Pattern Recognition Letters 138 (2020) 439–446

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

On the design of biometric-based user authentication protocol in smart city environment

Basudeb Bera[a], Ashok Kumar Das[a,*], Walter Balzano[b], Carlo Maria Medaglia[c]

[a] Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India
[b] DIETI, University of Naples "Federico II", I-80138, Naples, Italy
[c] Link Campus University, Rome, Italy

ARTICLE INFO

Article history:
Received 16 May 2020
Revised 2 August 2020
Accepted 22 August 2020
Available online 22 August 2020

Keywords:
Smart city
User authentication
Key agreement
Biometrics
Security
AVISPA
MIRACL

ABSTRACT

Among the security services, like authentication, access control, key management and intrusion detection, user authentication is very much needed for a smart city environment because an external authorized user may require the real time data to be accessed directly from the deployed Internet of Things (IoT) enabled smart devices. Using the established session key between the user and an access smart device though mutual authentication and key agreement process, the real time data can be securely accessed. To deal with this issue, we propose a new user authentication scheme in smart city environment using three factors of a legal registered user (mobile device, password and biometrics). The proposed scheme is shown to be robust against a number of potential attacks needed in an IoT-based smart city deployment. The simulation study for formal security verification using the widely-accepted "Automated Validation of Internet Security Protocols and Applications (AVISPA)" tool demonstrates that the proposed scheme is also secure. Furthermore, experiments on various cryptographic primitives have been carried out using "MIRACL Cryptographic SDK: Multiprecision Integer and Rational Arithmetic Cryptographic Library" under both server and Raspberry PI 3 settings. Finally, a comprehensive comparative analysis shows the effectiveness and better security of the proposed scheme as compared with other state of art user authentication schemes.

1. Introduction

In order to deliver valuable time saving along with resource saving benefits, a smart city construction is essential. It needs higher degrees of network connectivity to assist new ultra-modern features [1]. At the same time, it also results in increasing the security and privacy issues [2]. Through the use of existing Internet of Things (IoT) smart devices it is possible to gather the data from multiple resources and then send the data to the storage centers (e.g., cloud servers) with the help of the existing networks. As a result, the IoT will produce a huge volume of data that can be leveraged for efficiency, safety, and potential applications as well as services for city residents [3–5]. However, it gives an opportunity to the malicious attackers to create a possible entry point to attacking the system [1].

The communication among various entries in a smart city environment takes place over the public communication channels (for example, through the Internet). Hence, an attacker can not only intercept the communicating messages, but can also modify, update or inject malicious message contents in between the communication. As a result various types of attacks, including "replay", "man-in-the-middle", "privileged-insider", "online/offline guessing" and "impersonation" attacks are possible. Apart from these attacks, the attacker can also trace which party is communicating with whom for how long period during communication. This poses that in a smart city environment, "anonymity" and "untraceability" properties need to be also preserved.

User authentication is one of the promising security services that provides a user to access directly the real-time information from some specified registered IoT smart devices. For this purpose, in this work, we design a new three-factor user authentication scheme that relies on "user password", "user personal biometrics" and "user mobile device".

* Corresponding author.
E-mail addresses: basudeb.bera@research.iiit.ac.in (B. Bera), ashok.das@iiit.ac.in (A.K. Das), walter.balzano@gmail.com (W. Balzano), c.medaglia@unilink.it (C.M. Medaglia).

https://doi.org/10.1016/j.patrec.2020.08.017
0167-8655/

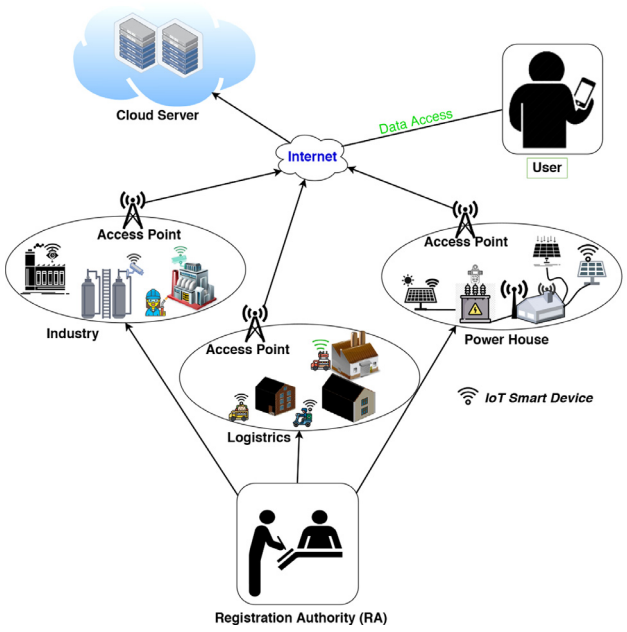

Fig. 1. A user authentication model for smart city environment.

1.1. Network model

A generic architecture for user authentication in a smart city environment is described in Fig. 1. The trusted registration authority (*RA*) takes the responsibility to register all the IoT smart devices (SD_j) and access points (AP_l) for various applications (for example, industry, power house, and so on) prior to their functioning in the smart city environment. A user (U_i) can also be registered by the *RA* while sending his/her pseudo identity and pseudo password to the *RA* via a secure channel (e.g., via person). After successfully registration process, if a register user wants to access a real time data from some designated smart devices, he/she should first be authenticated by the respective access point, and then can access real time data from the specified smart devices via access point after establishing session keys between them (user and smart devices). For backup purpose, the real time data can be also stored in the cloud securely that can be further utilized for Big data analytics purpose.

1.2. Attack model

We imitate the broadly-accepted "Dolev-Yao (DY) attack model" [6] and also "Canetti and Krawczyk's model (CK-adversary model)" [7] in this paper. Under the DY attack model, an adversary $\mathcal{A}$ not only can push the malicious data but can also modify or delete the legitimate message information transmitted in between communication via public channel. Moreover, under CK-adversary model, $\mathcal{A}$ can compromise a session state and reveal the session secret credential and also the session key from the communicated messages utilizing the session hijacking attack. Since any application in the smart city environment cannot be monitored in 24 $\times$ 7 h, and also due to hostile environment or power exhaustion, $\mathcal{A}$ can physically compromise some of the smart devices. Once a smart device is physically capture, $\mathcal{A}$ can extract all the credential from the compromised device's memory utilizing "power analysis attack" [8] and $\mathcal{A}$ can apply impersonation attack using these extracted information.

1.3. Paper outline

Section 2 provides a related work on recently proposed user authentication schemes. Various phases related to our proposed user authentication scheme are discussed in Section 3. Sections 4 and 5 give the detailed security analysis including formal security verification. We then provide a comparative study in Section 6. The paper is finally concluded in Section 7.

2. Related work

Several authentication and batch verification schemes have been proposed in Vehicular Ad Hoc Networks (VANETs) [9–12] that are suitable for smart city environment. Dhillon and Kalra [13] presented a biometric based user authentication protocol for the Internet of Things (IoT) environment. Though they used perception hashing operation to design their scheme, their scheme cannot support "user untraceability" and "anonymity", "Ephemeral Secret Leakage (ESL) attack under CK-adversary model", and "privileged insider attack". Altaibi [14] designed a biometric based user authentication for wireless sensor network (WSN). Their scheme is not secure as it cannot protect user untraceability and anonymity, ESL attack under CK-adversary model, and privileged insider attack.

Kang et al. [15] proposed another biometric-based user authentication for IoT environment. They pointed out that Kaul and Awasthi's authentication scheme [16] was vulnerable to "user anonymity", "offline password guessing attack", "user impersonation attack", and "time synchronization problem". However, Kang et al.'s scheme [15] is also vulnerable to "ESL attack under privileged insider attack".

Ryu et al. [17] demonstrated that Wu et al.'s scheme [18] had security flaws and does not handle outsider attack under stolen smart card, user impersonation attack, and also does not preserve user anonymity. Furthermore, Li et al. [19] proposed a "three factor user authentication protocol" for WSN. They pointed out Amin et al.'s scheme [20] scheme fails to protect denial-of-service (DoS) attack and preserve forward secrecy. At the same time, the scheme [19] cannot also provide user untraceability, and it is vulnerable to "ESL attack under CK-adversary model".

3. Proposed biometric-based user authentication scheme

In this section, we discuss a new biometric-based user authentication scheme smart city environment, which is based on user password and personal biometrics. The proposed scheme has the following phases, namely 1) "registration phase" where all the IoT smart devices, users and access points are registered by trusted registration authority (*RA*), 2) "login and authentication phase" where a legal registered user mutually authenticate with an authorized IoT smart device with the help of an access point acting as gateway node prior to accessing the real-time data from the IoT smart device, 3) "user password and biometric update phase" where a legal register user may wish to update his/her password and/or biometrics at any time with contacting the *RA* for security reasons, and 4) "dynamic IoT smart device addition phase" which allows a new IoT smart device to be added into the existing smart city environment after the initial deployment of the nodes.

We utilize the "random nonces (secrets)" and "current system timestamps" in order to protect replay attack in the network. Therefore, it is a typical supposition that all the network entities are synchronized with their clocks. This is a usual assumption used in designing the authentication protocols in different networking environments [21–29]. For better understanding in the various phases of the proposed scheme, we make use of the notations tabulated in Table 1.

Table 1
Notations and their significance.

Symbol	Significance
ID_X, PID_X	Real and pseudo-identities of an entity X, respectively
TID_X	Temporary identity of an entity X
TC_X	Temporal credential of an entity X
p	A sufficiently large prime
$f_l(x, y)$	A symmetric bivariate t-degree polynomial over the Galois field $GF(p)$: $f(x,y)=\sum_{u=0}^{t}\sum_{v=0}^{t} b_{u,v}x^u y^v$ where $b_{u,v} \in Z_p = \{0, 1, 2, \cdots, p-1\}$
MK_X	Master key of an entity X
$\|\|$, $\oplus$	Concatenation & bitwise XOR operations, respectively
TS_X	"Current timestamp generated by an entity X"
ΔT	"Maximum transmission delay associated with a message"
$h(\cdot)$	"Collision-resistant cryptographic one-way hash function"
$E(\cdot)/D(\cdot)$	Symmetric encryption/decryption
PW_{U_i}, Bio_{U_i}	Password and personal biometrics of a user U_i, respectively
$Gen(\cdot)/Rep(\cdot)$	"Fuzzy extractor generation & reproduction functions"
SK_{U_i,SD_j}	Session key shared between user U_i and smart device SD_j

3.1. Registration phase

In this phase, the trusted registration authority (RA) is responsible for registering all the IoT smart devices and access points for various smart applications in a smart city environment.

3.1.1. Offline IoT smart device registration

Prior to deployment of each IoT smart device (SD_j) in a particular application in smart city environment, the RA does the following steps:

Step SDR_1*:* For each SD_j, the RA selects a unique identity ID_{SD_j} and a unique random master key MK_{SD_j}, and calculates the temporal credential as $TC_{SD_j} = h(ID_{SD_j} \| MK_{SD_j} \| RTS_{SD_j} \| MK_{RA})$, where RTS_{SD_j} is the registration timestamp of SD_j and MK_{RA} is the RA's own master secret key that is known to the RA only.

Step SDR_2*:* The RA then pre-loads the credentials $\{ID_{SD_j}, TC_{SD_j}, h(\cdot)\}$ prior to the deployment of SD_j in its target field.

3.1.2. Access point registration

For registration of an access point (AP_l) belonging to a particular application having n_{sd} number of registered IoT smart devices SD_j ($j = 1, 2, \cdots, n_{sd}$), the RA executes the following steps:

Step APR_1*:* The RA picks a unique identity ID_{AP_l} and a unique master key MK_{AP_l}, and computes the pseudo identity $PID_{AP_l} = h(ID_{AP_l} \| MK_{AP_l} \| MK_{RA} \| RTS_{AP_l})$ corresponding to ID_{AP_l}, where RTS_{AP_l} is the registration timestamp of AP_l.

Step APR_2*:* Next, RA picks a "t-degree symmetric bivariate polynomial of the form: $f_l(x,y) = \sum_{u=0}^{t}\sum_{v=0}^{t} b_{u,v}x^u y^v \pmod p$ over a Galois (finite) field $GF(p)$" such that $t \gg$ the number of access points AP_l and users placed for a particular application in a smart city so that the proposed scheme will be "unconditionally secure and t-collusion resistant" against user mobile devices capture attacks [30].

After that, the RA computes its own polynomial share $f_l(PID_{AP_l}, y)$, and jumps to next step.

Step APR_3*:* The RA loads the credentials $\{PID_{AP_l}, \{(ID_{SD_j}, TC_{SD_j})| j = 1, 2, \cdots, n_{sd}\}, h(\cdot), f_l(PID_{AP_l}, y)\}$ prior to the placement of AP_l. For security reasons, it is assumed that each AP_l will be put under a physical locking system as it was the case in [31], and all credentials will be stored in the secure/tamper-resistant database of AP_l so that stolen verifier attack is not feasible by an adversary. Therefore, physical capturing of access points will not be possible in our scheme.

3.1.3. User registration

To register at the RA, a user U_i and the RA perform the following steps:

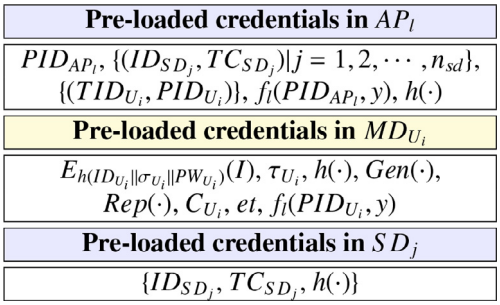

Fig. 2. Pre-loaded information stored in SD_j, AP_l and U_i during registration phase.

Step UR_1*:* U_i first picks an identity ID_{U_i}, a chosen-password PW_{U_i}, and also imprints his/her personal biometrics Bio_{U_i} (for example, fingerprint) at the sensor of a specific terminal (say, his/her mobile device MD_{U_i}). U_i then computes a "biometric secret key σ_{U_i}" and a "public reproduction parameter τ_{U_i}" using fuzzy extractor technique [32] as $Gen(Bio_{U_i}) = (\sigma_{U_i}, \tau_{U_i})$, where the $Gen(\cdot)$ and $Rep(\cdot)$ represent the "fuzzy extractor probabilistic generation function" and "deterministic reproduction function" respectively, those are described as follows.

- $Gen(\cdot)$: It is a "probabilistic algorithm", which takes the biometrics data Bio_{U_i} as input, and produces a "secret biometric key σ_{U_i}" and a "public reproduction parameter τ_{U_i}" as output, where $Gen(Bio) = \{\sigma_{U_i}, \tau_{U_i}\}$.
- $Rep(\cdot)$: This is a "deterministic algorithm", which takes a "noisy biometrics data Bio'_{U_i}", the "public parameter τ_{U_i}", and "et (an error tolerance threshold value)" related to Bio_{U_i}, and then reproduces (recovers) the "biometrics secret key σ_{U_i}". That is, $Rep(Bio'_{U_i}, \tau_{U_i}) = \sigma_{U_i}$ with the condition that the "Hamming distance ($Hd(\cdot)$) between registered biometrics Bio_{U_i} and current biometrics Bio'_{U_i} is less than or equal to et" [32], that is, $Hd(Bio_{U_i}, Bio'_{U_i}) \le et$.

Step UR_2*:* U_i generates threee random secrets, say α, β and γ, and computes the pseudo identity and password as $PID_{U_i} = h(ID_{U_i} \| \alpha)$ and $RPW_{U_i} = h(PW_{U_i} \| \sigma_{U_i} \| \beta)$, respectively. U_i sends the registration request $RReq_{U_i} = \{PID_{U_i}, (RPW_{U_i} \oplus \gamma)\}$ to the RA via secure channel (for example, in person)

Step UR_3*:* After receiving $RReq_{U_i}$, the RA calculates $A_{U_i} = h(PID_{U_i} \| MK_{RA} \| ID_{RA})$ and $B_{U_i} = h(MK_{RA} \| RTS_{U_i} \| ID_{RA}) \oplus (RPW_{U_i} \oplus \gamma)$, where RTS_{U_i} is the registration timestamp of U_i. Next, the RA calculates a polynomial share $f_l(PID_{U_i}, y)$, and sends the registration reply $RRep_{U_i} = \{A_{U_i}, B_{U_i}, f_l(PID_{U_i}, y)\}$ to U_i securely.

Step UR_4*:* After receiving $RRep_{U_i}$, U_i calculates $B^*_{U_i} = B_{U_i} \oplus \gamma = h(MK_{RA} \| RTS_{U_i} \| ID_{RA}) \oplus RPW_{U_i}$. U_i then encrypts the information $I = \{A_{U_i}, B^*_{U_i}, \beta, PID_{U_i}, PID_{AP_l}\}$ using the constructed symmetric key $K_{U_i} = h(ID_{U_i} \| \sigma_{U_i} \| PW_{U_i})$, that is, U_i stores the information $\{E_{h(ID_{U_i}\|\sigma_{U_i}\|PW_{U_i})}(I), \tau_{U_i}, h(\cdot), Gen(\cdot), Rep(\cdot), C_{U_i}, et, f_l(PID_{U_i}, y)\}$ in his/her mobile device MD_{U_i}, where $C_{U_i} = h(A_{U_i} \| \sigma_{U_i} \| \beta)$. In addition, the RA also stores the information $\{(TID_{U_i}, PID_{U_i})\}$ related to all registered users U_i under the AP_l in the secure/tamper-resistant database of AP_l.

Various credentials that are preloaded into SD_j, AP_l and MD_{U_i} are provided in Fig. 2.

3.2. Login and authentication phase

If a regal registered user U_i wants to access the real time data directly from a registered smart device SD_j, the following steps are executed:

Step LA_1*:* U_i first inputs his/her identity ID_{U_i} and password PW'_{U_i}, and also imprints biometric Bio'_{U_i} at the sensor of MD_{U_i}.

MD_{U_i} calculates $\sigma_{U_i} = Rep(Bio'_{U_i}, \tau_{U_i})$, and forms a symmetric key $K'_{U_i} = h(ID_{U_i} ||\sigma_{U_i} ||PW'_{U_i})$. MD_{U_i} then decrypts the encrypted information using K'_{U_i} as $I = \{A_{U_i}, B^*_{U_i}, \beta, PID_{U_i}, PID_{AP_l}\} = D_{K'_{U_i}}[E_{h(ID_{U_i}||\sigma_{U_i}||PW_{U_i})}(I)]$. MD_{U_i} calculates $C'_{U_i} = h(A_{U_i} ||\sigma_{U_i} ||\beta)$, and checks if $C'_{U_i} = C_{U_i}$ holds or not. If the condition holds, U_i is authentic and MD_{U_i} further computes $RPW'_{U_i} = h(PW'_{U_i} ||\sigma_{U_i} ||\beta)$ and $X_{U_i} = h(MK_{RA} ||RTS_{U_i} ||ID_{RA}) = B^*_{U_i} \oplus RPW'_{U_i}$.

Step LA_2: MD_{U_i} generates a current timestamp TS_{U_i} and a random secret r_{U_i}, and computes $Y_{U_i} = h(A_{U_i} ||X_{U_i} ||RPW'_{U_i} ||r_{U_i}) \oplus h(f_l(PID_{U_i}, PID_{AP_l}) ||TS_{U_i})$, and $Z_{U_i} = h(TID_{U_i} ||Y_{U_i} ||f_l(PID_{U_i}, PID_{AP_l}) ||ID_{SD_j} ||TS_{U_i})$. Next, MD_{U_i} sends the login message $Msg_1 = \{TID_{U_i}, ID_{SD_j}, Y_{U_i}, Z_{U_i}, TS_{U_i}\}$ to AP_l via public channel.

Step LA_3: After receiving the message Msg_1 at time $TS^*_{U_i}$, AP_l checks the validity of timestamp by $|TS^*_{U_i} - TS_{U_i}| < \Delta T$, where ΔT is the maximum transmission delay. If it is valid, AP_l fetches PID_{U_i} corresponding TID_{U_i} from its database. AP_l then computes $f_l(PID_{AP_l}, PID_{U_i})$ and $Z'_{U_i} = h(TID_{U_i} ||Y_{U_i} ||f_l(PID_{AP_l}, PID_{U_i}) ||ID_{SD_j} ||TS_{U_i})$. After that, AP_l checks if $Z'_{U_i} = Z_{U_i}$ hold or not. If it is so, AP_l computes $M_1 = h(A_{U_i} ||X_{U_i} ||RPW'_{U_i} ||r_{U_i}) = Y_{U_i} \oplus h(f_l(PID_{AP_l}, PID_{U_i}) ||TS_{U_i})$.

Step LA_4: AP_l generates current timestamp TS_{AP_l} and fetches TC_{SD_j} corresponding to the received ID_{SD_j} from U_i. AP_l computes $M_2 = h(M_1 ||PID_{U_i} ||PID_{AP_l}) \oplus h(TC_{SD_j} ||ID_{SD_j} ||TS_{AP_l})$, generates a new random temporary identity $TID^{new}_{U_i}$ for U_i and calculates $M_3 = TID^{new}_{U_i} \oplus h(TID_{U_i} ||PID_{U_i} ||PID_{AP_l} ||f_l(PID_{AP_l}, PID_{U_i}))$. AP_l then constructs the message $Msg_2 = \{ID_{SD_j}, M_2, M_3, TS_{AP_l}\}$ and sends it to SD_j via public channel.

Step LA_5: After getting the message at time $TS^*_{AP_l}$, SD_j verifies the timestamp by $|TS^*_{AP_l} - TS_{AP_l}| < \Delta T$. If it is valid, SD_j computes $M_4 = h(M_1 ||PID_{U_i} ||PID_{AP_l}) = M_2 \oplus h(TC_{SD_j} ||ID_{SD_j} ||TS_{AP_l})$, and generates a random secret r_{SD_j} and current timestamp TS_{SD_j} to calculate $M_5 = h(ID_{SD_j} ||TC_{SD_j} ||r_{SD_j})$, session key $SK_{SD_j,U_i} = h(M_4 ||M_5 ||TS_{SD_j})$ shared with the user U_i, $M_6 = M_5 \oplus h(M_4 ||ID_{SD_j} ||TS_{SD_j})$, session key verifier $SKV_{SD_j,U_i} = h(SK_{SD_j,U_i} ||TS_{SD_j})$ and $M_7 = M_3 \oplus h(M_5 ||TS_{SD_j})$. SD_j then constructs the message $Msg_3 = \{ID_{SD_j}, M_6, M_7, SKV_{SD_j,U_i}, TS_{SD_j}\}$ and sends it to U_i via open channel.

Step LA_6: After receiving Msg_3 at time $TS^*_{SD_j}$, U_i verifies the timestamp by $|TS^*_{SD_j} - TS_{SD_j}| < \Delta T$. If the condition is valid, U_i proceeds to calculate $M_8 = h(h(A_{U_i} ||X_{U_i} ||RPW'_{U_i} ||r_{U_i}) ||PID_{U_i} ||PID_{AP_l})$, $M_9 = M_6 \oplus h(M_8 ||ID_{SD_j} ||TS_{SD_j})$, session key $SK_{U_i,SD_j} = h(M_8 ||M_9 ||TS_{SD_j})$, and session key verifier $SKV_{U_i,SD_j} = h(SK_{U_i,SD_j} ||TS_{SD_j})$. Next, U_i checks if $SK_{U_i,SD_j} = SK_{SD_j,U_i}$, and if it is valid then U_i derives $TID^{new}_{U_i} = M_7 \oplus h(M_9 ||TS_{SD_j}) \oplus h(TID_{U_i} ||PID_{U_i} ||PID_{AP_l} ||f_l(PID_{U_i}, PID_{AP_l}))$ and updates TID_{U_i} by $TID^{new}_{U_i}$ in MD_{U_i}. Also, AP_l updates TID_{U_i} with $TID^{new}_{U_i}$ in its secure database.

The overall phase is finally briefed in Fig. 3.

3.3. User password/biometric update phase

To update password/bioemtrics, a legal registered user U_i follows the following steps with his/her mobile device MD_{U_i} without contacting the *RA*:

Step PBU_1: U_i first inputs his/her identity ID_{U_i} and old password $PW^o_{U_i}$, and also imprints current biometric $Bio^o_{U_i}$ at the sensor of MD_{U_i}. MD_{U_i} calculates $\sigma^o_{U_i} = Rep(Bio^o_{U_i}, \tau_{U_i})$ and decrypts the encrypted information using the calculated symmetric key $K'_{U_i} = h(ID_{U_i} ||\sigma^o_{U_i} ||PW^o_{U_i})$ as $I = \{A_{U_i}, B^*_{U_i}, \beta, PID_{U_i}, PID_{AP_l}\} = D_{K'_{U_i}}[E_{h(ID_{U_i}||\sigma_{U_i}||PW_{U_i})}(I)]$. MD_{U_i} calculates $C^o_{U_i} = h(A_{U_i} ||\sigma^o_{U_i} ||\beta)$, and checks if $C^o_{U_i} = C_{U_i}$ holds or not. If the condition holds, U_i is authentic and MD_{U_i} asks U_i to input his/her new password and biometric.

Step PBU_2: U_i inputs his/her new password, say $PW^n_{U_i}$ and imprints biometric, say $Bio^n_{U_i}$ at the sensor of the mobile device MD_{U_i}. MD_{U_i} then calculates $Gen(Bio^n_{U_i}) = (\sigma^n_{U_i}, \tau^n_{U_i})$, $RPW^o_{U_i} = h(PW^o_{U_i} ||\sigma^o_{U_i} ||\beta)$, $RPW^n_{U_i} = h(PW^n_{U_i} ||\sigma^n_{U_i} ||\beta)$, $B^n_{U_i} = B^*_{U_i} \oplus (RPW^o_{U_i} \oplus RPW^n_{U_i}) = h(MK_{RA} ||RTS_{U_i} ||ID_{RA}) \oplus RPW^n_{U_i}$, $C^n_{U_i} = h(A_{U_i} ||\sigma^n_{U_i} ||\beta)$ and $I' = \{A_{U_i}, B^n_{U_i}, \beta, PID_{U_i}, PID_{AP_l}\}$.

Step PBU_3: Finally, MD_i replaces $\{E_{h(ID_{U_i}||\sigma_{U_i}||PW_{U_i})}(I), \tau_{U_i}, C_{U_i}\}$ with $\{E_{h(ID_{U_i}||\sigma^n_{U_i}||PW^n_{U_i})}(I'), \tau^n_{U_i}, C^n_{U_i}\}$ in its memory.

Finally, the user password/biometric update phase is also summarized in Fig. 4.

3.4. Dynamic IoT smart device addition phase

Sometimes IoT smart devices may be physical capture by an adversary (as stated in the threat model in Section 1.2) and also they may be power exhausted. Hence, there is a need to deploy some new smart devices in the existing network. To deploy a new smart device under an existing AP_l, the *RA* needs to follow the same steps as described in Section 3.1.1.

4. Security analysis

We show that the proposed scheme is resilient against the following potential attacks.

4.1. Replay attack

In the login and authentication phase in Section 3.2, three messages $Msg_1 = \{TID_{U_i}, ID_{SD_j}, Y_{U_i}, Z_{U_i}, TS_{U_i}\}$, $Msg_2 = \{ID_{SD_j}, M_2, M_3, TS_{AP_l}\}$, and $Msg_3 = \{ID_{SD_j}, M_6, M_7, SKV_{SD_j,U_i}, TS_{SD_j}\}$ are communicated via public channel. In each message, timestamp and random nonce are injected. If an adversary $\mathcal{A}$ tries to replay the old messages, the receiver can easily detect the old replay messages by checking the current timestamp. Therefore, the proposed scheme is resilient against the "replay attack".

4.2. Man-in-the-middle (MiTM) attack

An adversary $\mathcal{A}$ captures the message Msg_1 on the fly from the public open channel and tries to construct another valid request message, say Msg'_1. For constructing a valid authentication request message Msg'_1, $\mathcal{A}$ needs to compute $RPW'_{U_i} = h(PW'_{U_i} ||\sigma_{U_i} || \beta)$, Y_{U_i}, and Z_{U_i}. Since each value is generated using the random nonce and permanent secrets $\{PW'_{U_i}, \sigma_{U_i}, PID_{U_i}, PID_{AP_l}\}$, $\mathcal{A}$ can not derive the password with polynomial time and the others secrets. Similarly, for other messages Msg_2 and Msg_3, $\mathcal{A}$ can not also create other valid messages without long term secrets. Hence, the proposed scheme is protected from the MiTM attack.

4.3. Impersonation attacks

In this attack, an adversary $\mathcal{A}$ can impersonate as an authorized party during the login and authentication process. For user impersonation attack, $\mathcal{A}$ may pretend as a legitimate user U_i and try to communicate with a valid message. To achieving this goal, $\mathcal{A}$ may pick a random nonce r'_{U_i} and timestamp TS'_{U_i} to construct a valid authentication request message Msg_1. Here, $RPW'_{U_i} = h(PW'_{U_i} ||\sigma_{U_i} ||\beta)$, $X_{U_i} = B^*_{U_i} \oplus RPW'_{U_i}$, $Y_{U_i} = h(A_{U_i} ||X_{U_i} ||RPW'_{U_i} ||r_{U_i}) \oplus h(f_l(PID_{U_i}, PID_{AP_l}) ||TS_{U_i})$, and $Z_{U_i} = h(TID_{U_i} ||Y_{U_i} ||f_l(PID_{U_i}, PID_{AP_l}) ||ID_{SD_j} ||TS_{U_i})$. Without any knowledge of the permanent secrets $\{PW'_{U_i}, \sigma_{U_i}, PID_{U_i}, PID_{AP_l}\}$, $\mathcal{A}$ will not be able to construct the legal

User (U_i)	Access Point (AP_l)	IoT Smart Device (SD_j)
$\{E_{h(ID_{U_i}\|\|\sigma_{U_i}\|\|PW_{U_i})}(I), \tau_{U_i}, h(\cdot),$ $Gen(\cdot), Rep(\cdot), C_{U_i}, et, f_l(PID_{U_i}, y)\}$	$\{PID_{AP_l}, \{(ID_{SD_j}, TC_{SD_j}) \mid j = 1, 2, \cdots, n_{sd}\}$ $(TID_{U_i}, PID_{U_i}), f_l(PID_{AP_l}, y), h(\cdot)\}$	$\{ID_{SD_j}, TC_{SD_j}\}$
Input: ID_{U_i}, PW'_{U_i} and Bio'_{U_i}		
MD_{U_i} calculates $\sigma_{U_i} = Rep(Bio'_{U_i}, \tau_{U_i})$ $K'_{U_i} = h(ID_{U_i} \|\|\sigma_{U_i} \|\|PW'_{U_i})$ and decrypts $I = \{A_{U_i}, B^*_{U_i}, \beta, PID_{U_i}, PID_{AP_l}\} =$ $D_{K'_{U_i}}[E_{h(ID_{U_i}\|\|\sigma_{U_i}\|\|PW_{U_i})}(I)]$. MD_{U_i} calculates $C'_{U_i} = h(A_{U_i} \|\|\sigma_{U_i} \|\|\beta)$, checks if $C'_{U_i} = C_{U_i}$? If valid, U_i is authenticated.		
Computes $RPW'_{U_i} = h(PW'_{U_i} \|\|\sigma_{U_i} \|\|\beta)$, $X_{U_i} = h(MK_{RA} \|\|RTS_{U_i} \|\|ID_{RA}) = B^*_{U_i} \oplus RPW'_{U_i}$. Generates timestamp TS_{U_i} and random secret r_{U_i}. Computes $Y_{U_i} = h(A_{U_i} \|\|X_{U_i} \|\|RPW'_{U_i} \|\|r_{U_i}) \oplus$ $h(f_l(PID_{U_i}, PID_{AP_l}) \|\|TS_{U_i})$, and $Z_{U_i} = h(TID_{U_i}$ $\|\|Y_{U_i} \|\|f_l(PID_{U_i}, PID_{AP_l}) \|\|ID_{SD_j} \|\|TS_{U_i})$. $Msg_1 = \{TID_{U_i}, ID_{SD_j}, Y_{U_i}, Z_{U_i}, TS_{U_i}\}$ $\longrightarrow$	Checks if $\|TS^*_{U_i} - TS_{U_i}\| < \Delta T$? Accept/Reject? Fetches PID_{U_i} corresponding to TID_{U_i}. Computes $f_l(PID_{AP_l}, PID_{U_i})$, $Z'_{U_i} = h(TID_{U_i}$ $\|\|Y_{U_i} \|\|f_l(PID_{AP_l}, PID_{U_i}) \|\|ID_{SD_j} \|\|TS_{U_i})$. Checks if $Z'_{U_i} = Z_{U_i}$? If so, computes $M_1 = h(A_{U_i} \|\|X_{U_i} \|\|RPW'_{U_i} \|\|r_{U_i})$ $= Y_{U_i} \oplus h(f_l(PID_{AP_l}, PID_{U_i}) \|\|TS_{U_i})$. Generates timestamp TS_{AP_l}, fetches TC_{SD_j}, computes $M_2 = h(M_1 \|\|PID_{U_i} \|\|PID_{AP_l}) \oplus h(TC_{SD_j}$ $\|\|ID_{SD_j} \|\|TS_{AP_l})$, and picks $TID^{new}_{U_i}$ for U_i, $M_3 = TID^{new}_{U_i} \oplus h(TID_{U_i} \|\|PID_{U_i}$ $\|\|PID_{AP_l} \|\|f_l(PID_{AP_l}, PID_{U_i}))$. $Msg_2 = \{ID_{SD_j}, M_2, M_3, TS_{AP_l}\}$ $\longrightarrow$	Checks if $\|TS^*_{AP_l} - TS_{AP_l}\| < \Delta T$? Accept/Reject? Computes $M_4 = h(M_1 \|\|PID_{U_i} \|\|PID_{AP_l}) =$ $M_2 \oplus h(TC_{SD_j} \|\|ID_{SD_j} \|\|TS_{AP_l})$. Generates r_{SD_j} and current timestamp TS_{SD_j}. Calculates $M_5 = h(ID_{SD_j} \|\|TC_{SD_j} \|\|r_{SD_j})$, session key $SK_{SD_j,U_i} = h(M_4 \|\|M_5 \|\|TS_{SD_j})$, $M_6 = M_5 \oplus h(M_4 \|\|ID_{SD_j} \|\|TS_{SD_j})$, $SKV_{SD_j,U_i} = h(SK_{SD_j,U_i} \|\|TS_{SD_j})$, and $M_7 = M_3 \oplus h(M_5 \|\|TS_{SD_j})$. $Msg_3 = \{ID_{SD_j}, M_6, M_7, SKV_{SD_j,U_i}, TS_{SD_j}\}$ $\longleftarrow$
Checks if $\|TS^*_{SD_j} - TS_{SD_j}\| < \Delta T$? Accept/Reject? Computes $M_8 = h(h(A_{U_i} \|\|X_{U_i} \|\|RPW'_{U_i} \|\|r_{U_i})$ $\|\|PID_{U_i} \|\|PID_{AP_l})$, $M_9 = M_6 \oplus h(M_8 \|\|ID_{SD_j} \|\|TS_{SD_j})$, $SK_{U_i,SD_j} = h(M_8 \|\|M_9 \|\|TS_{SD_j})$, and $SKV_{U_i,SD_j} =$ $h(SK_{U_i,SD_j} \|\|TS_{SD_j})$. Checks if $SK_{U_i,SD_j} = SK_{SD_j,U_i}$? If valid, derives $TID^{new}_{U_i} = M_7 \oplus h(M_9 \|\|TS_{SD_j})$ $\oplus h(TID_{U_i} \|\|PID_{U_i} \|\|PID_{AP_l} \|\|f_l(PID_{U_i}, PID_{AP_l}))$. Updates TID_{U_i} by $TID^{new}_{U_i}$ in MD_{U_i}.	Updates TID_{U_i} by $TID^{new}_{U_i}$ in its secure database.	
Shared secret session key $SK_{U_i,SD_j} (= SK_{SD_j,U_i})$ between U_i and SD_j		

Fig. 3. Summary of login and authentication phase.

User (U_i)	User Mobile Device (MD_{U_i})
Inputs identity ID_{U_i}, old password $PW^o_{U_i}$. Imprints current biometric $Bio^o_{U_i}$ at MD_{U_i}'s sensor.	Compute $\sigma^o_{U_i} = Rep(Bio^o_{U_i}, \tau_{U_i})$. Compute $K'_{U_i} = h(ID_{U_i} \|\|\sigma^o_{U_i} \|\|PW^o_{U_i})$. Retrieve $I = \{A_{U_i}, B^*_{U_i}, \beta, PID_{U_i}, PID_{AP_l}\}$ $= D_{K'_{U_i}}[E_{h(ID_{U_i}\|\|\sigma_{U_i}\|\|PW_{U_i})}(I)]$. Compute $C^o_{U_i} = h(A_{U_i} \|\|\sigma^o_{U_i} \|\|\beta)$. Check if $C^o_{U_i} = C_{U_i}$? If it holds, ask U_i to input new password and biometric.
Input new password $PW^n_{U_i}$. Imprint new biometric $Bio^n_{U_i}$ at MD_{U_i}'s sensor.	Compute $Gen(Bio^n_{U_i}) = (\sigma^n_{U_i}, \tau^n_{U_i})$, $RPW^o_{U_i} = h(PW^o_{U_i} \|\|\sigma^o_{U_i} \|\|\beta)$, $RPW^n_{U_i} = h(PW^n_{U_i} \|\|\sigma^n_{U_i} \|\|\beta)$, $B^n_{U_i} = B^*_{U_i} \oplus (RPW^o_{U_i} \oplus RPW^n_{U_i})$, $C^n_{U_i} = h(A_{U_i} \|\|\sigma^n_{U_i} \|\|\beta)$, $I' = \{A_{U_i}, B^n_{U_i}, \beta, PID_{U_i}, PID_{AP_l}\}$. Replace $\{E_{h(ID_{U_i}\|\|\sigma_{U_i}\|\|PW_{U_i})}(I), \tau_{U_i},$ $C_{U_i}\}$ with $\{E_{h(ID_{U_i}\|\|\sigma^n_{U_i}\|\|PW^n_{U_i})}(I'),$ $\tau^n_{U_i}, C^n_{U_i}\}$ in its memory.

Fig. 4. Summary of user password/biometric update phase.

message. Similarly, the access point and IoT smart device impersonation attacks are impossible for $\mathcal{A}$. Thus, "user impersonation attack" and "access point and IoT smart device impersonation attacks" are protected in the proposed scheme.

4.4. IoT smart device physical capture attack

Due to hostile environment IoT smart devices may physically captured by an adversary $\mathcal{A}$. As mention in the attack model in Section 1.2, $\mathcal{A}$ can utilize power analysis to extract all the stored information $\{TC_{SD_j}, ID_{SD_j}, h(\cdot)\}$ from smart device's memory. Since these credentials are unique and distinct to all other devices, by compromising a smart device $\mathcal{A}$ can not compromise the session keys between a user and other non-compromising smart devices. Therefore, the proposed scheme is secure against "device physical capture attack".

4.5. Privileged-insider and stolen mobile device attacks

During user registration, a user U_i sends a registration request $RReq_{U_i} = \{PID_{U_i}, RPW_{U_i} \oplus \gamma\}$ to RA via secure channel. Since PID_{U_i} and RPW_{U_i} contain user biometric secret information and password, and these are protected by cryptographic hash function $h(\cdot)$, $\mathcal{A}$ can not find these secret credentials. If $\mathcal{A}$ has the U_i's mobile device MD_{U_i}, without knowledge of the secret information, such as PW_{U_i}, σ_{U_i}, α, β and γ, he/she can not succeed. The information $\{E_{h(ID_{U_i}||\sigma_{U_i}||PW_{U_i})}(I), \tau_{U_i}, h(\cdot), Gen(\cdot), Rep(\cdot), C_{U_i}, et, f_l(PID_{U_i}, y)\}$ can be extracted from the MD'_{U_i}s memory using power analysis as mention in attack model in Section 1.2. To guess a correct password PW_{U_i} from I, he/she first needs to decrypt I that requires the symmetric key K_{U_i}. To correctly construct K_{U_i}, $\mathcal{A}$ needs to correctly guess user biometric secret σ_{U_i} and identity ID_{U_i} simultaneously. Thus, the proposed scheme is secure against "privileged-insider and stolen mobile device attacks".

4.6. Ephemeral secret leakage (ESL) attack

In the login and authentication phase, a legal user U_i constructs a session key SK_{U_i,SD_j} $(= SK_{SD_j,U_i})$ shared with a registered smart device SD_j as $SK_{U_i,SD_j} = h(M_8 \|\|M_9 \|\|TS_{SD_j})$, where $M_8 = h(h(A_{U_i} \|\|X_{U_i} \|\|RPW'_{U_i} \|\|r_{U_i}) \|\|PID_{U_i} \|\|PID_{AP_l})$ and $M_9 = M_6 \oplus h(M_8 \|\|ID_{SD_j} \|\|TS_{SD_j})$. The construction of SK_{U_i,SD_j} for a particular session required both ephemeral secret (short term secrets), such as random nonce (secret) r_{U_i}, and long term secrets PW_{U_i}, σ_{U_i}, PID_{U_i}, and master keys. If an adversary $\mathcal{A}$ wants to compute the session key for a

particular session, he/she needs to know both ephemeral and long term secrets. Since random nonce and current timestamp are injected into a session key, the session keys are different for different sessions. Thus, disclosure of a session key for a particular session cannot effect to the other session keys in different sessions. As a result, "perfect forward and backward secrecy" properties are preserved. This shows that the proposed scheme is secure against the "ESL attack under CK-adversary model".

4.7. Password change attack

Suppose an adversary $\mathcal{A}$ has a stolen mobile device MD_{U_i} of a register user U_i. He/she can extract all stored credentials $\{E_{h(ID_{U_i}||\sigma_{U_i}||PW_{U_i})}(I),\ \tau_{U_i},\ h(\cdot),\ Gen(\cdot),\ Rep(\cdot),\ C_{U_i},\ et,\ f_l(PID_{U_i}, y)\}$ using power analysis attack. Now, if $\mathcal{A}$ wants to change or update the password PW_{U_i} of U_i, $\mathcal{A}$ needs to guess correctly the old password and biometric secret σ_{U_i}. However, to decrypt the encrypted information $E_{h(ID_{U_i}||\sigma_{U_i}||PW_{U_i})}(I)$ it requires the key $K_{U_i} = h(ID_{U_i}||\sigma_{U_i}||PW_{U_i})$. However, it is a "computationally infeasible task" to guess simultaneously all the secrets (identity, password and biometric key). Hence, the proposed scheme is not vulnerable to "password change attack".

4.8. User anonymity and untraceability

Assume an adversary $\mathcal{A}$ hijacks the login and authentication messages Msg_1, Msg_2 and Msg_3 through communication channels. Each communicated message is not injected directly with the real identity ID_{U_i} of a register user U_i. Instead of ID_{U_i}, a temporary identity TID_{U_i} of U_i is used, and in each session it is also updated. Thus, $\mathcal{A}$ cannot identify the same in subsequent sessions for U_i. Since each communicated message is created with random nonce and current timestamp, in every session the messages are dynamic in nature. Therefore, $\mathcal{A}$ cannot differentiate whether two login and authentication request messages are same or not. Thus, the proposed scheme satisfies both anonymity and untrecability properties.

5. Formal security verification using AVISPA: simulation study

In recent years, formal security verification using automated software validation tools becomes popular in the security domain. Among other automated software validation tools, "Automated Validation of Internet Security Protocols and Applications (AVISPA)" [33] is a very popular tool that has been used in many authentication schemes, such as [22–29]. AVISPA has four backends, which are OFMC, CL-AtSe, SATMC and TA4SP [33]. Out of these four backends, OFMC and CL-ATse are very popular as they support implementation of "bitwise XOR operation" as compared to other backends: SATMC and TA4SP. A tested security protocol is required to be implemented using the "High Level Protocol Specification Language (HLPSL)" which is a role-oriented language by its design. More details on AVISPA and its HLPSL can be found in [33].

In our implementation, we have considered registration and login & authentication phases. There are three basic roles for a user U_i, an access point AP_l and an IoT smart device SD_j, apart from usual two mandatory roles for the "session" and "goal and environment". We have then simulated the proposed using the "Security Protocol ANimator for AVISPA (SPAN)" tool [34]. The simulation results under the popularly used OFMC and CL-AtSe backends are demonstrated in Fig. 5. Note that other backends: SATMC and TA4SP do not presently support bitwise XOR operations in the HLPSL implementation. The proposed scheme relies on bitwise XOR operations. Due to this reason, we have omitted the simulation results of the proposed scheme under the SATMC and TA4SP backends because the results will be "inconclusive" in this scenario.

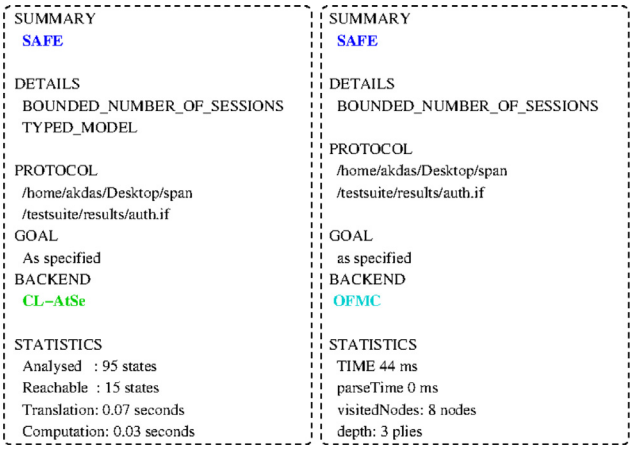

Fig. 5. AVISPA simulation results under CL-AtSe and OFMC backends.

AVISPA implements the "Dolev-Yao threat model (DY model)" [6] and as a result, an intruder which is always denoted by i in AVISPA, not only can eavesdrop the communicated messages, but can also delete, modify or insert the fake message contents in between the communication of the entities in a network. Under the OFMC backend simulation, 44 milliseconds (ms) was required and there were 8 visited nodes and 3 plies of depth. On the other side, CL-AtSe backend simulation analyzed 95 states and out of these states, 15 states were reachable, and it needed a translation time of 0.07 s and the computation time of 0.03 s. The results illustrated in Fig. 5 clearly exhibit that the proposed scheme is secure against both "replay" and "man-in-the-middle" attacks.

6. Comparative analysis

In this section, we supply the performance analysis of the proposed scheme based on "communication and computation costs" for the login and authentication phase shown in Fig. 3. In addition, we also provide a detailed comparative analysis on "communication and computation costs" and "security and functionality features" among the proposed scheme and other existing relevant schemes, such as the schemes described by Dhillon and Sheetal [13], Alotaibi [14] and Li et al. [19].

We first evaluate cryptographic primitives using the widely-accepted "Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL)" [35] for computing the execution time. MIRACL, a C/C++ based programming software library, is a popular software-based tool used by the cryptographers as the "gold standard open source SDK for elliptic curve cryptography (ECC)".

Assume T_{ecm}, T_{eca}, T_{senc}/T_{sdec}, T_h, T_{mul} and T_{add} denote the execution time for "elliptic curve point (scalar) multiplication", "elliptic curve point addition", "symmetric key (Advanced Encryption Standard (AES-128)) encryption/decryption", "one-way hash function using SHA-256 hashing algorithm", "modular multiplication over a finite field" and "modular addition over a finite field", respectively. We have used a non-singular elliptic curve of the type: "$y^2 = x^3 + ux + v \pmod p$" with $u, v \in Z_p$. for the elliptic curve point addition and multiplication.

In the following, we consider the following two platforms:

- The first platform we have considered for MIRACL is on a server setting running on Ubuntu 18.04.4 LTS, with memory: 7.7 GiB, processor: Intel® Core™ i7-8565U CPU @ 1.80GHz × 8, OS type: 64-bit and disk: 966.1 GB. We have run each experiment for a cryptographic primitive for 100 iterations. After that

Table 2
Execution time (in ms) of cryptographic primitives using MIRACL under a server setting.

Primitive	Max. time (ms)	Min. time (ms)	Average time (ms)
T_h	0.149	0.024	0.055
T_{ecm}	2.998	0.284	0.674
T_{eca}	0.002	0.001	0.002
T_{senc}	0.003	0.001	0.001
T_{sdec}	0.002	0.001	0.001
T_{mul}	0.007	0.001	0.002
T_{add}	0.003	0.001	0.001

Table 3
Execution time (in ms) of cryptographic primitives using MIRACL under Raspberry PI 3 setting.

Primitive	Max. time (ms)	Min. time (ms)	Average time (ms)
T_h	0.643	0.274	0.309
T_{ecm}	4.532	2.206	2.288
T_{eca}	0.021	0.015	0.016
T_{senc}	0.038	0.017	0.018
T_{sdec}	0.054	0.009	0.014
T_{mul}	0.016	0.009	0.011
T_{add}	0.013	0.008	0.010

the maximum, minimum and average run-time in milliseconds are calculated for each cryptographic primitive from these 100 runs. Table 2 tabulates the experimental results in this platform.

- The second platform we have considered for MIRACL is on a user's mobile device or an IoT smart device setting running on "Raspberry PI 3 B+ Rev 1.3, with CPU: 64-bit, Processor: 1.4 GHz Quad-core, 4 cores, Memory (RAM): 1GB, and OS: Ubuntu 20.04 LTS, 64-bit [36]". We have also run each experiment for a cryptographic primitive for 100 iterations. After that the maximum, minimum and average run-time in milliseconds are measured for each cryptographic primitive from these 100 runs. Finally, Table 3 tabulates the experimental results in this platform.

We assume that T_{fe}, T_{poly} and T_{ph} denote the execution time needed for "fuzzy extractor" $Gen(\cdot)/Rep(\cdot)$ function, evaluating a t-degree polynomial over a finite field $GF(p)$, and "perceptual hashing operation", respectively. It is further assume that $T_{fe} \approx T_{ecm}$ [37] and $T_{ph} \approx T_{ecm}$. If the Horner's rule [38] is applied, the evaluation of a t-degree uni-variate polynomial requires t "modular multiplications" and t "modular additions", that is, $T_{poly} = t(T_{mul} + T_{add})$. In Li et al.'s scheme [19], T_f and T_{inv} denote the "conversion function" and "inverse conversion function" of the Bose–Chaudhuri–Hocquenghem (BCH) code [39], and it is assumed that $T_f \approx T_h$ and $T_{inv} \approx T_h$.

For computation costs comparison among the proposed schemes and other existing schemes, we use the average execution time (in milliseconds) of primitives that are reported from our experiments in Tables 2 and 3 for an access point and a user/smart device, respectively. we have considered the login and authentication phases. In our proposed scheme, the total computation cost of a user U_i and an IoT smart device is $20T_h + T_{fe} + T_{poly} \approx 10.568$ ms, if $t = 100$, and that for an access point is $6T_h \approx 0.33$ ms. The comparative study reported in Table 4 shows that the proposed scheme performance is comparable with other schemes with respect to computational costs.

For communication cost analysis, the "identity", "random nonce", "cryptographic hash function (if SHA-256 hash algorithm is applied)", and "timestamp" are considered as 160, 160, 256 and 32 bits, respectively. In proposed scheme, the communication cost due to three messages Msg_1, Msg_2 and Msg_3 need 864 bits, 704 bits and 960 bits respectively, which altogether require 2528 bits. The comparative analysis on communication costs in terms of number of messages and bits required for transmitting the messages among the considered schemes in Table 5 exhibits that the proposed scheme requires low communication cost as compared to other existing relevant schemes [13,14,19].

Table 4
Comparison of computation costs.

Protocol	User/Smart device	Access point
Proposed	$20T_h + T_{fe} + T_{poly}$ ≈ 10.568 ms	$6T_h$ ≈ 0.33 ms
Dhillon and Sheetal [13]	$13T_h + T_{ph}$ ≈ 6.305 ms	$7T_h$ ≈ 0.385 ms
Alotaibi [14]	$12T_h + 2T_{fe} + 2T_{senc/sdec}$ ≈ 8.316 ms	$6T_h + 2T_{senc/sdec}$ ≈ 0.332 ms
Li et al. [19]	$14T_h + 5T_{ecm} + T_f + T_{inv}$ ≈ 18.363 ms	$8T_h + T_{ecm}$ ≈ 1.114 ms

Table 5
Comparison of communication costs.

Protocol	No. of messages	Total cost (in bits)
Proposed	3	2528
Dhillon and Sheetal [13]	4	4832
Alotaibi [14]	4	3584
Li et al. [19]	4	3584

Table 6
Comparison of security & functionality attributes.

Attribute	Dhillon and Sheetal [13]	Alotaibi [14]	Li et al. [19]	Proposed
SFA_1	✓	✓	✓	✓
SFA_2	✓	✓	✓	✓
SFA_3	×	×	×	✓
SFA_4	×	×	×	✓
SFA_5	✓	✓	✓	✓
SFA_6	×	×	×	✓
SFA_7	×	✓	×	✓
SFA_8	✓	✓	✓	✓
SFA_9	✓	✓	×	✓
SFA_{10}	×	✓	✓	✓
SFA_{11}	✓	✓	✓	✓

✓: "a scheme supports an attribute or resists an attack"; × : "a scheme does not support an attribute or it does not resist an attack"; *N/A*: "not applicable in a scheme'. SFA_1: "replay attack"; SFA_2: "man-in-the-middle attack"; SFA_3: "privileged insider attack"; SFA_4: "user anonymity and/or traceability"; SFA_5: "IoT smart device physical capture attack"; SFA_6: "ESL attack under CK-adversary model"; SFA_7: "mobile device stolen/lost attack"; SFA_8: "user/gateway node/server impersonation attack"; SFA_9: "formal security verification under AVISPA tool"; SFA_{10}: "support to dynamic node addition phase"; SFA_{10}: "password/biometric update phase".

Finally, in Table 6, we compare the performance of the proposed scheme and other schemes [13,14,19] for the considered eleven "security and functionality attributes" (SFA_1–SFA_{11}). It is clear that the proposed scheme provides significantly better security features and more functionality features as compared to other relevant schemes.

7. Concluding remarks

In this work, we attempted to design an important security service, namely user authentication, needed for securing a smart city environment. A user having mobile device can directly access the real time data from accessed IoT smart devices for which the user is allowed provided that a mutual authentication is successful. A registered user is allowed to update his/her password/biometric at any time after registration locally without the help of the *RA*. In addition, IoT smart devices can be dynamically added in the network after initial deployment. A detailed security analysis reveals that the proposed scheme is secure against several potential

attacks including user anonymity and untraceability properties. Furthermore, comparative study shows that the proposed scheme performs significantly better than other schemes in order to deploy in smart city scenario.

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.

Acknowledgments

The authors would like to thank the anonymous reviewers and the Associate Editor for their valuable feedback on the paper which helped us to improve its quality as well as presentation.

References

[1] M. Sookhak, H. Tang, Y. He, F.R. Yu, Security and privacy of smart cities: a survey, research issues and challenges, IEEE Commun. Surv. Tutor. 21 (2) (2019) 1718–1743.
[2] A. Solanas, C. Patsakis, M. Conti, I.S. Vlachos, V. Ramos, F. Falcone, et al., Smart health: a context-aware health paradigm within smart cities, IEEE Commun. Mag. 52 (8) (2014) 74–81.
[3] A. Gharaibeh, M.A. Salahuddin, S.J. Hussini, A. Khreishah, I. Khalil, M. Guizani, et al., Smart cities: a survey on data management, security, and enabling technologies, IEEE Commun. Surv. Tutor. 19 (4) (2017) 2456–2501.
[4] S. Barra, A. Castiglione, M. De Marsico, M. Nappi, K.R. Choo, Cloud-based biometrics (biometrics as a service) for smart cities, nations, and beyond, IEEE Cloud Comput. 5 (5) (2018) 92–100.
[5] S. Barra, A. Castiglione, F. Narducci, M. De Marsico, M. Nappi, Biometric data on the edge for secure, smart and user tailored access to cloud services, Future Gener. Comput. Syst. 101 (2019) 534–541.
[6] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inf. Theory 29 (2) (1983) 198–208.
[7] R. Canetti, H. Krawczyk, Universally composable notions of key exchange and secure channels, in: International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'02), Amsterdam, The Netherlands, 2002, pp. 337–351.
[8] T.S. Messerges, E.A. Dabbish, R.H. Sloan, Examining smart-card security under the threat of power analysis attacks, IEEE Trans. Comput. 51 (5) (2002) 541–552.
[9] P. Vijayakumar, M. Azees, A. Kannan, L. Jegatha Deborah, Dual authentication and key management techniques for secure data transmission in vehicular ad hoc networks, IEEE Trans. Intell. Transp. Syst. 17 (4) (2016) 1015–1028.
[10] M. Azees, P. Vijayakumar, L.J. Deboarh, EAAP: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks, IEEE Trans. Intell. Transp. Syst. 18 (9) (2017) 2467–2476.
[11] P. Vijayakumar, V. Chang, L.J. Deborah, B. Balusamy, P. Shynu, Computationally efficient privacy preserving anonymous mutual and batch authentication schemes for vehicular ad hoc networks, Future Gener. Comput. Syst. 78 (2018) 943–955.
[12] J. Shen, D. Liu, X. Chen, J. Li, N. Kumar, P. Vijayakumar, Secure real-time traffic data aggregation with batch verification for vehicular cloud in VANETs, IEEE Trans. Veh. Technol. 69 (1) (2020) 807–817.
[13] P.K. Dhillon, S. Kalra, A lightweight biometrics based remote user authentication scheme for IoT services, J. Inf. Secur. Appl. 34 (2017) 255–270.
[14] M. Alotaibi, An enhanced symmetric cryptosystem and biometric-based anonymous user authentication and session key establishment scheme for WSN, IEEE Access 6 (2018) 70072–70087.
[15] D. Kang, J. Jung, H. Kim, Y. Lee, D. Won, Efficient and secure biometric-based user authenticated key agreement scheme with anonymity, Secur. Commun. Netw. 2018 (2018) 1–14.
[16] S.D. Kaul, A.K. Awasthi, Security enhancement of an improved remote user authentication scheme with key agreement, Wirel. Pers. Commun. 89 (2) (2016) 621–637.
[17] J. Ryu, T. Song, J. Moon, H. Kim, D. Won, Cryptanalysis of improved and provably secure three-factor user authentication scheme for wireless sensor networks, in: Computational Science and Technology, Springer, 2019, pp. 49–58.
[18] F. Wu, L. Xu, S. Kumari, X. Li, An improved and provably secure three-factor user authentication scheme for wireless sensor networks, Peer-to-Peer Netw. Appl. 11 (2018) 1–20.
[19] X. Li, J. Peng, M.S. Obaidat, F. Wu, M.K. Khan, C. Chen, A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems, IEEE Syst. J. 14 (1) (2020) 39–50.
[20] R. Amin, S.H. Islam, G. Biswas, M.K. Khan, N. Kumar, A robust and anonymous patient monitoring system using wireless medical sensor networks, Future Gener. Comput. Syst. 80 (2018) 483–495.
[21] C.C. Chang, H.D. Le, A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks, IEEE Trans. Wirel. Commun. 15 (1) (2016) 357–366.
[22] M. Wazid, A.K. Das, S. Kumari, X. Li, F. Wu, Design of an efficient and provably secure anonymity preserving three-factor user authentication and key agreement scheme for TMIS, Secur. Commun. Netw. 9 (13) (2016) 1983–2001.
[23] A.K. Das, S. Kumari, V. Odelu, X. Li, F. Wu, X. Huang, Provably secure user authentication and key agreement scheme for wireless sensor networks, Secur. Commun. Netw. 9 (16) (2016) 3670–3687.
[24] A.K. Das, A.K. Sutrala, S. Kumari, V. Odelu, M. Wazid, X. Li, An efficient multi-gateway-based three-factor user authentication and key agreement scheme in hierarchical wireless sensor networks, Secur. Commun. Netw. 9 (13) (2016) 2070–2092.
[25] M. Wazid, A.K. Das, N. Kumar, V. Odelu, A. Goutham Reddy, K. Park, et al., Design of lightweight authentication and key agreement protocol for vehicular ad hoc networks, IEEE Access 5 (2017) 14966–14980.
[26] M. Wazid, A.K. Das, N. Kumar, J.J.P.C. Rodrigues, Secure three-factor user authentication scheme for renewable-energy-based smart grid environment, IEEE Trans. Ind. Inf. 13 (6) (2017) 3144–3153.
[27] S. Challa, A.K. Das, V. Odelu, N. Kumar, S. Kumari, M.K. Khan, et al., An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks, Comput. Electr. Eng. 69 (2018) 534–554.
[28] A. Dua, N. Kumar, A.K. Das, W. Susilo, Secure message communication protocol among vehicles in smart city, IEEE Trans. Veh. Technol. 67 (5) (2017) 4359–4373.
[29] J. Srinivas, A.K. Das, N. Kumar, J.J.P.C. Rodrigues, TCALAS: Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment, IEEE Trans. Veh. Technol. 68 (7) (2019) 6903–6916.
[30] C. Blundo, A.D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung, Perfectly secure key distribution for dynamic conferences, Inf. Comput. 146 (1) (1998) 1–23.
[31] E. Bertino, N. Shang, J.S.S. W., An efficient time-bound hierarchical key management scheme for secure broadcasting, IEEE Trans. Dependable Secure Comput. 5 (2) (2008) 65–70.
[32] Y. Dodis, L. Reyzin, A. Smith, Fuzzy extractors: how to generate strong keys from biometrics and other noisy data, in: Proceedings of the Advances in Cryptology (Eurocrypt'04), LNCS, vol. 3027, 2004, pp. 523–540.
[33] Automated validation of internet security protocols and applications, 2019. http://www.avispa-project.org/. Accessed on October 2019.
[34] SPAN, the security protocol ANimator for AVISPA, 2019. http://www.avispa-project.org/. Accessed on May 2020.
[35] MIRACL cryptographic SDK: multiprecision integer and rational arithmetic cryptographic library, 2020. Accessed on April 2020 https://github.com/miracl/MIRACL.
[36] Raspberry pi 3 model b+, 2020. Accessed on June 2020 https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/.
[37] D. He, S. Zeadally, B. Xu, X. Huang, An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks, IEEE Trans. Inf. Forensics Secur. 10 (2015) 2681–2691.
[38] D.E. Knuth, The Art of Computer Programming: Seminumerical Algorithms, vol. 2, third ed., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.
[39] R. Chien, Cyclic decoding procedures for Bose-Chaudhuri-Hocquenghem codes, IEEE Trans. Inf. Theory 10 (4) (1964) 357–363.