = Y_i \oplus h(d_i^*\|T_{id})$ and $ZZ_i^* = h(d_i^{**}\|d_i^*\|L_i^*\|ID_i^{new}\|T_{id})$, and checks whether $ZZ_i^* = ZZ_i$ holds. If it holds, the card computes $e_i^{new} = h(h(ID_i^{new}\|PW_i\|B_i) \bmod l)$, $f_i^{new} = dd_i^{**} \oplus h(ID_i^{new}\|PW_i\|B_i)$, and $g_i^{new} = L_i \oplus h(ID_i^{new} \oplus PW_i \oplus B_i)$. Finally, the card replaces the old information with $< e_i^{new}, f_i^{new}, g_i^{new} >$.

F. PASSWORD CHANGE

In this phase, an authorized user U_i updates the password PW_i locally.

Step 1: U_i inserts his/her smart card into a card reader and carries out Step 1 of the login phase to verify the validity of fingerprint, password, and identity.

Step 2: U_i inputs a new password PW_i^{new}, and the card calculates $e_i^{new} = h(h(ID_i\|PW_i^{new}\|B_i) \bmod l)$, $d_i' = f_i \oplus$



U_i GWN S_j

Input $< ID_i^*, PW_i^*, fng_i >$
$B_i^* = BH(r_i, fng_i)$
$e_i^* = h(h(ID_i^* \| PW_i^* \| B_i) \bmod l)$
If$(e_i^* \neq e_i)$, abort
Else, $d_i^* = f_i \oplus h(ID_i^* \| PW_i^* \| B_i^*)$
$L_i^* = g_i \oplus h(ID_i^* \oplus PW_i^* \oplus B_i^*)$
Generate K_i and T_1
$M_1 = (ID_i \| SCN_i \| K_i)^2 \bmod n$
$M_2 = h(d_i^* \| L_i^* \| K_i \| T_1)$
Choose ID_j
$EID_j = ID_j \oplus h(ID_i \| K_i \| T_1)$

$MSG_1 =< M_1, M_2, T_1, EID_j >$

Decrypt M_1 to obtain ID_i', SCN_i', K_i'
Retrieve x_i according to ID_i' and SCN_i'
$L_i' = h(SCN_i' \| X_{GWN})$
$d_i' = h(ID_i' \| X_{GWN} \| x_i)$
$K_i' = M_2 \oplus h(d_i' \| T_1)$
$M_2' = h(d_i' \| L_i' \| K_i' \| T_1)$
If $(M_2' \neq M_2)$, abort
Else, $ID_j' = EID_j \oplus h(ID_i \| K_i \| T_1)$
$X_j' = h(ID_j' \| X_{GWN})$
$M_3 = h(ID_i' \| ID_j' \| ID_{GWN} \| X_j' \| K_i' \| T_2)$
$M_4 = ID_i' \oplus h(ID_{GWN} \| X_j' \| T_2)$
$M_5 = K_i \oplus h(ID_i' \| ID_j' \| X_j' \| T_2)$

$MSG_2 =< ID_{GWN}, M_3, M_4, M_5, T_2 >$

If $(|T_3 - T_2| \leq \Delta T)$ is false, abort
Else, $ID_i^{**} = M_4 \oplus h(ID_{GWN} \| X_j \| T_2)$
$K_i^{**} = M_5 \oplus h(ID_i^{**} \| ID_j \| X_j \| T_2)$
$M_3^{**} = h(ID_i^{**} \| ID_j \| ID_{GWN} \| X_j \| K_i^{**} \| T_2)$
If $(M_3^{**} \neq M_3)$, abort
Else, $SK_j = h(ID_i^{**} \| ID_j \| K_i^{**} \| K_j)$
$M_6 = h(SK_j \| X_j \| K_j \| T_3)$
$M_7 = K_i^{**} \oplus K_j$

$MSG_3 =< M_6, M_7, T_3 >$

If $(|T_4 - T_3| \leq \Delta T)$ is false, abort
Else, $K_j' = M_7 \oplus K_i'$
$SK_{GWN} = h(ID_i' \| ID_j \| K_i' \| K_j')$
$M_6' = h(SK_{GWN} \| X_j' \| K_i' \| T_3)$
If$(M_6' \neq M_6)$, abort
Else, $M_8 = h(SK_{GWN} \| ID_i' \| d_i' \| K_j')$

$MSG_4 =< M_7, M_8 >$

$K_j^* = M_7 \oplus K_i$
$SK_i = h(ID_i \| ID_j \| K_i \| K_j^*)$
$M_8^* = h(SK_i \| ID_i \| d_i \| K_j^*)$
If$(M_8^* \neq M_8)$, abort
Else, accept GWN and S_j

FIGURE 3. Authentication and key agreement of our proposed protocol.

$h(ID_i \| PW_i \| B_i)$, $f_i^{new} = d_i' \oplus h(ID_i \| PW_i^{new} \| B_i)$, $L_i' = g_i \oplus h(ID_i \oplus PW_i^{new} \oplus B_i)$, and $g_i^{new} = L_i' \oplus h(ID_i \oplus PW_i^{new})$.

Step 3: The card updates $< e_i, f_i, g_i >$ with $< e_i^{new}, f_i^{new}, g_i^{new} >$.

G. SMART CARD REVOCATION

If U_i's smart card is stolen or lost, U_i obtains a new smart card as follows.

Step 1: U_i sends the identity ID_i and his/her credential to SA through a secure channel. SA first verifies U_i's credential. If it is valid, it computes $d_i^{new} = h(ID_i \| X_{GWN} \| x_i)$ and $L_i^{new} = h(SCN_i^{new} \| X_{GWN})$, where SCN_i^{new} is the new smart card number. Then the new card storing $<d_i^{new}, L_i^{new}, SCN_i^{new}, l, n, BH(,), h()>$ is sent to U_i securely. Then, SA updates the database with SCN_i^{new}.

Step 2: U_i inserts the smart card into a card reader, inputs $< ID_i, PW_i >$ and imprints fng_i. The card picks up a random number r_i, computes $B_i = BH(r_i, fng_i)$, $e_i^{new} = h(h(ID_i \| PW_i \| B_i) \bmod l), f_i^{new} = d_i^{new} \oplus h(ID_i \| PW_i \| B_i)$, and $g_i^{new} = L_i^{new} \oplus h(ID_i \oplus PW_i \oplus B_i)$. Finally, the smart card stores $< e_i^{new}, f_i^{new}, g_i^{new}, SCN_i^{new}, l, n, r_i, BH(,), h() >$ into its memory and deletes $< d_i^{new}, L_i^{new} >$.

VI. SECURITY ANALYSIS

We first conduct a formal verification using ProVerif to demonstrate that our protocol fulfills the required security properties. Furthermore, we also present comprehensive heuristic security analysis and comparison.

A. FORMAL VERIFICATION WITH ProVerif

ProVerif [51] is a widely used formal verification tool for automatic security analysis of security protocols, which is used to prove the secrecy and authentication properties of our proposed protocol.

First we define the channels and types. c1 is the public channel between the user device and GWN and c2 is the public channel between GWN and the sensor.

```
free c1:channel.
free c2:channel.
```

The basic types of variables are defined as follows:

```
type key.
type nonce.
type fingerprint.
type timestamp.
type N.
type Q.
type P.
type User.
type Server.
type Sensor.
```

The cryptographic functions are modeled as follows:

```
(* Hash operation *)
fun Hash(bitstring): bitstring.
(* BH operation *)
fun BH(nonce, fingerprint): bitstring.
(* Rabin cryptosystem *)
fun rabinEnc(bitstring, N):bitstring.
reduc forall x: bitstring, p: P, q: Q, n: N;
rabinDec(n, p, q, rabinEnc(x, n)) = x.
(* XOR operation *)
fun XOR(bitstring, bitstring): bitstring.
reduc forall x: bitstring, y: bitstring;
XORagain(XOR(x, y), y) = x.
(* Mod operation *)
fun Mod(bitstring, bitstring): bitstring.
(* Concat operation *)
fun Concat(bitstring, bitstring):bitstring.
reduc forall x: bitstring, y: bitstring;
Split(Concat(x, y)) = (x, y).
(* Type convertion *)
fun timestamp2(timestamp): bitstring.
fun nonce2(nonce): bitstring.
fun key2(key): bitstring.
fun bit2key(bitstring): key.
(* Check timestamp fresh *)
fun checkFresh(timestamp, bool): bool
reduc forall t: timestamp;
checkFresh(t, true) = true
otherwise forall t: timestamp;
checkFresh(t, false) = false
```

The secret keys are defined as follows:

```
(* Secrecy assumptions *)
not attacker(new p).
not attacker(new q).
not attacker(new XGWNTemp).
not attacker(new XjTemp).
```

The following events and queries are defined:

```
event scAccept(User).
event serverAccept(User).
event sensorGen(User, Server).
event serverGen(Sensor).
event userGen(Server, Sensor).
query inj-event(userGen(server, sensor))
==> inj-event(serverGen(sensor)).
query inj-event(serverGen(sensor))
==> inj-event(sensorGen(user, server)).
query inj-event(sensorGen(user, server))
==> inj-event(serverAccept(user)).
query event(serverAccept(user))
==> event(scAccept(user)).
```

The process of the user is modeled as follows:

```
    let processUser(IDiEx: bitstring, PWEx: bitstring, fngEx:
fingerprint, e: bitstring, g: bitstring, f: bitstring, r: nonce, B:
bitstring, SCN: bitstring, l: bitstring, IDj: bitstring, IDi: bitstring) =
    let BEx = BH(r, fngEx) in
    let e' = Hash(Mod(Hash(Concat(Concat(IDiEx, PWEx), B)),
l)) in
    if e' = e then
    event scAccept(user);
    let dEx = XORagain(f,Hash(Concat(Concat(IDiEx, PWEx),
BEx))) in
    let LEx = XORagain(g, Hash(XOR(XOR(IDiEx, PWEx),
BEx))) in
    new KiTemp: nonce;
    let Ki = nonce2(KiTemp) in
    new T1: timestamp;
    let M1 = rabinEnc(Concat(Concat(IDi, SCN), Ki), n) in
    let M2 = Hash(Concat(Concat(Concat(dEx, LEx), Ki),
timestamp2(T1))) in
    let EIDj = XOR(IDj, Hash(Concat(Concat(IDi, Ki),
timestamp2(T1)))) in
    out(c1, (M1, M2, T1, EIDj));
    in(c1, (M7: bitstring, M8: bitstring));
```

```
let KjEx = XOR(M7, Ki) in
let SKi = bit2key(Hash(Concat(Concat(Concat(IDi, IDj), Ki), KjEx))) in
let M8Ex = Hash(Concat(Concat(Concat(key2(SKi), IDi), dEx), KjEx)) in
if M8Ex = M8 then
event userGen(server, sensor).
```

The process of the gateway node is modeled as follows:

```
let processServer(XGWN: bitstring, x: nonce, p: P, q: Q,
IDi: bitstring, IDGWN: bitstring, IDj: bitstring) =
in(c1, (M1: bitstring, M2: bitstring, T1: timestamp, EIDj: bitstring));
let (temp: bitstring, Ki’: bitstring) = Split(rabinDec(n, p, q, M1)) in
let (IDi’: bitstring, SCN’: bitstring) = Split(temp) in
let L’ = Hash(Concat(SCN’, XGWN)) in
let d’ = Hash(Concat(Concat(IDi’, XGWN), nonce2(x))) in
(* let Ki’ = XORagain(M2, Hash(Concat(d’, timestamp2(T1)))) in *)
let M2’ = Hash(Concat(Concat(Concat(d’, L’), Ki’), timestamp2(T1))) in
if M2’ = M2 then
event serverAccept(user);
let IDj’ = XORagain(EIDj, Hash(Concat(Concat(IDi, Ki’), timestamp2(T1)))) in
let Xj’ = Hash(Concat(IDj’, XGWN)) in
new T2: timestamp;
let M3 = Hash(Concat(Concat(Concat(Concat(Concat(IDi’, IDj’), IDGWN), Xj’), Ki’), timestamp2(T2))) in
let M4 = XOR(IDi’, Hash(Concat(Concat(IDGWN, Xj’), timestamp2(T2)))) in
let M5 = XOR(Ki’, Hash(Concat(Concat(Concat(IDi’, IDj’), Xj’), timestamp2(T2)))) in
out(c2, (IDGWN, M3, M4, M5, T2, true));
in(c2, (M6: bitstring, M7: bitstring, T3: timestamp, isFresh: bool));
new T4: timestamp;
if checkFresh(T4, isFresh) then
let Kj’ = XORagain(M7, Ki’) in
let SKGWN = bit2key(Hash(Concat(Concat(Concat(IDi’, IDj), Ki’), Kj’))) in
let M6’ = Hash(Concat(Concat(Concat(key2(SKGWN), Xj’), Ki’), timestamp2(T3))) in
if M6’ = M6 then
event serverGen(sensor);
let M8 = Hash(Concat(Concat(Concat(key2(SKGWN), IDi’), d’), Kj’)) in
out(c1, (M7, M8)).
```

The process of SNs is modeled as follows:

```
let processSensor(Xj: bitstring, IDj: bitstring) =
in(c2, (IDGWN: bitstring, M3: bitstring, M4: bitstring, M5: bitstring, T2: timestamp, isFresh: bool));
if checkFresh(T2, isFresh) then
let IDiExEx = XORagain(M4, Hash(Concat(Concat(IDGWN, Xj), timestamp2(T2)))) in
let KiExEx = XORagain(M5, Hash(Concat(Concat(Concat(IDiExEx, IDj), Xj), timestamp2(T2)))) in
let M3ExEx = Hash(Concat(Concat(Concat(Concat(Concat(IDiExEx, IDj), IDGWN), Xj), KiExEx), timestamp2(T2))) in
if M3ExEx = M3 then
new Kj: bitstring;
let SKj = bit2key(Hash(Concat(Concat(Concat(IDiExEx, IDj), KiExEx), Kj))) in
new T3: timestamp;
let M6 = Hash(Concat(Concat(Concat(key2(SKj), Xj), Kj), timestamp2(T3))) in
let M7 = XOR(KiExEx, Kj) in
event sensorGen(user, server);
out(c2, (M6, M7, T3, true)).
```

The whole protocol is modeled as follows.

```
(* Start process *)
process
(* Constants *)
(* Share constants between user and server *)
new SCN: bitstring;
new l: bitstring;
new IDj: bitstring;
new IDi: bitstring;
new PW: bitstring;
new r: nonce;
(* User/Smartcard constants *)
new fng: fingerprint;
(* Server constants *)
new x: nonce;
new XGWNTemp: key;
let XGWN = key2(XGWNTemp) in
new IDGWN: bitstring;
(* Sensor constants *)
new XjTemp: key;
let Xj = key2(XjTemp) in
(* Rabin parameters *)
new p: P;
new q: Q;
(* Constants computed *)
let d = Hash(Concat(Concat(IDi, XGWN), nonce2(x))) in
let L = Hash(Concat(SCN, XGWN)) in
let B = BH(r, fng) in
let e = Hash(Mod(Hash(Concat(Concat(IDi, PW), B)), l)) in
let f = XOR(d, Hash(Concat(Concat(IDi, PW), B))) in
let g = XOR(L, Hash(XOR(XOR(IDi, PW), B))) in
(
(!(processUser(IDi, PW, fng, e, g, f, r, B, SCN, l, IDj, IDi))) |
(!processServer(XGWN, x, p, q, IDi, IDGWN, IDj)) |
(!processSensor(Xj, IDj))
)
```

The outcome of executing the processes in ProVerif version 1.96 is given below. The results demonstrate that our protocol achieves session key secrecy and mutual authentication.

```
RESULT event(serverAccept(user[])) ==>
event(scAccept(user[])) is true.
RESULT inj-event(sensorGen(user[],server[])) ==>
inj-event(serverAccept(user[])) is true.
RESULT inj-event(serverGen(sensor[])) ==>
inj-event(sensorGen(user[],server[])) is true.
RESULT inj-event(userGen(server[],sensor[])) ==>
inj-event(serverGen(sensor[])) is true.
RESULT not attacker(SKj[]) is true.
RESULT not attacker(SKGWN[]) is true.
RESULT not attacker(SKi[]) is true.
```

B. ANALYSIS OF SECURITY PROPERTIES

We first show that our 3FA protocol could overcome weaknesses in Amin et al.'s authentication protocol, and then we show that our protocol achieves all the desired security features.

1) RESISTING TYPE I SCLA

Type I SCLA is infeasible in our protocol. We explain why below.

Assume that the opponent A extracts the smart card information $< e_i, f_i, g_i, SCN_i, l, n, r_i, BH(\cdot,\cdot), h() >$ of the legal user U_i, where $B_i = BH(r_i, fng_i)$, $e_i = h(h(ID_i\|PW_i\|B_i) \bmod l)$, $f_i = d_i \oplus h(ID_i\|PW_i\|B_i)$, and $g_i = L_i \oplus h(ID_i \oplus PW_i \oplus B_i)$. Then A can guess ID_i^* and PW_i^*, and computes $e_i^* = h(h(ID_i^*\|PW_i^*\|B_i) \bmod l)$, as presented in Section 4.1. However, A cannot verify the correctness of ID_i^* and PW_i^* definitely because e_i is a "fuzzy verifier" [59], [60].

Therefore, our protocol is secure against type I SCLA.

2) RESISTING TYPE II SCLA

Moreover, type II SCLA is also infeasible in our protocol.

Suppose A could also intercept the message $MSG_1 =< M_1, M_2, T_1, EID_j >$ sent by U_i in the login phase, where $d_i^* = f_i \oplus h(ID_i^*\|PW_i^*\|B_i^*)$, $L_i^* = g_i \oplus h(ID_i^* \oplus PW_i^* \oplus B_i^*)$, $M_1 = (ID_i\|SCN_i\|K_i)^2 \bmod n$, $M_2 = h(d_i^*\|L_i^*\|K_i\|T_1)$. A can derive $d_i^* = f_i \oplus h(ID_i^*\|PW_i^*\|B_i^*)$, $L_i^* = g_i \oplus h(ID_i^* \oplus PW_i^* \oplus B_i^*)$, where g_i is revealed from U_i's smart card. Due to the hardness of quadratic residue problem, it is impossible for the adversary to compute R_1 from the value $M_1 = (ID_i\|SCN_i\|K_i)^2 \bmod n$. Therefore, A is unable to calculate $M_2^* = h(d_i^*\|L_i^*\|K_i\|T_1)$, which is a necessary to check the correctness of ID_i^* and PW_i^*.

Thus, our 3FA protocol is completely secure against type II SCLA.

3) RESISTING KSSTIA

In Amin et al.'s protocol, the static value $h(ID_i'\|X_j')$ is used to protect the ephemeral random numbers, where X_j' is the sensor key shared between S_j and GW. As a result, the disclosure of ephemeral random number K_i will lead to the compromise of the static value $h(ID_i'\|X_j')$, which in turn will cause the compromise of ephemeral random numbers in other authentication sessions. In our proposed protocol, we avoid this risk by introducing the mechanisms of timestamp and hash. Specifically, we compute $M_5 = K_i \oplus h(ID_i'\|ID_j'\|X_j'\|T_2)$. In this case, even though K_i is compromised, the opponent can only obtain the hashed value $h(ID_i'\|ID_j'\|X_j'\|T_2)$, which is dynamic in each authentication session and will not endanger the ephemeral random number in other authentication sessions. Thus our 3FA protocol is immune from KSSTIA.

4) RESISTING USER IMPERSONATION ATTACK

The opponent A cannot carry out user impersonation attack against our protocol. Assume that A has the user U_i's smart card and has extracted the data $< e_i, f_i, g_i, SCN_i, l, n, r_i, BH(\cdot,\cdot), h() >$ stored in it. We also assume that A has intercepted the messages exchanged in the previous authentication sessions. In our protocol, A has to possess all the authentication factors, i.e., PW_i, the smart card, and the biometric, to produce a legal message $MSG_1 =< M_1, M_2, T_1, EID_j >$. Specifically, the key to proving the legitimacy of U_i is the value $M_2 = h(d_i^*\|L_i^*\|K_i\|T_1)$. The critical fields of the computation of M_2 are the values $d_i^* = f_i \oplus h(ID_i^*\|PW_i^*\|B_i^*)$ and $L_i^* = g_i \oplus h(ID_i^* \oplus PW_i^* \oplus B_i^*)$. However, without either PW_i, the smart card, or the biometric, A cannot calculate d_i^* or L_i^*.

5) RESISTING GATEWAY IMPERSONATION ATTACK

In our protocol, the opponent A is unable to impersonate as GWN to either U_i or S_j. In order to impersonate as GWN to S_j, A needs to compute a legal value $M_3 = h(ID_i'\|ID_j'\|ID_{GWN}\|X_j'\|K_i'\|T_2)$. However, without knowing the value $X_j' = h(ID_j'\|X_{GWN})$, it is infeasible for A to compute M_3. Moreover, since we use the hash algorithm and timestamp, A cannot obtain any useful information from the messages from the previous authentication sessions.

In contrast, to impersonate as GWN to either U_i, A needs to compute a legal value $M_8 = h(SK_{GWN}\|ID_i'\|d_i'\|K_j')$. To do so, A needs to have knowledge of K_i to compute the value $SK_{GWN} = h(ID_i\|ID_j\|K_i\|K_j)$. To get K_i, A has to know the secret key p and q of GW. It is impossible because the secret key is carefully protected by the administrator. The other way left for A is to decrypt the value $M_1 = (ID_i\|SCN_i\|K_i)^2 \bmod n$, which is computationally infeasible because of the hardness of quadratic residue problem. Thus, the protocol can withstand the gateway node impersonation attack.

6) RESISTING SN IMPERSONATION ATTACK

Suppose A tries to impersonate S_j after capturing the messages exchanged in the previous authentication sessions. A needs to generate $MSG_3 =< M_6, M_7, T_3 >$ to impersonate S_j, where $SK_j = h(ID_i^{**}\|ID_j\|K_i^{**}\|K_j)$, $M_6 = h(SK_j\|X_j\|K_j\|T_3)$, and $M_7 = K_i^{**} \oplus K_j$. Thus, A has to know K_i in order to compute $M_6 = h(SK_j\|X_j\|K_j\|T_3)$. Similar to the analysis of gateway impersonation attack, A is unable to obtain K_i. Thus A cannot carry out the SN impersonation attack.

7) RESISTING MODIFICATION ATTACK

In our protocol, the opponent A is unable to modify any of the messages $MSG_1 =< M_1, M_2, T_1, EID_j >$, $MSG_2 =< ID_{GWN}, M_3, M_4, M_5, T_2 >$, $MSG_3 =< M_6, M_7, T_3 >$, or $MSG_4 =< M_7, M_8 >$. Assume that A intercepts one of these messages, and then transmits a modified one. However, each message is protected by a hash value computed with a secret value. For instance, in MSG_1, A cannot calculate $M_2 = h(d_i^*\|L_i^*\|K_i\|T_1)$, since $d_i^* = f_i \oplus h(ID_i^*\|PW_i^*\|B_i^*)$ and $L_i^* = g_i \oplus h(ID_i^* \oplus PW_i^* \oplus B_i^*)$ are secret values which cannot be computed without knowing either PW_i, the smart card, or the biometric. Any modification will be detected by the receiver of the message who will check the correctness of the hash value in each message. Hence, our protocol is secure against modification attacks.

8) RESISTING REPLAY ATTACK

In our protocol, A may attempt to replay old messages sent by the entities. However, the timestamp mechanism and the challenge-response mechanism are used in all the messages involved to resist replay attacks. Specifically, $MSG_1 =< M_1, M_2, T_1, EID_j >$, $MSG_2 =< ID_{GWN}, M_3, M_4, M_5, T_2 >$ and $MSG_3 =< M_6, M_7, T_3 >$ are protected by a hash value which is computed with a shared secret between the sender and receiver. As a result, A cannot bypass the timestamp. If A would replay a previous message, it will be detected by the receiver instantly through checking the timestamp and the hash value.

On the other hand, $MSG_4 =< M_7, M_8 >$ contains a challenge K_i, which is chosen by U_i. Additionally, these two messages are also protected by a hash value computed with K_i. Thus, A cannot bypass the challenge-response mechanism. Then GW and U_i could discover message replay by validating the freshness of K_i.

Thus, our 3FA protocol could defend against replay attacks.

9) RESISTING PRIVILEGED INSIDER ATTACK

In practice, users may register across different information systems with the same password. If a privileged insider may somehow obtain the password of the user, he/she can use it to impersonate as this user to access the services of other systems. In our protocol, U_i only submits ID_i during the registration phase. As a result, an insider cannot obtain U_i's password. Hence, our protocol can withstand privileged insider attack.

10) RESISTING STOLEN VERIFIER ATTACK

In this attack, an opponent steals the verification information (e.g., plaintext or hashed passwords) stored in the server. In our protocol, the server maintains a database storing $< ID_i, SCN_i, x_i,$ Personal credential $>$, which has no information related to the password. Thus, stolen verifier attack is not possible in our protocol.

11) MUTUAL AUTHENTICATION

An adversary cannot generate legal $M_2 = h(d_i^*\|L_i^*\|K_i\|T_1)$ without knowing U_i's private key d_i^* and L_i^*. So GWN can authenticate U_i by verifying the correctness of M_2. Similarly, U_i can authenticate GWN by verifying the correctness of $M_8 = h(SK_{GWN}\|ID_i'\|d_i'\|K_j')$. Hence, U_i and GWN are mutually authenticated.

On the other hand, S_j authenticates GWN by verifying the correctness of $M_3 = h(ID_i'\|ID_j'\|ID_{GWN}\|X_j'\|K_i'\|T_2)$. At the same time, GWN could authenticate S_j by verifying the correctness of $M_6 = h(SK_j\|X_j\|K_j\|T_3)$. Hence, our 3FA protocol also achieves mutual authentication between GWN and S_j.

12) SESSION KEY AGREEMENT

In a successful authentication session, the session key $SK = h(ID_i\|ID_j\|K_i\|K_j)$ is established between U_i and S_j to protect future communication. It is worth pointing out that the secrecy of SK is dependent on the secrecy of the random numbers involved. All these values are carefully protected by the secret values shared between the participants in each message.

Suppose the session key $SK = h(ID_i\|ID_j\|K_i\|K_j)$ of one session is disclosed to the opponent A. However, he/she cannot compute any of the past and future session keys by using SK because the session key is protected by $h()$ and the random numbers $< K_i, K_j >$ are different in each session. As a result, our 3FA protocol achieves session key agreement and known key security.

13) USER ANONYMITY

Privacy is of increasing importance in the IoT and cloud computing era [65]–[70]. Suppose the opponent A first captures all the messages transmitted between the participants during the protocol execution and then tries to guess the identity of the user. In our proposed protocol, U_i's identity ID_i is included in the field $M_1 = (ID_i\|SCN_i\|K_i)^2 \bmod n$ in the first message. To get ID_i, A has to know the secret key (p, q) of GWN. It is impossible to do so because the secret key is carefully protected by the administrator. The other way left for A to obtain ID_i is to decrypt the value $M_1 = (ID_i\|SCN_i\|K_i)^2 \bmod n$, which is computationally infeasible due to the hardness of quadratic residue problem. Hence, our 3FA protocol achieves user anonymity.

14) USER UNTRACEABILITY

To track a user, A captures these messages involved in different authentication sessions and checks whether they have the same field to learn whether the same user are involved. However, A cannot trace U_i by capturing the authentication messages. Assume that A intercepts $MSG_1 =< M_1, M_2, T_1, EID_j >$, $MSG_2 =< ID_{GWN}, M_3, M_4, M_5, T_2 >$, $MSG_3 =< M_6, M_7, T_3 >$, and $MSG_4 =< M_7, M_8 >$. We note that the computation of each field involves the timestamp and a random number which are different in each session. As a result, the messages of each session are also different. Therefore, our protocol resists user tracking attacks and achieves user untraceability.

15) BIOMETRIC TEMPLATE PRIVACY

Biometric template privacy is preserved in our protocol. First, the user provides no biometric templates to the server, and the server stores no information related to the user's biometric template. Second, the biometric information is first converted by the biohashing algorithm and then protected by the hash function. Since these two mechanisms are both one-way operation, the information stored in the smart card will not leak

TABLE 2. Comparison of security features.

	Wu et al.'s protocol [48]	Amin et al.'s protocol [39]	Our protocol
Resisting type I SCLA	×	×	✓
Resisting type II SCLA	×	×	✓
Resisting KSSTIA	✓	×	✓
Resisting user impersonation attack	✓	✓	✓
Resisting gateway impersonation attack	✓	✓	✓
Resisting SN impersonation attack	✓	✓	✓
Resisting modification attack	✓	✓	✓
Resisting replay attack	✓	✓	✓
Resisting privileged insider attack	✓	✓	✓
Resisting stolen-verifier attack	✓	✓	✓
Mutual authentication	✓	✓	✓
Secure key agreement	✓	✓	✓
User anonymity	✓	✓	✓
User untraceability	✓	×	✓
Biometric template privacy	✓	✓	✓
Smart card revocation	×	✓	✓

TABLE 3. Efficiency comparisons.

	U_i	GWN	S_j	Total
Das's protocol [45]	$9T_H$	$11T_H$	$5T_H$	$25T_H$
Das et al.'s protocol [38]	$12T_H + 2T_{ECM}$	$10T_H$	$9T_H + 2T_{ECM}$	$31T_H + 4T_{ECM}$
Wu et al.'s protocol [48]	$11T_H + 2T_{ECM}$	$10T_H$	$3T_H + 2T_{ECM}$	$24T_H + 4T_{ECM}$
Li et al.'s protocol [49]	$6T_H + 2T_S$	$7T_H + 6T_S$	$5T_H + 2T_S$	$18T_H + 10T_S$
Amin et al.'s protocol [39]	$12T_H$	$15T_H$	$5T_H$	$32T_H$
Our protocol	$8T_H + 1T_M$	$12T_H + 1T_{QR}$	$5T_H$	$25T_H + T_M + T_{QR}$

biometrics. Therefore, biometric template privacy is achieved in our protocol.

16) SMART CARD AND USER REVOCATION

In our scheme, a database storing the user identity and smart card number is maintained, through which the invalid smart card will be detected. Thus, lost/stolen smart card can be revoked by removing the card number from the database.

C. COMPARISON OF SECURITY FEATURES

In Table 2, we present the comparison of our 3FA protocol with the ones in [39] and [48].

From Table 2, we note that both Wu et al.'s protocol and Amin et al.'s protocol are susceptible to several attacks, e.g., SCLA. Wu et al.'s protocol cannot provide smart card revocation. Amin et al.'s protocol is prone to KSSTIA and cannot provide user untraceability. Table 2 shows that our new protocol is the only one that is free from security attacks and provides the required features.

VII. EFFICIENCY ANALYSIS

We evaluate the efficiency of our new protocol and compare it with other protocols. Since SNs are constrained in terms of critical resources such as memory, processing power and energy, special attention must be given to the computation cost of security protocols for WSN [71].

In Table 3, we summarize the computational time of our new protocol and the related ones in [38], [39], [45], [48], and [49]. We focus only on the login and authentication phase and ignore the bit XOR operation because it requires very low computation. T_H, T_S, T_M, T_{QR}, T_{ECM} denote the cost for executing the hash, the symmetric encryption/decryption, the modular squaring, the computation of a square root modulo N, and ECC point multiplication respectively. It is worth noting that the modular squaring is as efficient as the hash operation while the computation of a square root modulo N is similar to modular exponentiation.

Table 3 shows the results of the comparison. Our protocol is as efficient as the most efficient one of these previously proposed protocols at the mobile device and SNs. Although the computation cost for the gateway of our proposed scheme is higher than that of Amin et al.'s protocol and Das's protocol, generally it is not a concern, because the gateway is powerful and has no resource constraints. Moreover, Das's protocol and Amin et al.'s protocol [39] is prone to SCLA.

VIII. CONCLUSION

We have analyzed the three-factor mutual authentication protocol of Amin et al. and we have shown its security drawbacks. The protocol of Amin et al. suffers from Type I SCLA and Type II SCLA. In particular, the user identity and password can be exhaustively guessed in an offline manner with the secrets stored in the stolen smart card and the intercepted authentication messages. Furthermore, the protocol suffers from KSSTIA when the temporal parameters in an authentication session are disclosed. Finally, the protocol is prone to tracking attack and fails to fulfill user untraceability.

Next, we have presented a lightweight and secure three-factor authentication protocol based on Rabin cryptosystem. We conducted a formal verification of the proposed protocol by using ProVerif to demonstrate that it fulfills the required security features. Furthermore, we also present a comprehensive heuristic security analysis to demonstrate that our protocol is capable of withstanding all the possible active and passive attacks including addressing the weaknesses revealed in the protocol of Amin et al., and we further show that our proposed protocol support all the desired security features. A performance analysis of our proposed protocol shows that it can be deployed in practice for Internet-integrated WSN, while achieving a balance between security and efficiency.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their constructive comments which helped them improve the quality and presentation of this paper.

REFERENCES

[1] S. Hong *et al.*, "SNAIL: An IP-based wireless sensor network approach to the Internet of Things," *IEEE Wireless Commun.*, vol. 17, no. 6, pp. 34–42, Dec. 2010.
[2] R. Roman, C. Alcaraz, J. Lopez, and N. Sklavos, "Key management systems for sensor networks in the context of the Internet of Things," *Comput. Elect. Eng.*, vol. 37, no. 2, pp. 147–159, Mar. 2011.
[3] J. Granjal, E. Monteiro, and J. S. Silva, "Security in the integration of low-power Wireless Sensor Networks with the Internet: A survey," *Ad Hoc Netw.*, vol. 24, pp. 264–287, Jan. 2015.
[4] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. McCann, and K. Leung, "A survey on the IETF protocol suite for the Internet of Things: Standards, challenges, and opportunities," *IEEE Wireless Commun.*, vol. 20, no. 6, pp. 91–98, Dec. 2013.
[5] *6LoWPAN Working Group*. Accessed on Jan. 15, 2017. [Online]. Available: http://tools.ietf.org/wg/6lowpan/
[6] *ROLL Working Group*. Accessed on Jan. 15, 2017. [Online]. Available: http://tools.ietf.org/wg/roll/
[7] R. Roman and J. Lopez, "Integrating wireless sensor networks and the Internet: A security analysis," *Internet Res.*, vol. 19, no. 2, pp. 246–259, 2009.
[8] J. Astorga, E. Jacob, N. Toledo, and J. Unzilla, "Enhancing secure access to sensor data with user privacy support," *Comput. Netw.*, vol. 64, pp. 159–179, May 2014.
[9] J. Qi, X. Hu, Y. Ma, and Y. Sun, "A hybrid security and compressive sensing-based sensor data gathering scheme," *IEEE Access*, vol. 3, pp. 718–724, 2015.
[10] Z. Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu, "Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Trans. Commun.*, vol. E98-B, no. 1, pp. 190–200, 2015.
[11] Z. Fu, F. Huang, X. Sun, A. V. Vasilakos, and C.-N. Yang, "Enabling semantic search based on conceptual graphs over encrypted outsourced data," *IEEE Trans. Serv. Comput.*, to be published, doi: 10.1109/TSC.2016.2622697.
[12] H. Li, D. Liu, Y. Dai, and T. H. Luan, "Engineering searchable encryption of mobile cloud networks: When QoE meets QoP," *IEEE Wireless Commun.*, vol. 22, no. 4, pp. 74–80, Aug. 2015.
[13] D. He, S. Zeadally, N. Kumar, and J.-H. Lee, "Anonymous authentication for wireless body area networks with provable security," *IEEE Syst. J.*, to be published, doi: 10.1109/JSYST.2016.2544805.
[14] D. He and S. Zeadally, "Authentication protocol for an ambient assisted living system," *IEEE Commun. Mag.*, vol. 35, no. 1, pp. 71–77, Jan. 2015.
[15] K. T. Nguyen, M. Laurent, and N. Oualha, "Survey on secure communication protocols for the Internet of Things," *Ad Hoc Netw.*, vol. 32, pp. 17–31, Sep. 2015.
[16] S. Raza, S. Duquennoy, A. Chung, D. Yazar, T. Voigt, and U. Roedig, "Securing communication in 6LoWPAN with compressed IPsec," in *Proc. 7th Int. Conf. DCOSS*, Jun. 2011, pp. 1–8.
[17] S. Ray and G. Biswas, "Establishment of ECC-based initial secrecy usable for IKE implementation," in *Proc. World Congr. Eng. (WCE)*, vol. 1. London, U.K., Jul. 2012, pp. 1–6.
[18] S. Kumari, M. K. Khan, and M. Atiquzzaman, "User authentication schemes for wireless sensor networks: A review," *Ad Hoc Netw.*, vol. 27, pp. 159–194, Apr. 2015.
[19] Q. Jiang, J. Ma, X. Lu, and Y. Tian, "An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks," *Peer-Peer Netw. Appl.*, vol. 8, no. 6, pp. 1070–1081, Nov. 2014.
[20] D. He, N. Kumar, and N. Chilamkurti, "A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks," *Inf. Sci.*, vol. 321, pp. 263–277, Nov. 2015.
[21] H. Xiong, "Cost-effective scalable and anonymous certificateless remote authentication protocol," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2327–2339, Dec. 2014.
[22] H. Xiong and Z. Qin, "Revocable and scalable certificateless remote authentication protocol with anonymity for wireless body area networks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 7, pp. 1442–1455, Jul. 2015.
[23] J. Shen, H. Tan, S. Moh, I. Chung, Q. Liu, and X. Sun, "Enhanced secure sensor association and key management in wireless body area networks," *J. Commun. Netw.*, vol. 17, no. 5, pp. 453–462, Oct. 2015.
[24] F. Li and P. Xiong, "Practical secure communication for integrating wireless sensor networks into the Internet of Things," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3677–3684, Oct. 2013.
[25] J. Astorga, E. Jacob, M. Huarte, and M. Higuero, "Ladon^1: End-to-end authorisation support for resource-deprived environments," *IET Inf. Secur.*, vol. 6, no. 2, pp. 93–101, Jun. 2012.
[26] Q. Jiang, J. Ma, G. Li, and X. Li, "Improvement of robust smart-card-based password authentication scheme," *Int. J. Commun. Syst.*, vol. 28, no. 2, pp. 383–393, Jan. 2015.
[27] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1086–1090, Mar. 2009.
[28] S. Kumari *et al.*, "A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps," *Future Generat. Comput. Syst.*, vol. 63, pp. 56–75, Oct. 2016.
[29] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, and Y. Yang, "An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 76, pp. 37–48, Dec. 2016.
[30] Q. Jiang, N. Kumar, J. Ma, J. Shen, D. He, and N. Chilamkurti, "A privacy-aware two-factor authentication protocol based on elliptic curve cryptography for wireless sensor networks," *Int. J. Netw. Manage.*, 2016, doi: 10.1002/nem.1937.
[31] F. Wei, J. Ma, Q. Jiang, J. Shen, and C. Ma, "Cryptanalysis and improvement of an enhanced two-factor user authentication scheme in wireless sensor networks," *Inf. Technol. Control*, vol. 45, no. 1, pp. 62–70, 2016.
[32] M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Netw.*, vol. 20, pp. 96–112, Sep. 2014.

[33] R. Amin and G. P. Biswas, "A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," *Ad Hoc Netw.*, vol. 36, pp. 58–80, Jan. 2016.

[34] M. S. Farash, M. Turkanović, M. Kumari, and S. Hölb, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment," *Ad Hoc Netw.*, vol. 36, pp. 152–176, Jan. 2016.

[35] C.-C. Chang and H. D. Le, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 357–366, Jan. 2016.

[36] Y. Lu, L. Li, H. Peng, and Y. Yang, "An energy efficient mutual authentication and key agreement scheme preserving anonymity for wireless sensor networks," *Sensors*, vol. 16, no. 6, 2016, Art. no. 837.

[37] F. Wu *et al.*, "An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment," *J. Netw. Comput. Appl.*, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2016.12.008

[38] A. K. Das *et al.*, "Provably secure user authentication and key agreement scheme for wireless sensor networks," *Secur. Commun. Netw.*, vol. 9, no. 16, pp. 3670–3687, Nov. 2016.

[39] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, L. Leng, and N. Kumar, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Comput. Netw.*, vol. 101, pp. 42–62, Jun. 2016.

[40] D. He and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment," *IEEE Syst. J.*, vol. 9, no. 3, pp. 816–823, Sep. 2015.

[41] X. Li, J. Niu, Z. Wang, and C. Chen, "Applying biometrics to design three-factor remote user authentication scheme with key agreement," *Secur. Commun. Netw.*, vol. 7, no. 10, pp. 1488–1497, Oct. 2014.

[42] Q. Jiang, F. Wei, S. Fu, J. Ma, G. Li, and A. Alelaiwi, "Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy," *Nonlinear Dyn.*, vol. 83, no. 4, pp. 2085–2101, Mar. 2016.

[43] Q. Jiang, M. K. Khan, X. Lu, J. Ma, and D. He, "A privacy preserving three-factor authentication protocol for e-health clouds," *J. Supercomput.*, vol. 72, no. 10, pp. 3826–3849, Oct. 2016.

[44] Q. Jiang, J. Ma, and F. Wei, "On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Syst. J.*, to be published, doi: 10.1109/JSYST.2016.2574719.

[45] A. K. Das, "A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks," *Peer-Peer Netw. Appl.*, vol. 9, no. 1, pp. 223–244, Jan. 2014.

[46] A. K. Das, "A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor," *Int. J. Commun. Syst.*, vol. 30, no. 1, p. e2933, Jan. 2017, doi: 10.1002/dac.2933.

[47] A. K. Das, "A secure and efficient user anonymity-preserving three-factor authentication protocol for large-scale distributed wireless sensor networks," *Wireless Pers. Commun.*, vol. 82, no. 3, pp. 1377–1404, Jun. 2015.

[48] F. Wu, L. Xu, S. Kumari, and X. Li, "An improved and provably secure three-factor user authentication scheme for wireless sensor networks," *Peer-Peer Netw. Appl.*, 2016, doi: 10.1007/s12083-016-0485-9.

[49] X. Li, J. Niu, S. Kumari, J. Liao, W. Liang, and M. K. Khan, "A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity," *Secur. Commun. Netw.*, vol. 9, no. 15, pp. 2643–2655, Oct. 2016.

[50] A. K. Das, A. K. Sutrala, V. Odelu, and A. Goswami, "A secure smartcard-based anonymous user authentication scheme for healthcare applications using wireless medical sensor networks," *Wireless Pers. Commun.*, 2016, doi: 10.1007/s11277-016-3718-6.

[51] B. Blanchet, "An efficient cryptographic protocol verifier based on prolog rules," in *Proc. IEEE Comput. Soc. Found. (CSFW)*, Jun. 2001, pp. 82–96.

[52] I. Natgunanathan, A. Mehmood, Y. Xiang, G. Beliakov, and J. Yearwood, "Protection of privacy in biometric data," *IEEE Access*, vol. 4, pp. 880–892, 2016.

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

[54] A. Juels and M. Sudan, "A fuzzy vault scheme," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun./Jul. 2002, p. 408.

[55] A. T. B. Jin, D. N. C. Ling, and A. Goh, "BioHashing: Two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognit.*, vol. 37, no. 11, pp. 2245–2255, Apr. 2004.

[56] R. Lumini and L. Nanni, "An improved BioHashing for human authentication," *Pattern Recognit.*, vol. 40, no. 3, pp. 1057–1065, Mar. 2007.

[57] M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," MIT Lab. Comput. Sci., Cambridge, MA, USA, Tech. Rep., 1979.

[58] H.-Y. Chien, "Combining Rabin cryptosystem and error correction codes to facilitate anonymous authentication with un-traceability for low-end devices," *Comput. Netw.*, vol. 57, no. 14, pp. 2705–2717, Oct. 2013.

[59] D. Wang, D. He, P. Wang, and C.-H. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Trans. Depend. Sec. Comput.*, vol. 12, no. 4, pp. 428–442, Jul./Aug. 2015.

[60] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE Trans. Depend. Sec. Comput.*, to be published, doi: 10.1109/TDSC.2016.2605087.

[61] D. Wang, Q. Gu, H. Cheng, and P. Wang, "The request for better measurement: A comparative evaluation of two-factor authentication schemes," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur.*, May 2016, pp. 475–486.

[62] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Adv. Cryptol.*, Santa Barbara, CA, USA, Aug. 1999, pp. 388–397.

[63] 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.

[64] M. Abdalla, F. Benhamouda, and P. MacKenzie, "Security of the J-PAKE password-authenticated key exchange protocol," in *Proc. IEEE S&P*, May 2015, pp. 571–587.

[65] A. G. Reddy, A. K. Das, E. J. Yoon, and K.-Y. Yoo, "A secure anonymous authentication protocol for mobile services on elliptic curve cryptography," *IEEE Access*, vol. 4, pp. 4394–4407, 2016.

[66] Z. Fu, X. Sun, S. Ji, and G. Xie, "Towards efficient content-aware search over encrypted outsourced data in cloud," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (IEEE INFOCOM)*, San Francisco, CA, USA, Apr. 2016, pp. 1–9, doi: 10.1109/INFOCOM.2016.7524606.

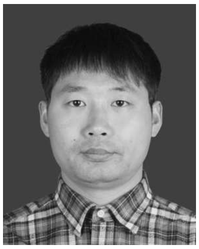
[67] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 12, pp. 2706–2716, Dec. 2016.

[68] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 2, pp. 340–352, Jan. 2016.

[69] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 11, pp. 2594–2608, Nov. 2016.

[70] S. Zeadally and M. Badra, Eds., *Privacy in a Digital, Networked World: Technologies, Implications and Solutions*. London, U.K.: Springer, Oct. 2015.

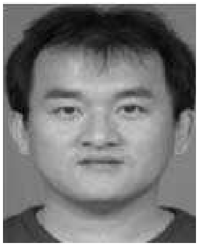
[71] Z. Liu, H. Seo, J. Großschädl, and H. Kim, "Efficient implementation of NIST-compliant elliptic curve cryptography for 8-bit AVR-based sensor nodes," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 7, pp. 1385–1397, Jul. 2016.



QI JIANG received the B.S. degree in computer science from Shaanxi Normal University in 2005, and the Ph.D. degree in computer science from Xidian University in 2011. He is currently an Associate Professor with the School of Cyber Engineering, Xidian University. His research interests include security protocols, wireless network security, and cloud security.

SHERALI ZEADALLY received the bachelor's degree in computer science from the University of Cambridge, U.K., and the Ph.D. degree in computer science from the University of Buckingham, U.K. He is currently an Associate Professor with the College of Communication and Information, University of Kentucky, Lexington, KY, USA. He is a fellow of the British Computer Society and the Institution of Engineering Technology, U.K.

JIANFENG MA received the M.S. degree in computer science and the Ph.D. degree from Xidian University, in 1992 and 1995, respectively. He is currently a Professor with the School of Cyber Engineering, Xidian University. He has published over 150 journal and conference papers. His research interests include applied cryptography, wireless network security, data security, and mobile security.



DEBIAO HE received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, in 2009. He is currently an Associate Professor with the State Key Lab of Software Engineering, Computer School, Wuhan University. His main research interests include cryptography and information security, in particular, cryptographic protocols.