Journal of Information Security and Applications 52 (2020) 102494

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

A three-factor anonymous user authentication scheme for Internet of Things environments

Hakjun Lee [a], Dongwoo Kang [a], Jihyeon Ryu [b], Dongho Won [c], Hyoungshick Kim [c], Youngsook Lee [d,*]

[a] Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, Gyeonggi-do, South Korea
[b] Department of Software, Sungkyunkwan University, Suwon, Gyeonggi-do, South Korea
[c] Department of Computer Engineering, Sungkyunkwan University, Suwon, Gyeonggi-do, South Korea
[d] Department of Cyber Security Department, Howon University, Gunsan, Jeollabuk-do, South Korea

ARTICLE INFO

Article history:
Available online 7 April 2020

Keywords:
Authentication
Key agreement
Wireless sensor network
Internet of things

ABSTRACT

To accelerate the deployment of fifth-generation (5G) cellular networks, millions of devices are being connected to massive Internet of Things (IoT) networks. However, advances in the scale of connectivity on 5G networks may increase the attack surface of these devices, thereby increasing the number of attack opportunities. To address the potential security risks in IoT systems, one feasible security practice involves the development of secure and efficient user authentication schemes. In 2017, Dhillon and Kalra proposed a three-factor user authentication scheme for IoT. We noted that their scheme suffers from several security weaknesses. In this study, we specifically demonstrate that the scheme proposed by Dhillon and Kalra (1) is not secured from a stolen mobile device attack; (2) does not prevent a user impersonation attack; (3) does not provide a session key agreement; (4) does not have a contingency plan (e.g., a revocation phase) for situations where a user's private key is compromised, or a mobile device is stolen or lost. We propose an improved three-factor user authentication scheme to resolve these security issues. Furthermore, we demonstrate that the proposed scheme provides desirable attributes for IoT environments and that its computation and communication costs are suitable for extremely low-cost IoT devices.

1. Introduction

The Internet of Things (IoT) is composed of resource-constrained nodes, and these densely scattered nodes in IoT environments provide continuous service, irrespective of time and location. Currently, IoT has been adopted for many applications, including healthcare, smart home, smart factory, and smart city. Furthermore, the advent of the fifth-generation (5G) cellular network and its commercialization has birthed the anticipation of a hyper-linked network to connect and share information not only between individual portable terminals but also between most (if not all) the objects we use in daily life. According to a study conducted by Park et al. [1], by the year 2020, approximately 50 billion sensor devices across the world will be connected to IoT networks, and the number of these devices is expected to increase exponentially with the commercialization of 5G networks. According to the 5G vision requirements of the International Telecommunication Union Radio Communication Standards Sector (ITU-R) [2], a massive IoT network accommodates approximately 1 million objects per km^2 (1 per m^2).

The development of IoT and massive IoT has tremendous potential, but these environments expose devices to a wide range of vulnerabilities due to an increased attack surface. Therefore, to protect user privacy in IoT environments, security properties such as (1) data security, (2) virtual network security, (3) service availability, and (4) data integrity must be provided [3]. In the network architecture, secure user authentication and key distribution mechanisms utilizing cryptography must support these IoT security requirements [4]. In IoT network, user nodes and sensor nodes that interact with each other are exposed to various threats. To strengthen the security of the IoT network, user authentication schemes must guarantee the following security and functional requirements [5,6]:

* Corresponding author.
E-mail addresses: hjlee@security.re.kr (H. Lee), dwkang@security.re.kr (D. Kang), jhryu@security.re.kr (J. Ryu), dhwon@security.re.kr (D. Won), hyoung@skku.edu (H. Kim), ysooklee@howon.ac.kr (Y. Lee).

https://doi.org/10.1016/j.jisa.2020.102494

(1) **User anonymity**: The authentication scheme must maintain anonymity to ensure user privacy. In essence, an attacker cannot uncover the actual identity of the user.
(2) **Unlinkability**: The scheme must prevent the attacker from tracking the activity of the user, thereby guaranteeing unlinkability and enhancing user privacy.
(3) **Mutual authentication**: The scheme must provide mutual authentication for participants to verify each other's legitimacy.
(4) **Session key agreement**: In the authentication scheme, the session key used to encrypt and decrypt the message must be fresh, and forward secrecy must be assured.
(5) **Resilience to various attacks**: The authentication scheme must achieve all key security goals and resist various known attacks.

When secret keys are exposed, all traffic in the network can be decrypted. Even when a key stored in physical memory is exposed through a side channel attack, a user authentication scheme must implement countermeasures that prevent the attacker from intruding and controlling the IoT network. The revocation mechanism is a simple and efficient countermeasure. With the revocation mechanism implemented, when a user's private key is lost or stolen, the administrator issues a new key to the user.

Lately, numerous authentication schemes have been proposed for enhanced security. In 2007, Dhillon and Kalra [7] presented a three-factor remote user authentication scheme that is efficient in terms of computational cost in resource-constrained IoT environments. However, we discovered some security defects in their scheme. In this study, we perform an investigation of the security of their scheme using cryptanalysis and propose a new authentication scheme that resolves the security issues. Through security analysis, we demonstrate that the proposed scheme ensures all security requirements, and through performance analysis, we demonstrate that the scheme is suitable in terms of computational and communication cost for application in IoT environments.

The remainder of this paper is organized as follows: In Section 2, previous studies are explored. In Section 3, the preliminary knowledge for this study is introduced for an understanding of the background. In Section 4, Dhillon and Kalra's scheme [7] is reviewed, and the cryptanalysis performed on the scheme is presented in Section 5. In Section 6, the proposed scheme is presented. In Section 7, we provide an informal and a formal security analysis of the proposed scheme. In Section 8, we present the performance comparisons with the related schemes. Finally, the conclusions of this study are presented in Section 9.

2. Related work

Since Lamport [8] first proposed a password-based authentication scheme, many related studies of two-factor authentication schemes have been proposed to improve the security and efficiency of various network environments [9–11]. In addition, two-factor authentication schemes using various cryptographic technologies such as symmetric key cryptography, asymmetric key cryptography, and hash functions have been studied to provide secure user authentication in a wireless sensor networks (WSNs) [12–16].

In 2006, Wong et al. [17] first proposed a lightweight and dynamic password-based user authentication scheme for securely accessing WSNs. However, Das [18] claimed that the scheme proposed by Wong et al. [17] has security drawbacks (e.g., it cannot resist many logged-in users with the same login ID attacks and stolen-verifier attacks). To enhance the security of the scheme proposed by Wong et al. [17], Das [18] proposed a two-factor user authentication sch-eme for strong authentication and session key establishment using the gateway (GW). Unfortunately, it was later revealed by Khan and Alghathbar [19] and He et al. [20] that the scheme proposed by Das [18] is vulnerable to various attacks, including impersonation, privileged-insider attacks, and GW-node bypassing, and it does not guarantee mutual authentication between the GW and sensor nodes. To resolve this security problem, Khan and Alghathbar [19] proposed an enhanced two-factor user authentication scheme and claimed that their scheme had several security advantages. However, Vaidya et al. [21] discovered that the Khan and Alghathbars scheme [19] is not secure against smart-card theft, forgery, and node capture attacks. In 2011, Yeh et al. [22] also reported vulnerabilities in the scheme presented by Das [18] and proposed a new user authentication scheme that uses smart cards for WSNs. Yeh et al. [22] applied the elliptic curve cryptography (ECC)-based mechanism to the scheme to make it suitable for higher security in WSNs. However, according to Xue et al. [23], the scheme proposed by Yeh et al. [22] not only requires additional storage overhead but also requires increased computational resources. Then, Xue et al. [23] proposed a new scheme with strengthened security, but Li et al. [24] reported that various security weaknesses still remained [23]; these included vulnerabilities to loss of a smart card, offline-password guessing, stolen-verifier, insider, and many logged-in users with the same login ID attacks. Turkanovic et al. [25] presented an improved mutual authentication scheme to resolve these security challenges, ensuring essential features such as mutual authentication, key agreement, password security, and low computational costs, using hash and exclusive-OR (XOR) operations. Farash et al. [26] found security failures in the scheme proposed by Turkanovic et al. [25]; they reported that the scheme does not guarantee untraceability and anony-mity of the sensor node. To overcome these security vulnerabilities, Farash et al. [26] proposed a user authentication scheme for WSNs, tailored for IoT. However, Kumari et al. [27] reported that the scheme proposed by Farash et al. [26] violates user and sensor-node anonymity and is not secure against various attacks.

In Dhillon and Kalra's study [7], they highlight that traditional two-factor authentication protocols are insecure in real-world situations when a password breach or loss of smart device occurs. Based on the IoT network model (See Section 3.1) applied to the schemes [25–27] described earlier in this section, Dhillon and Kalra [7] proposed a lightweight multi factor user authentication scheme using password, biometric, and mobile device. They claimed that their scheme is secure against offline password guessing, password change, denial of service, stolen mobile device, and impersonation attacks. However, we found that their solution is also insecure from a user impersonation attack via a stolen mobile device attack, and it does not provide a session key agreement and a revocation plan.

In this study, we perform a security analysis to demonstrate the security failures of the Dhillon and Kalras scheme [7]. We then propose an improved lightweight authentication scheme that uses only XOR, hash, and symmetric cryptography and is suitable for IoT environments.

3. Preliminaries

3.1. Network model and authentication process

Currently, various IoT architecture models are being used to achieve security, scalability, and efficient computational cost. Xue et al. [23] introduced five resource-constrained communication mechanisms that address users, sensor nodes, and single or multiple gateways. We briefly describe the fifth network model applied to the Dhillon and Kalra's scheme [7] and our scheme, which shares the session key between the mobile node MN_i and the sensor node N_j. This mutual authentication is performed utilizing the

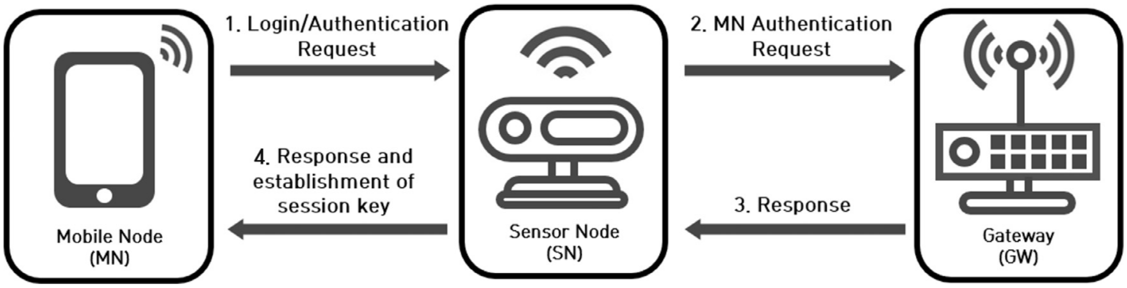

Fig. 1. User authentication model for IoT in the proposed scheme.

gateway GW, as shown in Fig. 1. The user authentication process is as follows:

(1) MN_i sends a login and authentication request to N_j to access the IoT network.
(2) Upon receipt of the request message, N_j sends the received request to GW for MN_i authentication.
(3) GW checks the message received from N_j, authenticates MN_i, and responds to N_j.
(4) N_j sends a response to MN_i, and then MN_i and N_j mutually establish a session key via authentication.

3.2. Bio-hash function

Biometrics provides a unique identification method for addressing security vulnerabilities in specific user credentials that can be forgotten or stolen, such as pins, passwords, and tokens. Imprint biometric characteristics vary slightly with each input for various reasons, such as dry or cracked skin, or the presence of dust on the imprint sensors [28]. To solve the problem of high false rejection rates, in 2004, Jin et al. [29] proposed a method of two-factor authentication based on inner products between tokenized pseudo-random numbers and user-specific fingerprint features. They created a user-specific compact code set called a bio-hash code. The bio-hash code randomly maps the biometric feature to a binary string using a user-specific token of pseudo-random numbers. The bio-hash has been applied to a variety of recently proposed schemes [30,31]. Bio-hash technology is efficient for biometrics-based multi-factor authentication schemes because it is suitable for small capacity devices [32].

4. Review of the dhillon and Kalra's scheme

In this section, we review Dhillon and Kalra's user authentication scheme [7], which consists of three steps: (1) registration, (2) login and authentication, and (3) the password change phase. Table 1 lists all the notations used in this paper.

Table 1
Notations.

Symbol	Description
MN_i	Mobile node (User)
N_j	Sensor node
GW	Gateway
ID_i, NID_j	Identities of MN_i and N_j
PW_i	MN_i's password
BIO_i	MN_i's biometrics
T_x	Timestamp
n_x, r_x	Random numbers
SK	Session key of between MN_i and N_j
$E_k(\cdot)$, $D_k(\cdot)$	Symmetric key encryption and decryption
$h(\cdot)$	Hash function
$H(\cdot)$	Bio-hash function
$\|\|$	Concatenation
$\oplus$	XOR operation
K_G	Private secret of GW
K_{GU}	Private key of MN_i
K_{GN}	Secret key shared between N_j and GW

4.1. Registration phase for user

In this phase, MN_i, a mobile node seeking to access the IoT service through a smart device application, registers with the GW, and the following operations are performed:

(a) MN_i selects its identity ID_i and password PW_i, inputs biometrics BIO_i, generates a random number r_i, and computes $MP_i = h(r_i||PW_i), MI_i = h(r_i||ID_i)$, and $MB_i = h(r_i||BIO_i)$.
(b) MN_i sends a request, $< MP_i, MI_i, MB_i >$, via a secure channel.
(c) After receiving the request message from MN_i, GW computes $x_i = h(MI_i||K_G), y_i = h(MP_i||K_{GU}), z_i = h(MB_i||K_{GU}), e_i = x_i \oplus y_i$, and $f_i = x_i \oplus z_i$.
(d) GW sends the response message, $< MI_i, e_i, f_i, x_i, K_{GU} >$, to MN_i.
(e) MN_i stores the received parameters along with r_i.

4.2. Registration phase for IoT node

In this phase, N_j registers with the GW, and the following operations are performed:

(a) N_j chooses a random number r_j and computes $MP_j = h(K_{GN}||r_j||NID_j), MR_j = r_j \oplus K_{GN}$, and $MPR_j = MP_j \oplus MR_j$.
(b) N_j sends the request, $< NID_j, MPR_j, MR_j, T_1 >$, to GW via a public channel.
(c) GW checks the freshness of T_1. If it is fresh, GW computes $MP_j = MPR_j \oplus MR_j, r_j = K_{GN} \oplus MR_j$, and $MP_j^* = h(K_{GN}||r_j||NID_j)$. GW then checks whether $MP_j^* \stackrel{?}{=} MP_j$. If it does, GW computes $x_j = h(NID_j||K_G), y_j = h(MP_j||K_{GN})$, and $z_j = x_j \oplus y_j$.
(d) Finally, GW sends the message, $< z_j, x_j, T_2 >$ to N_j, via an insecure open wireless channel.
(e) N_j checks the freshness of T_2. If it is fresh, N_j stores z_j and x_j in the memory storage.

4.3. Login and authentication phase

In this phase, MN_i, N_j, and GW carry out mutual authentication to set up a session key. The detailed description of the login and authentication phase is as follows:

(a) MN_i inputs ID_i, BIO_i, and PW_i and computes $MP_i = h(r_i||PW_i), MB_i = h(r_i||BIO_i), y_i^* = h(MP_i||K_{GU}), z_i^* = h(MB_i||K_{GU}), y_i = x_i \oplus e_i$, and $z_i = x_i \oplus f_i$. MN_i then checks if $y_i^* \stackrel{?}{=} y_i$ and $z_i^* \stackrel{?}{=} z_i$. If they are equal, MN_i generates a random number n_i and computes $UN_i = h(y_i||z_i||K_{GU}||T_1)$ and $UZ_i = n_i \oplus x_i$.
(b) MN_i sends the authentication request $M_1 =< MI_i, e_i, f_i, UZ_i, UN_i, T_1 >$ to N_j.
(c) N_j checks the freshness of T_1. If it is fresh, N_j computes $y_j = x_j \oplus z_j$ and $A_j = h(K_{GN}||T_1||T_2) \oplus y_j$.

(d) N_j sends the message $M_2 =< MI_i, e_i, f_i, UZ_i, UN_i, z_j, A_j, T_1, T_2 >$ to GW.
(e) GW_j checks the freshness of T_2. If it is fresh, GW computes $x_j^* = h(NID_j||K_G), y_j = z_j \oplus x_j^*, and y_j^* = A_j \oplus h(K_{GN}||T_1||T_2)$. GW then verifies whether $y_j^* \stackrel{?}{=} y_j$. If they are equal, GW computes $x_i^* = h(MI_i||K_G), z_i^* = f_i \oplus x_i^*, y_i^* = e_i \oplus x_i^*, UN_i^* = h(y_i^*||z_i^*||K_{GU}||T_1)$. GW then verifies whether $UN_i^* \stackrel{?}{=} UN_i$. If they are equal, GW computes $R_{ij} = x_i^* \oplus h(x_j^*||K_{GN}), H_j = h(x_j^*||K_{GN}||T_1||T_2||T_3)$, and $V_i = h(UN_i^*||T_1||T_2||T_3)$.
(f) GW sends $M_3 =< R_{ij}, H_j, V_i, T_1, T_2, T_3 >$ to N_j.
(g) N_j checks $T_{fresh} - T_3 \leq \Delta T$ and computes $H_j^* = h(x_j||K_{GN}||T_1||T_2||T_3)$. N_j verifies whether $H_j \stackrel{?}{=} H_j^*$. If they are equal, N_j chooses a random number m_j, and computes $x_i^* = R_{ij} \oplus h(x_j||K_{GN})$, and $n_i^* = UZ_i \oplus x_i^*$, $L_j = h(x_i^*||NID_j||T_1||T_2||T_3||T_4) \oplus m_j$ and $SK_{ij} = h(h(n_i^* \oplus m_j)||T_1||T_2)$.
(h) N_j sends $M_4 =< L_j, V_i, T_1, T_2, T_3 >$ to MN_i.
(i) MN_i checks $T_{fresh} - T_4 \leq \Delta T$. If they are equal, MN_i computes $V_i \stackrel{?}{=} h(UN_i||T_1||T_2||T_3), m_j^* = L_j \oplus h(x_i||NID_j||T_1||T_2||T_3||T_4)$, and $SK_{ij} = h(h(n_i \oplus m_j^*)||T_1||T_2)$.
(j) Finally, MN_i and N_j share the same session key $SK = h(h(n_i \oplus m_j)||T_1||T_2)$.

4.4. Password change phase

In this phase, MN_i performs the following process to change the password stored in its host mobile device:

(a) MN_i inputs BIO_i and PW_i and computes $MP_i = h(r_i||PW_i), MB_i = h(r_i||BIO_i), y_i^* = h(MP_i||K_{GU}), z_i^* = h(MB_i||K_{GU}), y_i = x_i \oplus e_i$, and $z_i = x_i \oplus f_i$.
(b) MN_i checks if $y_i^* \stackrel{?}{=} y_i$ and $z_i^* \stackrel{?}{=} z_i$. If they are equal, MN_i selects a new password, PW_i^{new}.
(c) MN_i computes new parameters $MP_i^{new} = h(r_i||PW_i^{new})$, $y_i^{new} = h(MP_i^{new}||K_{GU})$, and $e_i^{new} = x_i \oplus y_i^{new}$.
(d) Finally, MN_i replaces the old e_i with e_i^{new}.

5. Cryptanalysis of dhillon and Kalra's scheme

In this section, we conduct cryptanalysis of the Dhillon and Kalra's scheme [7]. For security analysis, we consider the following attacker capabilities:

(1) The attacker $\mathcal{A}$ can control the public channel by eavesdropping, inserting, deleting, altering, or intercepting public messages.
(2) If $\mathcal{A}$ somehow acquires a user's stolen or lost mobile device, he or she can perform a side channel attack to extract secret parameters from the device [33,34].
(3) $\mathcal{A}$ can enumerate all possible items offline in polynomial time in the Cartesian product $\mathcal{D}_{id} * \mathcal{D}_{pw}$, where $\mathcal{D}_{id}$ and $\mathcal{D}_{pw}$ represent the dictionary spaces of the identity and password, respectively [35–37].

5.1. Stolen mobile device attack

In the Dhillon and Kalra's scheme [7], $\mathcal{A}$ can simultaneously obtains the identifier and password of MN_i, from the stolen or lost users mobile device. $\mathcal{A}$ can perform offline guessing attacks using the following process:

(a) $\mathcal{A}$ extracts the secret parameters, $< MI_i, e_i, f_i, x_i, K_{GU}, r_i >$, from the user's mobile device.
(b) $\mathcal{A}$ selects the candidate identity ID_i^*, computes $MI_i^* = h(r_i||ID_i^*)$, and compares the extracted value with the calculated value, i.e., $MI_i \stackrel{?}{=} MI_i^*$.
(c) $\mathcal{A}$ selects the candidate password PW_i^*, computes $MP_i^* = h(r_i||PW_i^*)$ and $y_i^* = h(MP_i^*||K_{GU})$, and compares the extracted value with the calculated value, i.e., $y_i \stackrel{?}{=} y_i^*$.
(d) If the measurements show that they are matched, $\mathcal{A}$ has successfully found the correct identity and password. Otherwise, $\mathcal{A}$ chooses another ID_i^* and PW_i^*, and iterates steps (b) and (c) until the correct identity and password are found.
(e) $\mathcal{A}$ computes $x_i^* = e_i \oplus y_i^*$ and compares $x_i \stackrel{?}{=} x_i^*$. If they are the same, $\mathcal{A}$ proceeds to the next step.
(f) Finally, $\mathcal{A}$ obtains $z_i^* = f_i \oplus x_i^*$.

After successfully guessing MN_i's ID_i and PW_i through the above process, $\mathcal{A}$ can not only perform an impersonation attack using y_i^* and z_i^*, but also use the guessed identity and password to access another authentication system, or hack the user's sensitive data.

5.2. User impersonation attack

$\mathcal{A}$ can impersonate a legitimate user using the y_i^* and z_i^* values through the guessing attack. Moreover, $\mathcal{A}$ can more easily calculate y_i and z_i values only with e_i, f_i, and x_i values extracted from the user's mobile device without guessing ID_i^* and PW_i^* (e.g., $y_i^* = x_i \oplus e_i$ and $z_i^* = x_i \oplus f_i$).

The Dhillon and Kalra's scheme [7] allows the impersonation of a legitimate user during the login authentication phase through the following process:

(a) $\mathcal{A}$ inputs ID_A, PW_A and BIO_A and computes $MP_A = h(r_i||PW_A)$ and $MB_A = h(B_A||r_i)$.
(b) After this, $\mathcal{A}$ skips the calculation of the other parameters and instead injects the y_i^* and z_i^* into the local verification process.
(c) If $\mathcal{A}$ passes the local verification process, he or she generates a random number n_A and computes $UN_A = h(y_i^*||z_i^*||K_{GU}||T_1)$ and $UZ_A = n_A \oplus x_i$.
(d) $\mathcal{A}$ sends the authentication request, $M_1 =< MI_i, e_i, f_i, UZ_A, UN_A, T_1 >$, to N_j.
(e) Eventually, N_j and GW proceed with the rest of the login and authentication phase normally. Consequently, $\mathcal{A}$ and N_j establish a session key.

5.3. No provision for agreement of session key

In Dhillon and Kalra scheme [7], MN_i and N_j set up the session key SK, but they do not check to see whether the random numbers n_i and m_j included in the session key are correct, or they established the session key SK correctly after the mutual authentication. The protocol of reference [38,39] provides a session key agreement. The reason for ensuring the agreement of the session key is as follows: If, for some reason, an error occurs in the parameter value used to establish the session key, an erroneous session key may cause a communication failure. For this reason, the two nodes that set up the session key must perform a mutual process of checking whether the session key has been correctly calculated.

5.4. No provision for revocation

Revoking a user's stolen or lost mobile device is necessarily essential for authentication schemes in IoT environments [40]. If MN_i's legitimate mobile device is lost or stolen, an efficient revocation mechanism should be implemented to prevent future misuse of mobile devices and leakage of personal information. To support this mechanism, the server must maintain the users real identity

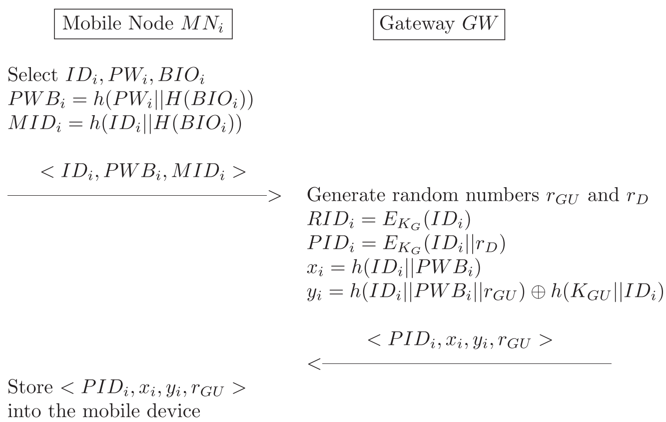

Fig. 2. Registration phase for user of the proposed scheme.

to detect invalid mobile devices [41]. However, Dhillon and Kalra [7] did not consider this feature in their scheme.

6. Proposed scheme

We suggest a three factor anonymous user authentication scheme for IoT environments. The proposed scheme contains the following four phases: (1) registration, (2) login and authentication, (3) password change, and (4) user-revocation phase.

6.1. Registration of user

The registration phase of the proposed scheme for MN_i is depicted in Fig. 2 and comprises the following operations:

(a) MN_i selects ID_i, PW_i, and BIO_i and computes $PWB_i = h(PW_i||H(BIO_i))$ and $MID_i = h(ID_i||H(BIO_i))$.
(b) MN_i sends $<ID_i$, PWB_i, $MID_i>$ to GW via the secure channel.
(c) GW selects random numbers r_{GU} and r_D, and computes $RID_i = E_{K_G}(ID_i)$, $PID_i = E_{K_G}(ID_i||r_{GU})$, $x_i = h(ID_i||PWB_i)$, and $y_i = h(ID_i||PWB_i||r_{GU}) \oplus h(K_{GU}||ID_i)$. GW stores a pair (RID_i, MID_i) in the database.
(d) GW sends $< PID_i, x_i, y_i, r_{GU} >$ to MN_i.
(e) Finally, MN_i stores the received parameters, $< PID_i$, x_i, y_i, $r_{GU} >$, in the mobile device.

6.2. Registration of IoT node

The registration phase of the proposed scheme for the sensor node N_j is depicted in Fig. 3 and consists of the following operations:

(a) N_j selects random number r_j and computes $MP_j = h(K_{GN}||r_j||NID_j)$ and $MI_j = r_j \oplus h(NID_j||K_{GN})$.
(b) N_j sends $< NID_j, MP_j, MI_j >$ to GW via the public channel.
(c) GW computes $r_j^* = MI_j \oplus h(NID_j||K_{GN})$ and $MP_j^* = h(K_{GN}||r_j^*||NID_j)$ and checks whether MP_j^* and MP_j are the same. If they are, GW computes $x_j = h(NID_j||K_{GN})$ and $y_j = x_j \oplus MP_j^*$.
(d) GW sends $< y_j >$ to N_j.
(e) N_j stores $< y_j >$ in the memory storage.

6.3. Login and authentication phase

In this phase, MN_i and N_j mutually authenticate each other with the support of GW to establish a session key. The login and authentication phase that are depicted in Fig. 4 are as follows:

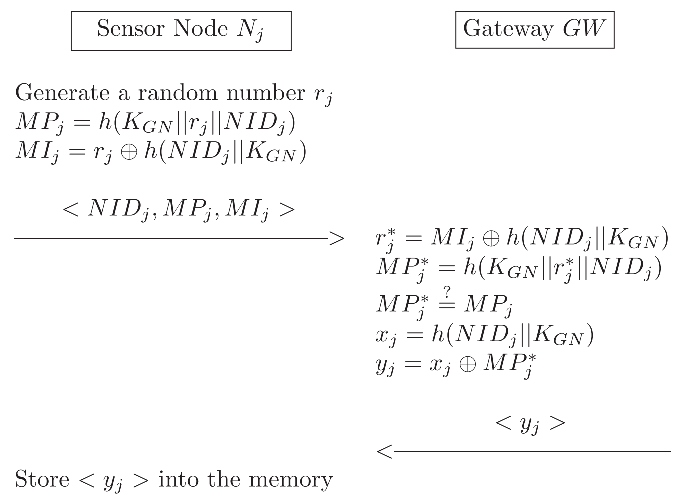

Fig. 3. Registration phase for the IoT node of the proposed scheme.

(a) MN_i enters ID_i, PW_i, and BIO_i, computes $PWB_i = h(PW_i||H(BIO_i))$ and $x_i^* = h(ID_i||PWB_i)$, and checks whether x_i^* and x_i are the same. If they are not, MN_i terminates this phase; otherwise, MN_i generates a random number n_i, and computes $A_i = y_i \oplus h(ID_i||PWB_i||r_{GU})$, $UN_i = h(A_i||PID_i||n_i)$, and $UZ_i = n_i \oplus A_i$.
(b) MN_i sends the request, $M_1 =< PID_i, UN_i, UZ_i, T_1 >$, to N_j.
(c) N_j checks the freshness of T_1. If it is fresh, N_j generates a random number n_j and computes $x_j = y_j \oplus h(K_{GN}||r_j||NID_j)$, $A_j = h(x_j) \oplus n_j$, and $B_j = h(x_j||n_j)$.
(d) N_j sends the message, $M_2 =< M_1, NID_j, A_j, B_j >$, to GW.
(e) After receiving the message from N_j, GW computes $x_j^* = h(NID_j||K_{GN})$, $n_j^* = h(x_j^*) \oplus A_j$, and $B_j^* = h(x_j^*||n_j^*)$ and checks whether B_j^* and B_j are the same. If they are not, GW terminates this phase; otherwise, GW gets MN_i's $< ID_i$, $r_D >$ by decrypting PID_i using a key K_G and computes $A_i^* = h(ID_i||K_{GU})$, $n_i^* = UZ_i \oplus A_i^*$, and $UN_i^* = h(A_i^*||PID_i||n_i^*)$ and checks whether UN_i^* and UN_i are the same. If they are not, GW terminates this phase; otherwise, GW generates r_D^{new} and computes $F_j = h(ID_i||n_i^*)$, $G_j = F_j \oplus x_j^*$, $R_{ij} = n_j^* \oplus n_i^*$, $H_j = h(x_j^*||n_j^*||n_i^*||F_j)$, and $PID_i^{new} = E_{K_G}(ID_i, r_D^{new})$.
(f) GW sends $M_3 =< PID_i^{new}, G_j, R_{ij}, H_j >$ to N_j.
(g) N_j computes $F_j^* = G_j \oplus x_j$, $n_i^* = R_{ij} \oplus n_j$ and $H_j^* = h(x_j||n_j||n_i^*||F_j^*)$ and checks whether $H_j^* = H_j$. If it does not, N_j terminates this phase; otherwise, N_j chooses a random number m_j and computes $L_j = h(NID_j||n_i^*) \oplus m_j$, $SK_{ji} = h(F_j^*||n_i^*||m_j)$, and $SV_j = h(SK_{ji}||T_1||T_2)$.
(h) N_j sends $M_4 =< PID_i^{new}, L_j, SV_j, T_2 >$ to MN_i.
(i) MN_i checks whether $T_{fresh} - T_2 \leq \Delta T$ and computes $m_j^* = L_j \oplus h(NID_j||n_i)$, $SK_{ij} = h(h(ID_i||n_i)||n_i||m_j^*)$, and $SV_i = h(SK_{ij}||T_1||T_2)$. If SV_i and SV_j are the same, MN_i and N_j successfully establish the same session key.

6.4. Password change phase

In this phase, MN_is password is changed on its mobile device. The details of this phase are as follows:

(a) MN_i inputs $ID_i, PW_i^{old}, PW_i^{new}$, and BIO_i, and computes $PWB_i^{old} = h(PW_i^{old}||H(BIO_i))$ and $x_i^* = h(ID_i||PWB_i^{old})$.
(b) MN_i checks whether x_i^* and x_i are the same. If they are not, MN_i terminates this phase. Otherwise, MN_i computes $A_i = y_i \oplus h(ID_i||PWB_i^{old}||r_{GU})$, $PWB_i^{new} = h(PW_i^{new}||H(BIO_i))$, $x_i^{new} = h(ID_i||PWB_i^{new})$, and $y_i^{new} = h(ID_i||PWB_i^{new}||r_{GU}) \oplus A_i \oplus y_i$.

Mobile Node MN_i	Sensor Node N_j	Gateway GW

Mobile Node MN_i:

Input ID_i, BIO_i, PW_i
$PWB_i = h(PW_i||H(BIO_i))$
$x_i^* = h(ID_i||PWB_i)$
$x_i^* \stackrel{?}{=} x_i$
Generate n_i
$A_i = y_i \oplus h(ID_i||PWB_i||r_{GU})$
$UN_i = h(A_i||PID_i||n_i)$
$UZ_i = n_i \oplus A_i$

$M_1 =< PID_i, UN_i, UZ_i, T_1 >$ ——>

Sensor Node N_j:

Check $T_{fresh} - T_1 \leq \Delta T$
Generate n_j
$x_j = y_j \oplus h(K_{GN}||r_j||NID_j)$
$A_j = h(x_j) \oplus n_j$
$B_j = h(x_j||n_j)$

$M_2 =< M_1, NID_j, A_j, B_j >$ ——>

Gateway GW:

$x_j^* = h(NID_j||K_{GN})$
$n_j^* = h(x_j^*) \oplus A_j$
$B_j^* = h(x_j^*||n_j^*)$
$B_j^* \stackrel{?}{=} B_j$
$< ID_i, r_D >= D_{K_G}(PID_i)$
$A_i^* = h(ID_i||K_{GU})$
$n_i^* = UZ_i \oplus A_i^*$
$UN_i^* = h(A_i^*||PID_i||n_i^*)$
$UN_i^* \stackrel{?}{=} UN_i$
Generate r_D^{new}
$F_j = h(ID_i||n_i^*)$
$G_j = F_j \oplus x_j^*$
$R_{ij} = n_j^* \oplus n_i^*$
$H_j = h(x_j^*||n_j^*||n_i^*||F_j)$
$PID_i^{new} = E_{K_G}(ID_i, r_D^{new})$

<—— $M_3 =< PID_i^{new}, G_j, R_{ij}, H_i >$

Sensor Node N_j:

$F_j^* = G_j \oplus x_j$
$n_i^* = R_{ij} \oplus n_j$
$H_j^* = h(x_j||n_j||n_i^*||F_j^*)$
$H_j^* \stackrel{?}{=} H_j$
Choose m_j
$L_j = h(NID_j||n_i^*) \oplus m_j$
$SK_{ji} = h(F_j^*||n_i^*||m_j)$
$SV_j = h(SK_{ji}||T_1||T_2)$

<—— $M_4 =< PID_i^{new}, L_j, SV_j, T_2 >$

Mobile Node MN_i:

Check $T_{fresh} - T_2 \leq \Delta T$
$m_j^* = L_j \oplus h(NID_j||n_i)$
$SK_{ij} = h(h(ID_i||n_i)||n_i||m_j^*)$
$SV_i = h(SK_{ij}||T_1||T_2)$
$SV_i \stackrel{?}{=} SV_j$

Fig. 4. Login and authentication phase of the proposed scheme.

(c) Finally, MN_i replaces the old x_i^{old} and y_i^{old} with x_i^{new} and y_i^{new}, respectively.

6.5. Revocation phase

To recover the secret parameters, MN_i performs a revocation mechanism for the mobile device as follows:

(a) When MN_i hopes to revoke or reissue a secret parameter, he or she inputs an old identity ID_i^{old}, a new identity ID_i^{new}, a new password PW_i^{new}, and BIO_i into his or her mobile device. MN_i then computes $PWB_i^{new} = h(PW_i^{new}||H(BIO_i))$, $MID_i^{old} = h(ID_i^{old}||H(BIO_i))$, and $MID_i^{new} = h(ID_i^{new}||H(BIO_i))$.

(b) MN_j sends the revocation request message, $< ID_i^{old}, ID_i^{new}, MID_i^{old}, MID_i^{new}, PWB_i^{new} >$, to GW via a secure channel.

(c) GW computes $RID_i^{old} = E_{K_G}(ID_i^{old})$ for verifying the identity of MN_i and then searches a pair $(RID_i^{old}, MID_i^{old})$ in the database to find a registered user. If the pairs (RID_i, MID_i) and $(RID_i^{old}, MID_i^{old})$ are equal, GW generates new random numbers r_D^{new} and r_{GU}^{new}, computes $PID_i^{new} =$

$E_{K_G}(ID_i, r_D^{new})$, $RID_i^{new} = E_{K_G}(ID_i^{new})$, $x_i^{new} = h(ID_i||PWB_i^{new})$, and $y_i^{new} = h(ID_i||PWB_i^{new}||r_{GU}^{new}) \oplus h(K_{GU}||ID_i^{new})$, and stores the new pair $(RID_i^{new}, MID_i^{new})$ in the database.

(d) GW sends $< PID_i^{new}, x_i^{new}, y_i^{new}, r_{GU}^{new} >$ to MN_i.

(e) MN_i stores the acquired parameters in the mobile device.

7. Security analysis of the proposed scheme

7.1. Informal security analysis

In this section, we perform an informal security analysis of the proposed scheme under the introduced attacker model to prove that it is secure against the various attacks that threaten the security and sustainability of IoT networks.

7.1.1. User anonymity

In the proposed scheme, we generate PID_i by encrypting MN_i's identity ID_i and a random number r_D with the secret key K_G, i.e., $PID_i = E_{K_G}(ID_i||r_D)$. It is different for each session because r_D is also changed simultaneously. After GW authenticates MN_i, GW changes the existing PID_i to a new PID_i^{new} and transmits it to MN_i. Therefore, even if $\mathcal{A}$ eavesdrops the public messages M_{1-4} on the public channel or extracts the secret parameters $< PID_i, x_i, y_i, r_{GU} >$ stored on the mobile device, the proposed scheme satisfies user anonymity because there is no way for $\mathcal{A}$ to recognize the real identity ID_i.

7.1.2. User untraceability

MN_i sends a message M_1 that includes PID_i, UN_i, and UZ_i to N_j via a public channel on which $\mathcal{A}$ can eavesdrop in the login and authentication phase. Because these parameters contain random values, such as n_i and r_D, that change and are different for each session, $\mathcal{A}$ cannot track the user's actions in the login and authentication phase, i.e., there is no message with the same value on the network. Therefore, the proposed scheme ensures the users untraceability.

7.1.3. Resistance to stolen mobile device attack

In the proposed scheme, to guess the user's ID_i and PW_i (personal identification information), $\mathcal{A}$ must have knowledge of the secret key K_{GU}. However, K_{GU} is not directly stored on the mobile device; it is protected with the hash function and is not sent via the public channel as plain text. Furthermore, even if we assume that $\mathcal{A}$ somehow obtains the secret key K_{GU}, he or she cannot guess PW_i without $H(BIO_i)$, which is unique to the user. Therefore, the proposed scheme resists stolen mobile device attacks.

7.1.4. Mutual authentication

MN_i and N_j authenticate each other with the assistance of GW in the login and authentication phase. Only a legal MN_i can calculate A_i using his or her information, which is again used by GW to confirm that MN is valid. Only if this verification process is completed, the next step can be performed. In addition, N_j, who calculates a valid x_j, can only be authenticated from GW. The verification process for N_j is performed immediately when GW receives the message M_2. MN_i determines whether N_j is legitimate by checking the fact that the message that N_j returns to MN_i contains valid information related to the random number n_i that MN_i has sent to GW. Therefore, the proposed scheme guarantees mutual authentication because all three participants check the validity of one another throughout the login and authentication process.

7.1.5. Session key agreement

After the login and authentication process, N_j generates the session key SK_{ji} using both random numbers of MN_i and N_j, calculates SV_j, and sends SV_j to MN_i. Then, MN_i also computes SK_{ij} and SV_i, using its own parameters and N_j's random number extracted from the received message. Then, MN_i checks if they share the same session key by checking whether SV_i and SV_j are equal. Because both parties need to calculate the session key correctly to complete the above process, the proposed scheme ensures a session key agreement.

7.1.6. Resistance to user impersonation attack

In the proposed scheme, $\mathcal{A}$ cannot disguise the user because the scheme resists a stolen mobile device attack through a local user verification process and mutual authentication. Therefore, as a secure session key agreement is guaranteed, the proposed scheme resists user impersonation attacks.

7.1.7. Resistance to replay attack

Even if $\mathcal{A}$ eavesdrops on messages M_{1-4} from the communication that is in the public channel and replays them, $\mathcal{A}$ cannot calculate the correct session key SK. To compute the session key SK, $\mathcal{A}$ would need to know n_i or m_j, and to know these, $\mathcal{A}$ needs GW's secret key K_G and K_{GU}. As there is no way for $\mathcal{A}$ to know the secret keys of GW from the message transmitted through the public channel, the proposed scheme is safe from replay attacks.

7.1.8. Local user verification

At the login and authentication phase of the proposed scheme, the mobile device checks the legitimacy of the user. Users who have entered the correct ID_i, PW_i, and BIO_i through the user verification process can perform the following authentication procedure. Therefore, the proposed scheme can block unauthorized access of $\mathcal{A}$ because the individual BIO_{mi} is unique.

7.1.9. Resistance to stolen-verifier attack

In the proposed scheme, GW does not directly receive MN_i's credentials such as PW_{mi} and $H(BIO_i)$. Furthermore, GW maintains the database with RID_i encrypted with its private key to confirm the legitimacy of the user, i.e., even if $\mathcal{A}$ steals the users registered information from the database for impersonation, it is difficult for $\mathcal{A}$ to know the actual identity of MN_i. Therefore, the proposed scheme is secure against stolen-verifier attacks.

7.1.10. Resistance to privileged-insider attack

The privileged-insider can attempt to impersonate a user by using a registration request message obtained at the user registration phase or additionally obtaining the stolen or lost mobile device of a user [47].

In the registration phase of the proposed scheme, MN_i sends ID_i and PWB_i, which contains PW_i and $H(BIO_{mi})$, to GW. However, an insider in a GW cannot guess MN_is PW_i without BIO_i if $\mathcal{A}$, as a malicious insider, extracts all the parameters $< PID_i, x_i, y_i, r_{GU} >$ stored in the device after he/she gets the stolen or lost mobile device of a user.

The insider needs BIO_i or the private key K_{GU} for MN_i to impersonate the user. It is impossible to determine BIO_i, which is an individual's biological characteristics, and if a security mechanism is applied that prevents insiders from knowing the secret key for users in GW's system, the insider cannot impersonate the user in any way.

Therefore, the insider cannot impersonate MN_i to access and communicate with N_j in the proposed scheme. Furthermore, in the password change phase of the proposed sch-eme, MN_i can change his or her password with PWB_i without the help of GW. The proposed scheme withstands privi-leged-insider attacks because it is impossible for the insider to know a MN_i's password.

7.1.11. User-friendly password change

A user's password can be changed from his or her end without server intervention. We apply this mechanism to the proposed scheme to allow the user to replace an old password with a new one after the user verification phase is executed. Therefore, the proposed scheme provides a user-friendly password changing process.

7.1.12. Forward secrecy

The computed session key between MN_i and N_j can be corrupted by $\mathcal{A}$. However, he or she cannot find significant correlations between the past, present, and future session keys because they contain random numbers n_i and m_j that are different in each session in the proposed scheme. Therefore, the proposed scheme guarantees forward security.

7.1.13. Resistance to sensor node impersonation attack

In this attack, we assume that $\mathcal{A}$ eavesdrops on the messages M_4 during the authentication and key agreement phase from the public channel and attempts to generate other messages $M_4 =< PID_i^{new}, L_j, SV_j, T_2 >$ to send them to MN_i. However, to generate M_3, $\mathcal{A}$ needs n_j and F_j. Therefore, $\mathcal{A}$ cannot impersonate a valid sensor node N_j in the proposed scheme. As a result, the proposed scheme is also secure against a sensor node impersonation attack.

7.1.14. Resistance to known session-specific temporary information attack

If the random numbers n_i and m_j are known to $\mathcal{A}$, he or she can attempt to compute the session key $SK = h(h(ID_i||n_i)||n_i||m_j^*)$. However, it require the knowledge of ID_i or $F_j = h(ID_i||n_j)$ from public messages M_2 and M_4. As we explained in Section 7.1.1 earlier, the proposed scheme ensures user anonymity through which ID_i is encrypted by the secret key K_{GU}. In addition, F_j is protected by x_j that is not transmitted as plain text. There is no way for $\mathcal{A}$ to get ID_i and the related parameters involving SK. Therefore, the proposed scheme resists the known session-specific temporary information attack.

7.1.15. Provisional revocation phase

In the proposed scheme, MN_i sends a revocation request to GW with $< ID_i^{old}, ID_i^{new}, MID_i^{old}, MID_i^{new}, PWB_i^{new} >$ when their mobile device is stolen or lost or when the secret parameters are exposed. Because GW maintains RID_i and MID_i in the database, when a revocation request is received from MN_i, GW computes $RID_i^{old} = E_{K_G}(ID_i^{old})$ and compares that the pairs (RID_i, MID_i) and $(RID_i^{old}, MID_i^{old})$ are same, to determine whether MN_i is a valid user. Since MID_i contains MN_i's ID_i and BIO_i, which is unique to the user, GW can only reissues the secret parameters to a legitimate user for recovery purposes. Thus, the proposed scheme can handle an unexpected case using provisional revocation.

7.2. Formal analysis using proverif

ProVerif is an automation tool for cryptographic protocol analysis, and it supports various cryptographic primitives such as symmetric and asymmetric encryptions, digital signatures, and hash functions. The principle by which ProVerif proves the security of a protocol by inputting and verifying the security attributes of the cryptographic primitives is introduced in the manual [48]. ProVerif is widely used by many researchers [49–51] to validate the security analysis of the key agreement and authentication schemes for various network environments. In this section, we verify the security of the proposed scheme using ProVerif, introduce ProVerif code as a description of the proposed scheme, and present the analysis results.

The execution of all the code described in Appendix A verifies the accuracy of all the events and queries and generates the simulation results presented in Fig. 5. All the authentication parameters, i.e., the queries and events between MN_i, N_j, and GW in the proposed scheme, perform successful mutual authentication and securely establish the session key as a result. Therefore, the proposed scheme can be considered secure for simulated attacks.

7.3. Formal analysis using the random oracle model

In this section, a formal security analysis of the proposed scheme is performed using a random oracle model. To this end, we first define a one-way hash function. A one-way hash function h: $\{0, 1\}^* \rightarrow \{0, 1\}^n$ maps data of an input $x \in \{0, 1\}^*$ of arbitrary size to a bit string of fixed size $h(x) \in \{0, 1\}^n$. The properties of a one-way hash function are as follows:

(1) **Pre-image resistance**: Given $y = h(x)$, it is computationally difficult to find an input x.
(2) **Second pre-image resistance**: Given $x \neq x'$, it is computationally difficult to find a different input x' such that $h(x') = h(x)$.
(3) **Collision resistance**: It is computationally difficult to find two different inputs x and x' such that $h(x) = h(x')$.

Theorem 1. *Assuming that the one-way hash function, $h(\cdot)$, behaves like an oracle, the proposed scheme is proven secure against $\mathcal{A}$ because it guarantees secure protection of MN_i's identity ID_i and GW's private key K_G.*

Reveal: *Given the hash value $y = h(x)$, the random oracle shall output the hash input value x unconditionally.*

Extract: *Given the encrypted message $C = E_{K_X}(P)$, the random oracle shall output the plain text P unconditionally.*

Proof. In the proposed scheme, we apply a method similar to that used for the formal security proof in [52,53]. We assume that $\mathcal{A}$ runs the experimental algorithm to derive ID_{mi} and K_G that are shown in Algorithm 1, $EXP1_{HASH}^{\mathcal{A}}$ for the proposed

Algorithm 1: Algorithm $EXP1_{HASH}^{\mathcal{A}}$.

```
1. Eavesdrop login request message
   ⟨PID_i, UN_i, UZ_i, T_1⟩ of MN_i
2. Call the Reveal oracle.
   Let (A'_i, PID'_i, n'_i) ← Reveal(UN_i)
3. Computes UZ'_i = n'_i ⊕ A'_i
4. if(UZ'_i = UZ_i) then
5.   Call the Reveal oracle.
     Let (K'_G, ID'_i) ← Reveal(A'_i)
6.   Call the Extract oracle.
     Let (ID''_i, r'_D) ← Reveal(PID_i)
7.     if(ID'_i = ID''_i) then
8.       Accept ID'_i as the correct identity
9.       Compute PID''_i = E_{K'_G}(ID'_i, r'_D)
10.      if(PID'_i = PID''_i) then
11.        Accept K'_G as the correct secret key
12.        return 1 (Success)
13.      else
14.        return 0
15.      end if
16.    else
17.      return 0
18.    end if
19. else
20.   return 0
21. end if
```

```
RESULT inj-event(endMNode(id)) ==> inj-event(beginMNode(id)) is true.
RESULT inj-event(endGateWay(id_6429)) ==> inj-event(beginGateWay(id_6429)) is true.
RESULT inj-event(endIotNode(id_17325)) ==> inj-event(beginIotNode(id_17325)) is true.
RESULT not attacker(SKji[]) is true.
RESULT not attacker(SKij[]) is true.
```

Fig. 5. Results of ProVerif code for the proposed scheme.

user authentication scheme. We define the success probability of $EXP1^{IAUAS}_{HASH,A}$ as $Success1^{\mathcal{A}}_{HASH} = |Pr[EXP1^{\mathcal{A}}_{HASH} = 1] - 1|$. The advantage function for this experiment becomes $Adv1^{IAUAS}_{HASH,A}(t, q_R, q_E) = max_A\{Success1^{\mathcal{A}}_{HASH}\}$, where the maximum value is determined by the execution time t and the number of queries q_R and q_E for the Reveal and Extract oracle, respectively. If $\mathcal{A}$ can successfully break the property of the hash function provided in Definition 1, $\mathcal{A}$ can directly derive ID_i and K_G by getting the desired input value of the hash function. We assume that the attacker performs the attack experiment detailed in Algorithm 1 after $\mathcal{A}$ detects the participant's full connection through the authentication request message transmitted in the public channel. However, it is difficult for $\mathcal{A}$ to invert the input value against a given hash value contained in the acquired messages, i.e., $Adv1^{\mathcal{A}}_{HASH}(t) \le \epsilon$ and $\forall \epsilon > 0$. We have $Adv1^{\mathcal{A}}_{HASH}(t, q_R, q_E) \le \epsilon$ because $Adv1^{\mathcal{A}}_{HASH}(t, q_R, q_E)$ depends on $Adv1^{\mathcal{A}}_{HASH}(t)$. As $Adv1^{\mathcal{A}}_{HASH}(t) \le \epsilon$ is negligible, we finally have $Adv1^{\mathcal{A}}_{HASH}(t, q_R, q_E) \le \epsilon$, which is also negligible. Consequently, $\mathcal{A}$ cannot acquires ID_i and K_G. Therefore, the proposed scheme is proven secure against the adversary $\mathcal{A}$ even if $\mathcal{A}$ can have full communication control on the public channel. □

Theorem 2. *Under the assumption that the one-way hash function $h(\cdot)$ behaves like an oracle, then the proposed scheme is proven secure against $\mathcal{A}$ by protecting ID_i, PW_i, and BIO_i of MN_i and K_G of GW.*

Proof. We assume that $\mathcal{A}$ executes the experimental algorithm $EXP2^{\mathcal{A}}_{HASH}$, which is detailed in Algorithm 2 , to derive ID_i, PW_i, BIO_i, and K_G. $\mathcal{A}$ exploits a side channel attack [33,54] to extract the secret parameters PID_i, x_i, y_i, and r_{GU} from the mobile device. We define the success probability of $EXP2^{\mathcal{A}}_{HASH}$ as $Success2^{\mathcal{A}}_{HASH} = |Pr[EXP\ 2^{\mathcal{A}}_{HASH} = 1] - 1|$. The advantage function for this experiment becomes $Adv2^{\mathcal{A}}_{HASH}(t, q_R) = max_A\{Success2^{\mathcal{A}}_{HASH}\}$, where the maximum value is determined by the execution time t and the number of queries q_R and q_E that for the Reveal and Extract oracle. If $\mathcal{A}$ can resolve the hash function problem, $\mathcal{A}$ can directly derive ID_{mi}, PW_{mi}, BIO_{mi}, and K_H. Consider the attack experiment shown in Algorithm 2. If $\mathcal{A}$ can successfully break the property of the hash function provided in Definition 2, $\mathcal{A}$ can directly derive ID_{mi}, PW_{mi}, BIO_{mi}, and K_H by getting the desired input value of the hash function. However, it is difficult for $\mathcal{A}$ to invert the input value against a given hash value contained in the extracted parameters, i.e., $Adv2^{\mathcal{A}}_{HASH}(t) \le \epsilon$, $\forall \epsilon > 0$. We have $Adv2^{\mathcal{A}}_{HASH}(t, q_R, q_E) \le \epsilon$, because $Adv2^{\mathcal{A}}_{HASH}(t, q_R, q_E)$ depends on $Adv2^{\mathcal{A}}_{HASH}(t)$. Because $Adv2^{\mathcal{A}}_{HASH}(t) \le \epsilon$ is negligible, we finally have $Adv2^{\mathcal{A}}_{HASH}(t, q_R, q_E) \le \epsilon$, which is also negligible. Consequently, $\mathcal{A}$ cannot acquire ID_i, PW_i, BIO_i, and K_G. Therefore, the proposed scheme is proven secure against $\mathcal{A}$ even if $\mathcal{A}$ can obtain the secret parameters stored in the mobile device. □

Algorithm 2: Algorithm $EXP2^{\mathcal{A}}_{HASH}$.

1. Extract the secret parameters, $< PID_i, x_i, y_i, r_{GU} >$, stored in the mobile device by the side channel attack.
2. Call the Reveal oracle. Let $(ID'_i, PWB'_i) \leftarrow Reveal(x_i)$
3. Call the Reveal oracle. Let $(PW'_i, BIO'_i) \leftarrow Reveal(PWB'_i)$
4. Compute $z_i = h(ID'_i||PWB'_i||r_{GU}) \oplus y'_i$
5. Call the Reveal oracle. Let $(K'_G, ID''_i) \leftarrow Reveal(z_i)$
6. **if**$(ID'_i = ID''_i)$ **then**
7. **Accept** ID'_{mi} **as the correct** ID_i **of** MN_i
8. Compute $PID'_i = E_{K'_G}(ID'_i||r'_{GU})$
9. **if**$(PID_i = PID'_i)$ **then**
10. **Accept** r'_{GU} **and** K'_G **as the correct** r_{GU} **and** K_G **of** MN_i
11. Compute $w_i = h(ID_i||h(PW'_i||H(BIO'_i))||r'_{GU})$
12. Compute $y'_i = w_i \oplus h(K'_G||ID'_i)$
13. **if**$(y_i = y'_i)$ **then**
14. **Accept** PW'_i **and** BIO'_i **as the correct** PW_i **and** BIO_i **of** MN_i
15. **return 1 (Success)**
16. **else**
17. **return 0**
18. **end if**
19. **else**
20. **return 0**
21. **end if**
22. **else**
23. **return 0**
24. **end if**

7.4. Authentication proof using BAN logic

In this subsection, we use Burrows-Abadi-Needham (BAN) logic [55] to provide the proof that MN_i and N_j perform a valid mutual authentication and to verify that the distributed session key between them is fresh. BAN logic is a formal logic that proves the belief that each of the entities participating in the authentication protocol trusts each other based on the source, freshness, and reliability of the messages. Many researchers [56–59] use it to analyze the security of cryptographic protocols.

The basic notations of BAN logic is as follows.

(1) $U \triangleleft C$: U sees condition C.
(2) $U| \equiv C$: Condition C is believed by U
(3) $\sharp(C)$: It makes a fresh C.
(4) $U| \sim C$: U expresses the condition C.
(5) $U \stackrel{K}{\longleftrightarrow} S$: U and S share a secret key K.
(6) $U \Rightarrow C$: Condition C is handled by U.
(7) $(C)_K$: C is encrypted under key K.

To prove mutual authentication of the proposed scheme, we use the following five rules of BAN logic.

(1) Rule 1: Message-meaning rule: $\frac{U|\equiv U \stackrel{K}{\leftrightarrow} S, U \triangleleft <C>_K}{U|\equiv S|\sim C}$: If U trusts that the key K is shared with S, U sees the C combined with K, then U trusts S once said C.
(2) Rule 2: Nonce-verification rule: $\frac{U|\equiv \#(C), U|\equiv S|\sim C}{U|\equiv S|\equiv C}$: If U trusts that C's freshness and U trusts S once said C, then U trusts that S trusts C.
(3) Rule 3: Believe rule: $\frac{U|\equiv C, U|\equiv M}{A|\equiv (C,M)}$: If U trusts C and M, (C, M) are also trusted by U.

(4) Rule 4: Freshness-conjuncatenation rule: $\frac{U|\equiv\#(C)}{A|\equiv\#(C,M)}$: If freshness of C is trusted by U, then U can trust the freshness of full condition.
(5) Rule 5: Jurisdiction rule: $\frac{U|\equiv S|\Rightarrow C, U|\equiv S|\equiv C}{U|\equiv C}$: If U trusts that S has jurisdiction over C, and U trusts that S trusts a condition C, then U also trusts C.

Since the main goal of the proposed scheme is to establish a session key between MN_i and N_j through mutual authentication, we must satisfy the following four goals.

(1) Goal 1: $MN_i \mid\equiv (MN_i \overset{SK}{\longleftrightarrow} N_j)$
(2) Goal 2: $N_j \mid\equiv (MN_i \overset{SK}{\longleftrightarrow} N_j)$
(3) Goal 3: $MN_i \mid\equiv N_j \mid\equiv (MN_i \overset{SK}{\longleftrightarrow} N_j)$
(4) Goal 4: $N_j \mid\equiv MN_i \mid\equiv (MN_i \overset{SK}{\longleftrightarrow} N_j)$

The four messages transmitted in the proposed scheme can be converted into the idealized form as follows.

(1) Using $M_1 =< PID_i, UN_i, UZ_i, T_1 >$, $MN_i \rightarrow N_j$: $UN_i = h(A_i||PID_i||n_i)$, $UZ_i = n_i \oplus A_i$. This is reduced as $MSG_1 : (PID_i, A_i, T_1)_{n_i}$
(2) Using $M_2 =< M_1, NID_j, A_j, B_j >$, $N_j \rightarrow GW$:$A_j = h(x_j) \oplus n_j$, $B_j = h(x_j||n_j)$. This is reduced as $MSG_2 : (M_1, NID_j, n_j)_{x_j}$
(3) Using $M_3 =< PID_i^{new}, G_j, R_{ij}, H_j >$, $GW_i \rightarrow N_j$:$G_j = F_j \oplus x_j^*$, $R_{ij} = n_j^* \oplus n_i^*$, $H_j = h(x_j^*||n_j^*||n_i^*||F_j)$. This is reduced as $MSG_3 : (F_j, n_j, n_i, K_{GN})_{x_j}$
(4) Using $M_4 =< PID_i^{new}, L_j, SV_j, T_2 >$, $N_j \rightarrow MN_i$: $L_j = h(NID_j||n_i^*) \oplus m_j$t,$SV_j = h(SK_{ji}||T_1||T_2)$. This is reduced as: $MSG_4 : (PID_i, m_j, T_1, T_2)_{n_i}$

To derive the goals of the proposed scheme, we define the following assumptions.

(1) A_1: $MN_i| \equiv \sharp(T_1)$
(2) A_2: $N_j| \equiv \sharp(n_j)$
(3) A_3: $GW| \equiv \sharp(K_{GN})$
(4) A_4: $N_j| \equiv \sharp(T_2)$
(5) A_5: $N_j \mid\equiv (N_j \overset{n_i}{\longleftrightarrow} MN_i)$
(6) A_6: $GW \mid\equiv (GW \overset{x_j}{\longleftrightarrow} N_j)$
(7) A_7: $N_j \mid\equiv (N_j \overset{x_j}{\longleftrightarrow} GW)$
(8) A_8: $MN_i \mid\equiv (MN_i \overset{n_i}{\longleftrightarrow} N_j)$
(9) A_9: $MN_i \mid\equiv N_j \Rightarrow (MN_i \overset{SK}{\longleftrightarrow} N_j)$
(10) A_{10}: $N_j \mid\equiv MN_i \Rightarrow (MN_i \overset{SK}{\longleftrightarrow} N_j)$

We describe the main proof of the proposed scheme using the BAN logic rules, messages and assumptions as follows.

(1) From MSG_1, we get V_1: $N_j \triangleleft (PID_i, A_i, T_1)_{n_i}$
(2) From A_5 and Rule 1, we get V_2: $N_j \mid\equiv MN_i \mid\sim (PID_i, A_i, T_1)_{n_i}$
(3) From A_1 and Rule 4, we get V_3: $N_j \mid\equiv \sharp(PID_i, A_i, T_1)_{n_i}$
(4) From V_1, V_2 and Rule 2, we get V_4: $N_j \mid\equiv MN_i \mid\equiv (PID_i, A_i, T_1)_{n_i}$
(5) From MSG_2, we get V_5: $GW \triangleleft (M_1, NID_j, n_j)_{x_j}$
(6) Using A_6 and Rule 1, we get V_6: $GW \mid\equiv N_j \mid\sim (M_1, NID_j, n_j)_{x_j}$
(7) From A_2 and Rule 4, we get V_7: $GW \mid\equiv \sharp(M_1, NID_j, n_j)_{x_j}$
(8) From V_5, V_6 and Rule 2, we get V_8: $GW \mid\equiv N_j \mid\equiv (M_1, NID_j, n_j)_{x_j}$
(9) From MSG_3, we get V_9: $N_j \triangleleft (F_j, n_j, n_i, K_{GN})_{x_j}$
(10) From A_7 and Rule 1, we get V_{10}: $N_j \mid\equiv GW \mid\sim (F_j, n_j, n_i, K_{GN})_{x_j}$
(11) From A_3 and Rule 4, we get V_{11}: $N_j \mid\equiv \sharp(F_j, n_j, n_i, K_{GN})_{x_j}$
(12) From V_9, V_{10} and Rule 2, we get V_{12}: $N_j \mid\equiv GW \mid\equiv (F_j, n_j, n_i, K_{GN})_{x_j}$
(13) From MSG_4, we get V_{13}: $MN_i \triangleleft (PID_i, m_j, T_1, T_2)_{n_i}$
(14) From A_8 and Rule 1, we get V_{14}: $MN_i \mid\equiv N_j \mid\sim (PID_i, m_j, T_1, T_2)_{n_i}$
(15) From A_4 and Rule 4, we get V_{15}: $MN_i \mid\equiv \sharp(PID_i, m_j, T_1, T_2)_{n_i}$
(16) From V_{13}, V_{14} and Rule 2, we get V_{16}: $MN_i \mid\equiv N_j \mid\equiv (PID_i, m_j, T_1, T_2)_{n_i}$
(17) From V_{12}, V_{16}, and $SK = h(F_j||n_i||m_j)$, we get V_{17}: $MN_i \mid\equiv (MN_i \overset{SK}{\longleftrightarrow} N_j)$ (Goal 1)
(18) From V_4, V_8, and $SK = h(h(ID_i||n_i)||n_i||m_j)$, we get V_{18}: $N_j \mid\equiv (MN_i \overset{SK}{\longleftrightarrow} N_j)$ (Goal 2)
(19) From A_9, V_{17} and Rule 5, we get V_{19}: $MN_i \mid\equiv N_j \mid\equiv (MN_i \overset{SK}{\longleftrightarrow} N_j)$ (Goal 3)
(20) From A_{10}, V_{18} and Rule 5, we get V_{20}: $N_j \mid\equiv MN_i \mid\equiv (MN_i \overset{SK}{\longleftrightarrow} N_j)$ (Goal 4)

From Goals 1, 2, 3, and 4 that we achieved above, we see that MN_i and N_j establish a session key through secure mutual authentication.

8. Performance analysis

In this section, we compare the computational and communication costs for the proposed scheme with other related schemes that have the same communication mechanism in IoT networks. We conducted a comparative analysis based on the computational cost and the amount of communication incurred during the login and authentication process.

We considered the 320-bit ECC (Elliptic multiplication) T_e, the 128-bit Advanced Encryption Standard (AES) algorithm T_s, and the 160-bit hash function T_h. We did not consider the XOR operation because it is negligible.

We assumed that the mobile node and gateway are computing environments on the following computing environments and evaluated the execution time of cryptographic operations. We refer to the experimental results of Abbasinezhad-Mood and Nikooghadam [60] for each cryptographic execution time on the following sensor node:

(1) Mobile node: Galaxy Note 9 Device, AP; Octa-Core Processor 2.7GHz + 1.7GHz, 8G memory, OS; Android 9.0, and Android Studio and Software Development Kits (SDK) tools.
(2) Sensor node: LPC1768 Device, ARM Cortex-M3(up to 100 MHz) processor, 512 kB flash memory, and 64 kB SRAM.
(3) Gateway: CPU; Intel(R) Pentium(R) processor G4600 (3.60 GHz), 8G memory, OS; Win10 64bit, and Visual Studio 2017 using the Crypto++ Library 8.1.

Based on our measurement results and the experimental results of Abbasinezhad-Mood and Nikooghadam [60], the cryptographic time of the mobile node, sensor node, and gateway are as follows:

(1) Mobile node: $T_e \approx 29.48\mu s$, $T_s \approx 76.2\mu s$, and $T_h \approx 106.38\mu s$
(2) Sensor node: $T_e \approx 1263\mu s$ and $T_h \approx 15.5\mu s$
(3) Gateway: $T_e \approx 2226\mu s$, $T_s \approx 5.4097\mu s$, and $T_h \approx 4.9465\mu s$

We summarize the results of the performance comparison in Table 3. It indicates that the Turkanovic et al.'s scheme [25] has significantly less computational complexity than other schemes. However, it has already been revealed by Farash et al. [26] that the Turkanovic et al. scheme [25] is vulnerable to various attacks. The computational costs of the schemes proposed by Das et al. [42], Chang et al. [43], Yang et al. [44], and Wu et al. [46] are inferior to that of the proposed scheme. Our comparison shows that the Banerjee et al.'s scheme [45] has the second-best performance. However, as shown in Table 2, their scheme does not include a revocation phase.

Using the method presented in [61,62], we compared the communications cost of the login and authentication phase. We assume that the lengths of the identity, timestamp, and random

Table 2
Comparison of security requirements.

Security property	[7]	[25]	[42]	[43]	[44]	[45]	[46]	Proposed
User anonymity	√	√	√	√	✗	√	√	√
User untraceablity	✗	✗	✗	✗	✗	✗	✗	√
Resistance to stolen mobile device attack	✗	✗	√	✗	✗	√	√	√
Mutual authentication	√	√	√	√	√	√	√	√
Session key agreement	√	√	√	✗	-	✗	√	√
Resistance to user impersonation attack	✗	√	√	√	√	√	√	√
Resistance to user replay attack	✗	√	√	√	√	√	√	√
Local user verification	√	√	√	✗	✗	√	√	√
Resistance to stolen-verifier attack	√	√	√	√	✗	√	√	√
Resistance to privileged-insider attack	√	√	✗	√	√	√	√	√
User-friendly password change	√	√	√	√	√	√	√	√
Forward secrecy	√	√	√	✗	√	√	√	√
Resistance to sensor node impersonation attack	√	√	√	√	√	√	√	√
Resistance to known session-specific temporary information attack	✗	√	√	√	√	√	√	√
Provisional revocation phase	✗	✗	✗	✗	✗	✗	✗	√

Table 3
Comparison of the computational cost.

Scheme	[7]	[25]	[42]	[43]	[44]	[45]	[46]	Proposed
MN(User)	$9T_h$	$7T_h$	$8T_h + 2T_e$	$7T_h + 2T_e$	$16T_h$	$9T_h$	$11T_h$	$9T_h$
SN	$6T_h$	$5T_h$	$9T_h + 1T_e$	$5T_h + 2T_e$	$16T_h$	$6T_h$	$5T_h$	$7T_h$
GW	$7T_h$	$7T_h$	$10T_h$	$9T_h$	$20T_h$	$6T_h$	$15T_h$	$8T_h + 2T_s$
Total	$22T_h$	$19T_h$	$27T_h + 3T_e$	$21T_h + 4T_e$	$52T_h$	$21T_h$	$31T_h$	$24T_h + 2T_s$
Time	$\approx 1085\mu s$	$\approx 856\mu s$	$\approx 1323\mu s$	$\approx 2585\mu s$	$\approx 2049\mu s$	$\approx 1080\mu s$	$\approx 1321\mu s$	$\approx 1116\mu s$

Table 4
Comparison of the communication cost.

Scheme	[7]	[25]	[42]	[43]	[44]	[45]	[46]	Proposed
MN(User)	832	672	672	512	864	800	864	480
SN	1760	1440	1184	1024	1728	2080	1408	1472
GW	576	576	512	512	1024	320	320	640
Messages	4	4	4	4	4	4	4	4
Total(bits)	2880	2688	2368	2048	3712	3200	2592	2592

number values are 128, 32, and 64 bits, respectively. The symmetric key encryption, the elliptic multiplication operation, and the hash function produce 256, 360, and 160 bits, respectively.

Table 4 summarizes the results of the comparison in terms of communication cost. The total communication cost of proposed scheme is 2112 bits, while the schemes of Das et al. [42], Turkanovic et al. [25], Chang et al. [43], Yang et al. [44], Banerjee et al. [45], Wu et al. [46], and Dhillon and Kalra [7] are 2368, 2688, 2048, 3712, 3200, 2592, and 2880 bits, respectively. The cost of the Chang et al.'s scheme [43] is less than the proposed scheme. However, their scheme is insecure, as previously mentioned.

We measured the performance of the proposed scheme using hardware approximations of mobile devices and sensor devices that can be used in real IoT environments. In the proposed scheme, the computation and communication costs of the mobile node and sensor node are slightly higher than those of some other schemes. However, this can be applied to extremely low-cost IoT devices because mobile nodes and sensor nodes use only XOR and hash operations for mutual authentication and session key establishment. Furthermore, the proposed scheme assures all security requirements. Therefore, the proposed scheme is suitable for application to IoT environments.

9. Conclusions

In this study, we report that the user authentication scheme of Dhillon and Kalra has some security pitfalls, and propose an enhanced scheme that solves these vulnerabilities and improves security. To prove the security strength of the proposed scheme, we performed informal and formal security analyses using the random oracle model, BAN logic, and ProVerif tool. The results of the analysis show that the proposed scheme is secure against various known attacks and satisfies all security requirements. Furthermore, we performed a comparative analysis of performance against other related schemes assuming the hardware specifications of mobile and sensor devices in a real IoT environment. The results of the analysis show that the proposed scheme is compatible with extremely low-cost IoT devices. Therefore, the scheme proposed in this study is practical for user authentication in IoT environments.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Hakjun Lee: Conceptualization, Software, Writing - original draft. **Dongwoo Kang:** Formal analysis. **Jihyeon Ryu:** Resources, Investigation. **Dongho Won:** Methodology, Validation. **Hyoungshick Kim:** Writing - review & editing. **Youngsook Lee:** Supervision.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1A2C1010159)

Appendix A

Fig. 6 presents the process definitions and identifiers of the proposed scheme. Here, we define the public and secure channels used between each party; predefined constants; secret key; session key; exclusive-OR, hash, and bio-hash functions; symmetric key cipher; and concatenation operation; and the start and end of communication between each node to be verified for the correspondence relationship of messages.

Fig. 7 shows the overall MN_i process code for the proposed scheme. We model the registration phase on lines 39–42 and the login and authentication phase on lines 43–60.

Fig. 8 shows the overall N_j process code for the proposed scheme. We model the registration phase on lines 62–67 and the login and authentication phase on lines 68–91.

```
1   (*......chanels......*)
2   free cha:channel [private].
3   free chb:channel.
4   free chc:channel.
5
6   (*......constants......*)
7   free IDi:bitstring [private].
8   free NIDj:bitstring.
9   free GW:bitstring.
10  free PWi:bitstring [private].
11  free BIOi:bitstring [private].
12
13  (*......secret key......*)
14  free KGU:bitstring [private].
15  free KGN:bitstring [private].
16  free KG:bitstring [private].
17
18  (*......shared key......*)
19  free SKij:bitstring [private].
20  free SKji:bitstring [private].
21
22  (*......functions......*)
23  fun concat(bitstring,bitstring) : bitstring.
24  fun syme(bitstring,bitstring):bitstring.
25  fun xor(bitstring,bitstring):bitstring.
26  fun h(bitstring):bitstring.
27  fun H(bitstring):bitstring.
28  reduc forall ma:bitstring, key:bitstring;    symd(syme(ma,key)
    ,key)=ma.
29  equation forall p:bitstring, q:bitstring; xor(xor(p,q),q)=p.
30
31  (*......events......*)
32  event beginGateWay(bitstring).
33  event endGateWay(bitstring).
34  event beginIotNode(bitstring).
35  event endIotNode(bitstring).
36  event beginMNode(bitstring).
37  event endMNode(bitstring).
```

Fig. 6. ProVerif code for the overall mobile node process.

```
38  (*......MN's process......*)
39  let pMNode=
40  let PWBi = h(concat(PWi,H(BIOi))) in
41  let MIDi = h(concat(IDi,H(BIOi))) in
42  out(cha,(IDi,PWBi,MIDi));
43  in(cha,(XPIDi:bitstring,Xxi:bitstring,Xyi:bitstring,XrGU:bitstring));
44  event beginMNode(IDi);
45  new ni:bitstring;
46  let xi'=h(concat(IDi,PWBi)) in
47  if Xxi=xi' then
48  let Ai=xor(Xyi,h(concat(IDi,concat(XrGU,PWBi)))) in
49  let UNi=h(concat(Ai,concat(XPIDi, ni))) in
50  let UZi=xor(Ai,ni) in
51  new T1:bitstring;
52  let M1=concat(XPIDi,concat(UNi,concat(UZi,T1))) in
53  out(chc,(M1));
54  in(chc,XM4:bitstring);
55  let (XXNPIDi:bitstring,XLj:bitstring,XSVj:bitstring,
56  XT2:bitstring)= XM4 in
57  let mi'=xor(XLj,h(concat(NIDj,ni))) in
58  let SKij=h(concat(h(concat(IDi,ni)),concat(ni, mi'))) in
59  let SVi=h(concat(SKij,concat(T1,XT2))) in
60  if(SVi = XSVj) then event endMNode(IDi).
```

Fig. 7. ProVerif code for the overall mobile node process.

```
61  (*......IoT Node's process......*)
62  let pFAgent=
63  new rj:bitstring;
64  let MPj=h(concat(KGN,concat(rj,NIDj))) in
65  let MIj=xor(rj,h(concat(NIDj,KGN))) in
66  out(chb,(NIDj,MPj,MIj));
67  in(chb,(Xyj:bitstring));
68  in(chc, (XM1:bitstring));
69  let (XPIDi:bitstring,XUNi:bitstring,XUZi:bitstring, XT1:bitstring)
    = XM1 in
70  event beginIotNode(NIDj);
71  new nj:bitstring;
72  let (XXM1:bitstring) = XM1 in
73  let xj=xor(Xyj,h(concat(NIDj,concat(rj,KGN)))) in
74  let Aj=xor(h(xj), nj) in
75  let Bj=h(concat(xj,nj)) in
76  let M2=concat(XXM1,concat(NIDj,concat(Aj, Bj))) in
77  out(chb,(M2));
78  in(chb,(XM3:bitstring));
79  let (XNPIDi:bitstring,XGj:bitstring,XRij:bitstring,
    XHj:bitstring)=XM3 in
80  let Fj'=xor(XGj,xj) in
81  let ni''=xor(XRij,nj) in
82  let Hj'=h(concat(xj,concat(nj,concat(ni'',Fj')))) in
83  if Hj'=XHj then
84  new mj:bitstring;
85  new T2:bitstring;
86  let Lj=xor(h(concat(NIDj,ni'')),mj) in
87  let SKji=h(concat(Fj',concat(ni'',mj))) in
88  let SVj=h(concat(SKji,concat(XT1,T2))) in
89  let M4 = concat(XNPIDi,concat(Lj,concat(SVj,T2))) in
90  out(chc,(M4));
91  event endIotNode(NIDj).
```

Fig. 8. ProVerif code for the overall mobile node process.

```
92  (*......GW's process......*)
93  let pHAgent=
94  in(cha,(XIDi:bitstring, XPWBi:bitstring, XMIDi:bitstring));
95  new rGU:bitstring;
96  new rD:bitstring;
97  let RIDi=syme(XIDi, KG) in
98  let PIDi=syme(concat(XIDi, rGU), KG) in
99  let xi=h(concat(XIDi,XPWBi)) in
100 let yi=xor(h(concat(XIDi,concat(XPWBi,rGU))),
    h(concat(KGU,XIDi))) in
101 out(cha,(PIDi,xi,yi,rGU));
102 in(chb,(XNIDj:bitstring, XMPj:bitstring, XMIj:bitstring));
103 let rj'=xor(XMIj,h(concat(XNIDj,KGN))) in
104 let MPj'=h(concat(KGN,concat(rj',XNIDj))) in
105 if MPj'=XMPj then
106 let xj'=h(concat(XNIDj,KGN)) in
107 let yj=xor(xj',MPj') in
108 out(chb, (yj));
109 event beginGateWay(GW);
110 let (XXNIDj:bitstring, XAj:bitstring, XBj:bitstring,
    XXXM1:bitstring) = XM2 in
111 let (XXPIDi:bitstring,XXUNi:bitstring,XXUZi:bitstring,
    XXT1:bitstring) = XXXM1 in
112 let xj''=h(concat(XXNIDj, KGN)) in
113 let nj'=xor(h(xj''),XAj) in
114 let Bj'=h(concat(xj'',nj')) in
115 let (IDi':bitstring, rD':bitstring) = symd(XXPIDi,KGU) in
116 let Ai'=h(concat(IDi',KGU)) in
117 let ni'=xor(XXUZi,Ai') in
118 let UNi'=h(concat(Ai',concat(XXPIDi,ni'))) in
119 if UNi'=XXUNi then
120 let Fj=h(concat(IDi,ni')) in
121 let Gj=xor(Fj,ni') in
122 let Rij=xor(ni',nj') in
123 let Hj=h(concat(xj'',concat(nj',concat(ni',Fj)))) in
124 new NrD:bitstring;
123 let NPIDi=syme(concat(IDi',NrD),KG) in
124 let M3=concat(NPIDi,concat(Gj,concat(Rij,Hj))) in
125 out(chb, (M3));
126 event endGateWay(GW).
```

Fig. 9. ProVerif code for the overall mobile node process.

Fig. 9 shows the overall *GW* process code for the proposed scheme. We model the registration phase on lines 93–108 and the login and authentication phase on lines 109–126.

The code shown in Fig. 10 is intended to model the attacker's capabilities and verify the equivalencies of interprocess. Lines 128–129 verify whether the session keys SK_{ij} and SK_{ji} are secure against the attacker. Lines 130–132 verify whether the internodal relationships of the proposed scheme are in the accurate procedure.

```
127 (*......quries......*)
128 query attacker(SKij).
129 query attacker(SKji).
130 query id:bitstring; inj-event(endIotNode(id)) ==> inj-
    event(beginIotNode(id)).
131 query id:bitstring; inj-event(endGateWay(id)) ==> inj-
    event(beginGateWay(id)).
132 query id:bitstring; inj-event(endMNode(id)) ==> inj-
    event(beginMNode(id)).
133 set traceDisplay=long.
134 process
135 ((!pMNode)|(!pFAgent)|(!pHAgent))
```

Fig. 10. ProVerif code for the overall mobile node process.

References

[1] Park O, Hwang H, Lee C, Shin J. Trends of 5g massive lot, electronics and telecommunications. Trends 2016;31(1):68–77.

[2] Series M. Imt vision–framework and overall objectives of the future development of imt for 2020 and beyond. Recommendation ITU 2015. 2083–0

[3] Ahmad I, Shahabuddin S, Kumar T, Okwuibe J, Gurtov A, Ylianttila M. Security for 5g and beyond. IEEE Communications Surveys & Tutorials 2019.

[4] Perrig A, Stankovic J, Wagner D. Security in wireless sensor networks. Commun ACM 2004;47(6):53–7.

[5] Mishra D. Efficient and secure two-factor dynamic id-based password authentication scheme with provable security. Cryptologia 2018;42(2):146–75.

[6] Srinivas J, Mukhopadhyay S, Mishra D. A self-verifiable password based authentication scheme for multi-server architecture using smart card. Wireless Personal Communications 2017;96(4):6273–97.

[7] Dhillon PK, Kalra S. Secure multi-factor remote user authentication scheme for internet of things environments. Int J Commun Syst 2017;30(16):e3323.

[8] Lamport L. Password authentication with insecure communication. Commun ACM 1981;24(11):770–2.

[9] Li L-H, Lin L-C, Hwang MS. A remote password authentication scheme for multiserver architecture using neural networks. IEEE Trans Neural Networks 2001;12(6):1498–504.

[10] Ramasamy R, Muniyandi AP. New remote mutual authentication scheme using smart cards. Trans Data Privacy 2009;2(2):141–52.

[11] Xu J, Zhu W-T, Feng DG. An improved smart card based password authentication scheme with provable security. Computer Standards & Interfaces 2009;31(4):723–8.

[12] Banerjee S, Mukhopadhyay D. Symmetric key based authenticated querying in wireless sensor networks. In: Proceedings of the first international conference on Integrated internet ad hoc and sensor networks. ACM; 2006. p. 22.

[13] Du W, Wang R, Ning P. An efficient scheme for authenticating public keys in sensor networks. In: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing. ACM; 2005. p. 58–67.

[14] Chatterjee S, Das AK. An effective ecc-based user access control scheme with attribute-based encryption for wireless sensor networks. Security and Communication Networks 2015;8(9):1752–71.

[15] Chung Y, Choi S, Won D. Anonymous authentication scheme for intercommunication in the internet of things environments. Int J Distrib Sens Netw 2015;11(11):305785.

[16] Park Y, Park Y. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. Sensors 2016;16(12):2123.

[17] Wong KH, Zheng Y, Cao J, Wang S. A dynamic user authentication scheme for wireless sensor networks. In: Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing-Vol 1 (SUTC'06)-Volume 01, IEEE Computer Society; 2006. p. 244–51.

[18] Das ML. Two-factor user authentication in wireless sensor networks. IEEE Trans Wireless Commun 2009;8(3):1086–90.

[19] Khan MK, Alghathbar K. Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'. Sensors 2010;10(3):2450–9.

[20] He D, Gao Y, Chan S, Chen C, Bu J. An enhanced two-factor user authentication scheme in wireless sensor networks. Ad hoc & sensor wireless networks 2010;10(4):361–71.

[21] Vaidya B, Makrakis D, Mouftah HT. Improved two-factor user authentication in wireless sensor networks. In: 2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications. IEEE; 2010. p. 600–6.

[22] Yeh H-L, Chen T-H, Liu P-C, Kim T-H, Wei HW. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors 2011;11(5):4767–79.

[23] Xue K, Ma C, Hong P, Ding R. A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. Journal of Network and Computer Applications 2013;36(1):316–23.

[24] Li C-T, Weng C-Y, Lee CC. An advanced temporal credential-based security scheme with mutual authentication and key agreement for wireless sensor networks. Sensors 2013;13(8):9589–603.

[25] Turkanović M, Brumen B, Hölbl M. 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 2014;20:96–112.

[26] Farash MS, Turkanović M, Kumari S, Hölbl M. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment. Ad Hoc Netw 2016;36:152–76.
[27] Kumari S, Das AK, Wazid M, Li X, Wu F, Choo K-KR, Khan MK. On the design of a secure user authentication and key agreement scheme for wireless sensor networks, concurrency and computation. Practice and Experience 2017;29(23):e3930.
[28] Mishra D, Das AK, Mukhopadhyay S. A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. Expert Syst Appl 2014;41(18):8129–43.
[29] Jin ATB, Ling DNC, Goh A. Biohashing: two factor authentication featuring fingerprint data and tokenised random number. Pattern Recognit 2004;37(11):2245–55.
[30] Chaudhry SA, Naqvi H, Khan MK. An enhanced lightweight anonymous biometric based authentication scheme for tmis. Multimed Tools Appl 2018;77(5):5503–24.
[31] Khan I, Chaudhry SA, Sher M, Khan JI, Khan MK. An anonymous and provably secure biometric-based authentication scheme using chaotic maps for accessing medical drop box data. J Supercomput 2018;74(8):3685–703.
[32] Chaudhry SA. A secure biometric based multi-server authentication scheme for social multimedia networks. Multimed Tools Appl 2016;75(20):12705–25.
[33] Jiang Q, Ma J, Yang C, Ma X, Shen J, Chaudhry SA. Efficient end-to-end authentication protocol for wearable health monitoring systems. Computers & Electrical Engineering 2017;63:182–95.
[34] Kumari S, Khan MK, Li X. A more secure digital rights management authentication scheme based on smart card. Multimed Tools Appl 2016;75(2):1135–58.
[35] Wang D, Gu Q, Cheng H, Wang P. The request for better measurement: a comparative evaluation of two-factor authentication schemes. In: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. ACM; 2016. p. 475–86.
[36] Jiang Q, Zeadally S, Ma J, He D. Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. IEEE Access 2017;5:3376–92.
[37] Lee H, Lee D, Moon J, Jung J, Kang D, Kim H, et al. An improved anonymous authentication scheme for roaming in ubiquitous networks. PLoS ONE 2018;13(3):e0193366.
[38] Sureshkumar V, Amin R, Anitha R. A robust mutual authentication scheme for session initiation protocol with key establishment. Peer-to-Peer Networking and Applications 2018;11(5):900–16.
[39] Yang L, Zheng Z. Cryptanalysis and improvement of a biometrics-based authentication and key agreement scheme for multi-server environments. PLoS ONE 2018;13(3):e0194093.
[40] Banerjee S, Odelu V, Das AK, Srinivas J, Kumar N, Chattopadhyay S, et al. A provably-secure and lightweight anonymous user authenticated session key exchange scheme for internet of things deployment. IEEE Internet Things J 2019.
[41] Li X, Peng J, Obaidat MS, Wu F, Khan MK, Chen C. A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems. IEEE Syst J 2019.
[42] Das AK, Kumari S, Odelu V, Li X, Wu F, Huang X. Provably secure user authentication and key agreement scheme for wireless sensor networks. Security and Communication Networks 2016;9(16):3670–87.
[43] Chang C-C, Le HD. A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks. IEEE Trans Wireless Commun 2015;15(1):357–66.
[44] Yang Z, Lai J, Sun Y, Zhou J. A novel authenticated key agreement protocol with dynamic credential for wsns. ACM Transactions on Sensor Networks (TOSN) 2019;15(2):22.
[45] Banerjee S, Chunka C, Sen S, Goswami RS. An enhanced and secure biometric based user authentication scheme in wireless sensor networks using smart cards. Wireless Personal Communications 2019:1–28.
[46] Wu F, Li X, Sangaiah AK, Xu L, Kumari S, Wu L, Shen J. A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks. Future Generation Computer Systems 2018;82:727–37.
[47] Das AK, Sutrala AK, Kumari S, Odelu V, Wazid M, Li X. An efficient multi–gateway-based three-factor user authentication and key agreement scheme in hierarchical wireless sensor networks. Security and Communication Networks 2016;9(13):2070–92.
[48] Blanchet B, Smyth B, Cheval V, Sylvestre M. Proverif 2.00: automatic cryptographic protocol verifier. In: user manual and tutorial, Version from; 2018. p. 05–16.
[49] Chaudhry SA, Khan I, Irshad A, Ashraf MU, Khan MK, Ahmad HF. A provably secure anonymous authentication scheme for session initiation protocol. Security and Communication Networks 2016;9(18):5016–27.
[50] Karuppiah M, Kumari S, Li X, Wu F, Das AK, Khan MK, Saravanan R, Basu S. A dynamic id-based generic framework for anonymous authentication scheme for roaming service in global mobility networks. Wireless Personal Communications 2017;93(2):383–407.
[51] Ryu J, Lee H, Kim H, Won D. Secure and efficient three-factor protocol for wireless sensor networks. Sensors 2018;18(12):4481.
[52] Das AK. A secure and effective user authentication and privacy preserving protocol with smart cards for wireless communications. Networking Science 2013;2(1–2):12–27.
[53] Lu Y, Li L, Yang X, Yang Y. Robust biometrics based authentication and key agreement scheme for multi-server environments using smart cards. PLoS ONE 2015;10(5):e0126323.
[54] Wu F, Xu L, Kumari S, Li X, Khan MK, Das AK. An enhanced mutual authentication and key agreement scheme for mobile user roaming service in global mobility networks. Ann Telecommun 2017;72(3–4):131–44.
[55] Burrows M, Abadi M, Needham RM. A logic of authentication, proceedings of the royal society of london. A Mathematical and Physical Sciences 1989;426(1871):233–71.
[56] Chatterjee S, Roy S, Das AK, Chattopadhyay S, Kumar N, Vasilakos AV. Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment. IEEE Trans Dependable Secure Comput 2016;15(5):824–39.
[57] Jung J, Kang D, Lee D, Won D. An improved and secure anonymous biometric-based user authentication with key agreement scheme for the integrated epr information system. PLoS ONE 2017;12(1):e0169414.
[58] Odelu V, Das AK, Goswami A. An effective and robust secure remote user authenticated key agreement scheme using smart cards in wireless communication systems. Wireless Personal Communications 2015;84(4):2571–98.
[59] Kang D, Jung J, Mun J, Lee D, Choi Y, Won D. Efficient and robust user authentication scheme that achieve user anonymity with a markov chain. Security and Communication Networks 2016;9(11):1462–76.
[60] Abbasinezhad-Mood D, Nikooghadam M. Design and hardware implementation of a security-enhanced elliptic curve cryptography based lightweight authentication scheme for smart grid communications. Future Generation Computer Systems 2018;84:47–57.
[61] Reddy AG, Das AK, Odelu V, Yoo KY. An enhanced biometric based authentication with key-agreement protocol for multi-server architecture based on elliptic curve cryptography. PLoS ONE 2016;11(5):e0154308.
[62] Kumari S, Khan MK, Atiquzzaman M. User authentication schemes for wireless sensor networks: a review. Ad Hoc Netw 2015;27:159–94.