Digital Communications and Networks 7 (2021) 235–244

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.keaipublishing.com/dcan

Lightweight authentication protocol for e-health clouds in IoT-based applications through 5G technology

Minahil [a], Muhammad Faizan Ayub [a], Khalid Mahmood [a], Saru Kumari [b,*], Arun Kumar Sangaiah [c]

[a] Department of Computer Science, COMSATS University Islamabad, Sahiwal Campus, 57000, Pakistan
[b] Department of Mathematics, Ch. Charan Singh University, Meerut, Utter Paradesh, 250001, India
[c] School of Computing Science and Engineering, VIT University, Vellore, Tamil Nadu, India

ARTICLE INFO

Keywords:
Authentication
Security protocol
IoT
5G
e-health cloud

ABSTRACT

Modern information technology has been utilized progressively to store and distribute a large amount of healthcare data to reduce costs and improve medical facilities. In this context, the emergence of e-Health clouds offers novel opportunities, like easy and remote accessibility of medical data. However, this achievement produces plenty of new risks and challenges like how to provide integrity, security, and confidentiality to the highly susceptible e-Health data. Among these challenges, authentication is a major issue that ensures that the susceptible medical data in clouds is not available to illegal participants. The smart card, password and biometrics are three factors of authentication which fulfill the requirement of giving high security. Numerous three-factor ECC-based authentication protocols on e-Health clouds have been presented so far. However, most of the protocols have serious security flaws and produce high computation and communication overheads. Therefore, we introduce a novel protocol for the e-Health cloud, which thwarts some major attacks, such as user anonymity, offline password guessing, impersonation, and stolen smart card attacks. Moreover, we evaluate our protocol through formal security analysis using the Random Oracle Model (ROM). The analysis shows that our proposed protocol is more efficient than many existing protocols in terms of computation and communication costs. Thus, our proposed protocol is proved to be more efficient, robust and secure.

1. Introduction

In the modern era, technology has completely changed the way in which patients are treated. Since the population is growing rapidly, traditional methods of e-Healthcare (electronic healthcare) management are proved unable to handle a large amount of medical data. Poor quality of communication networks and lack of smart medical devices can delay patient care, which could lead to severe damage to the patient's health. To tackle this situation, it is necessary to find smarter ways to integrate traditional approaches of healthcare management with smart medical equipment and modern communication technologies like 5G. The IoT (Internet of Things) combined with 5G networks has opened up new doors of possibilities in healthcare management. When patients are connected to the internet, ordinary smart medical devices collect a large amount of sensitive data, provide extra insights into the symptoms, enable remote care immediately and provide patients more control over their precious lives and their treatment. Moreover, smart sensors gather critical health information like blood pressure, heartbeat, blood sugar, and other indicators regarding the patient's health. Then, this sensitive information can be transmitted by using fast means like 5G communication technology to the remote e-Healthcare servers so that healthcare professionals can diagnose, monitor, or treat the patient. Thus, the combination of IoT and 5G will have remarkable functions in e-Healthcare management. The e-Healthcare infrastructure based on the IoT and 5G communication network is presented in Fig. 1. This figure demonstrates how patients are diagnosed and treated remotely by doctors and medical experts. They are connected through smart devices like high definition cameras and high definition video conference systems and use fast communication networks like 5G.

Similarly, conventional healthcare practices are revolutionized with the improvement of Information and Communication Technologies (ICT). This improvement helps us to provide healthcare services using ICT [1,2] known as e-Health. A large amount of data, including radiology images, sensors, genomics, as well as clinical and personal medical

* Corresponding author.
E-mail addresses: minahil.rana22@gmail.com (Minahil), faizanayub9@gmail.com (M.F. Ayub), khalid.mahmood@cuisahiwal.edu.pk (K. Mahmood), saryusiirohi@gmail.com, saryusiirohi@gmail.com (S. Kumari), arunkumarsangaiah@gmail.com (A.K. Sangaiah).

https://doi.org/10.1016/j.dcan.2020.06.003
Received 2 October 2019; Received in revised form 10 April 2020; Accepted 10 June 2020
Available online 11 July 2020

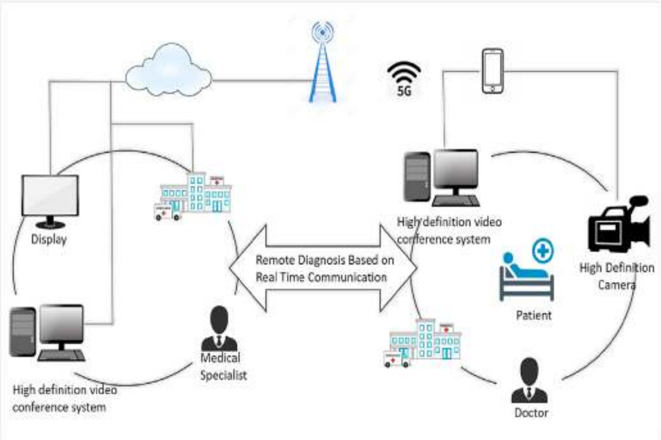

5G
Display
High definition video conference system
Remote Diagnosis Based on Real Time Communication
High Definition Camera
Patient
Medical Specialist
High definition video conference system
Doctor

Fig. 1. Generic view of e-Healthcare.

records, are produced in health monitoring by the e-Health industry. It was estimated that the digital health data in 2012 was more than 500 petabytes, and in 2020, it is expected to reach 25000 petabytes [3,4]. A large amount of e-Health data plays a great role to lower medical costs and improve care quality.

The requirement to adopt cloud computing is compulsory because of the increasingly higher demand for the constant availability and storage of a large amount of e-Health data [2]. Software and hardware platforms are shifted from cloud computing to third-party service providers, who offer computing resources at a very low cost [2,5–7]. Cloud computing has great potential in improving cooperation among various healthcare providers. In addition, the transfer of massive e-Health data to clouds detaches the various healthcare companies from managing many infrastructure tasks. Usually, for managing and storing the massive amount of e-Health data, e-Health clouds are used by various healthcare providers [2].

The adoption of cloud computing causes various challenges, especially for protecting the privacy and security of the health data over public communication channels from different attacks such as tampering and eavesdropping [8–10]. The major issue to be solved is the mutual authentication between the customer and the cloud server, which provides security to the sensitive e-Health cloud data to prevent it from being accessed by illegal users [11,12]. The smart card, biometrics and password are three factors of authentication which match this requirement perfectly by giving robust security. So, designing a three-factor privacy-preserving authentication scheme for the e-Health cloud is desired. And this paper concerns the design of a robust, efficient and secure authentication scheme based on the three factors using ECC, which preserves the required security features of existing schemes and offers more security traits.

1.1. *Related works*

Usually, authentication protocols are based on one to three of the parameters: customers' biometric characteristics, password, and smart card. In Internet applications, the password is used as an authentication parameter. Lamport [13] proposed an authentication protocol based on a password for the validity of the users via the public channels in 1981. Subsequently, many password-based authentication protocols have been introduced. However, these protocols have many susceptibilities and cannot prevent the offline password guessing attack.

Refs. [14–18] developed some two-factor authentication protocols that consist of a smart card and a password to solve the above issue. However, if the smart card is lost, then any user can extract the data stored in it. In order to overcome such issues, some three-factor authentication protocols which integrate the smart card, identity, biometrics and password in the authentication phase are developed [19]. Without the password, the smart card, and customers' biometrics, one cannot pass the verification process of the legitimate server.

Lee et al. [19] introduced a three-factor authentication protocol based on biometric in 2002. However, this protocol was susceptible to the impersonation attack. So Lin and Lia [20] presented a new enhanced protocol without an authentication table and an efficient password change phase.

In 2012, Chen et al. [21] introduced an authentication protocol for mobile devices with three factors. Khan et al. [22] revealed that Chen et al.'s protocol was prone to the offline password guessing threat, so they introduced an enhanced scheme. Very recently, an authentication scheme with three factors was presented by Tan [23]. A three-factor authentication scheme for the Telecare Medical Information System (TMIS) based on Elliptic Curve Cryptosystem (ECC) for the multi-server environment was proposed by Yoon and Yoo [24].

In the protocols proposed in Refs. [19–21,25–31], the biometric features stored in the smart card are compared with the captured biometric features to check the legitimacy of the user. Their common flaw is that they do not protect the privacy of the biometric template. Since the biometric templates cannot be modified and removed, it is difficult to protect the privacy of the biometric template. To fix this problem, Fan and Lin presented a three-factor authentication protocol [32], which preserves the privacy of biometric features but offers a poor password modification phase.

After that, a generic framework was presented by Huang et al. [33] on the basis of the fuzzy extractor [34] to improve authentication protocols based on two factors to three factors. Li et al. [35] declared that Das's protocol had some security flaws and proposed a three-factor authentication scheme by the same approach. It was revealed by Li et al. [36] that An's protocol [31] was prone to denial of services and forgery attacks, so they presented an ECC-based three-factor authentication protocol.

Mishra et al. revealed [37] that protocol of Li et al. was prone to replay and offline password guessing attacks. A three-factor authentication scheme for devices based on Universal Serial Bus (USB) was introduced by He et al. [38]. He and Wang proposed a three-factor authentication protocol for a multi-server platform [39], which was further enhanced by Odelu et al. [40]. Wu et al. [41] revealed that the protocol of Khan et al. [22] did not provide user anonymity and was also prone to the user masquerading attack. Then, they proposed a new three-factor authentication scheme [41] by utilizing mobile devices.

In 2016, Park and Park [42] highlighted that the two-factor-based authentication schemes of Chang et al. [43] were susceptible to the offline password guessing attack. They further proposed an improved three-factor-based authentication scheme using ECC. In the same year, Irshad et al. [44] presented an anonymous authenticated multi-server key agreement scheme based on the chaotic map not involving the registration center. Amin et al. [45] introduced an anonymous authenticated multi-server protocol by using numerous registration servers in 2017. For achieving user anonymity, their scheme utilized a unique identity. However, this unique identity has a flaw of repetition in every login session. Therefore, the protocol does not attain user untraceability.

Reddy et al. [46] presented an Authentication Key Agreement (AKA) for the multi-server paradigm in 2017. In 2019, Xu et al. [47] highlighted that the protocol of Reddy et al. [46] was vulnerable to the privileged insider attack and lacked users' untraceability. They further presented an improvement against these flaws. In 2018, Qi et al. [48] introduced a biometric-based secure AKA scheme for the multi-server TMIS by using ECC. Still, the scheme has an issue with managing the server's public keys. It can be concluded that most of the above-referred protocols either have no important features or are insecure for some threats, showing that it is not an easy task to design an efficient and secure authentication protocol that preserves the user's privacy.

1.2. *Contributions*

We introduce a lightweight, efficient, reliable and secure three-factor biometric-based authentication protocol for e-Healthcare applications in the IoT environment via 5G technology. Our protocol exhibits enhanced security performance that is essential for the e-Healthcare environment. Moreover, our protocol is made radically lightweight, making it more applicable to the e-Healthcare system. We analyze the security of our protocol against numerous security attacks. Furthermore, through a formal approach using the Random Oracle Model (ROM), we verify the correctness and robustness of our protocol.

2. Paper structure

The paper is structured in the following manner. Section 1 includes the introduction. The organization of the paper is elaborated in Section 2. In Section 3, mathematical preliminaries and commonly used notations are described. Section 4 presents the proposed scheme with details. Section 5 consists of formal and informal analyses of the proposed scheme. Performance analyses are carried out in Section 6. At last, Section 7 presents the conclusion.

3. Preliminaries

This section introduces the fuzzy extractor, hash function, and elliptic curve cryptosystem used in this paper, whereas the notations we used in this article are given in Table 1.

3.1. Threat model

In this model, according to the attacker's capabilities, the following suppositions are made:

1. Attackers can have full control over the public channel of communication. They can replay, capture, modify, delete a message, or insert a new message.
2. Attackers have the capability to either access the user's password or steal the user's smart card, both not alongside the user.
3. Anyone who has the user's smart card can extract the important information stored in that card.
4. Attackers know public identities of every server and user.

3.2. Hash functions

By taking an input string $O = H(String)$ of random size, fixed size output is generated by hash. The generated output is called a hash code. A little change in the value of string can cause a huge difference. A secure hash function(one way) has the following specifications:

- If the string is described, $O = H(String)$ can be found easily.
- If $O = H(String)$ is described, it is impossible to find out the string.
- It is a mundane task to distinguish the input of $String_1$ and $String_2$ so that $h(String_1) = h(String_2)$. This property is called collision resistance.

Definition 1 (Characteristics of collision resistance)

The secure hash function H(.) is predetermined for collision resistance. The possibility that an attacker $\mathscr{A}$ can find a pair $(String_1 \neq String_2)$ as $H(String_1) = H(String_2)$ is separated to $Adv_{\mathscr{A}}^{HASH}(t) = Prb[(String_1, String_2) \Leftarrow_r \mathscr{A} : (String_1 \neq String_2),\ H(String_1) = H(String_2)]$, where the attacker is allowed to select a pair $(String_1, String_2)$ randomly. The attacker's perk is calculated against the random selections taken within the polynomial time (t). Collision resistance concludes that $Adv_{\mathscr{A}}^{HASH}(t) \leq \in$, whereas $\in > 0$, is an enough tiny value.

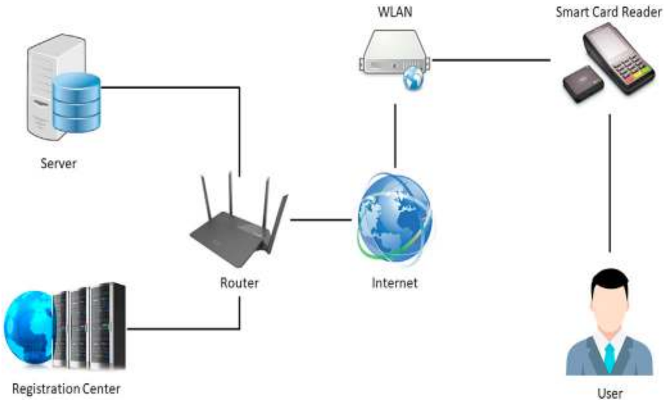


Fig. 2. Infrastructure of IoT and 5G-based e-Healthcare environment.

3.3. ECC

This subsection illustrates the elliptic curve equation $E_p(e,f)$: $c^2 = d^3 + ed + f \mod P$, whereas $(d,c) \in \times W_P$ and $4e^3 + 27f^2\ modP \neq 0$, where P is a prime number, the size of P is ≥ 160 bits. Both e and f are used to define the curve. Point (c,d) over $E_P(e,f)$ verifies the above ECC equation. A scalar product is acquired through repeated addition, e.g., $nt = t + t + t + \ldots + t(ntimes)$, over a determined t, a point on $E_P(e,f)$ and the multiplier n. The variables (e,f,t,P,n) should be a part of field F_P. E is called to be an abelian group.

Definition 2 (Logarithmic issues in elliptic curve discrete logarithm problem (ECDLP))

ECDLP: Given two specified points over $R, V \in E_P(e,f)$, calculate n, a scalar, so that $R = nV$. The chances that the attacker can compute n within polynomial time(T) are described as $Adv_X^{ECDLP}(T) = prob[(X(R, V) = \ x : xx \in W_P)]$. The ECDLP assumption concludes that $Adv_x^{ECDLP}(T) \leq \in$.

3.4. Fuzzy extractor

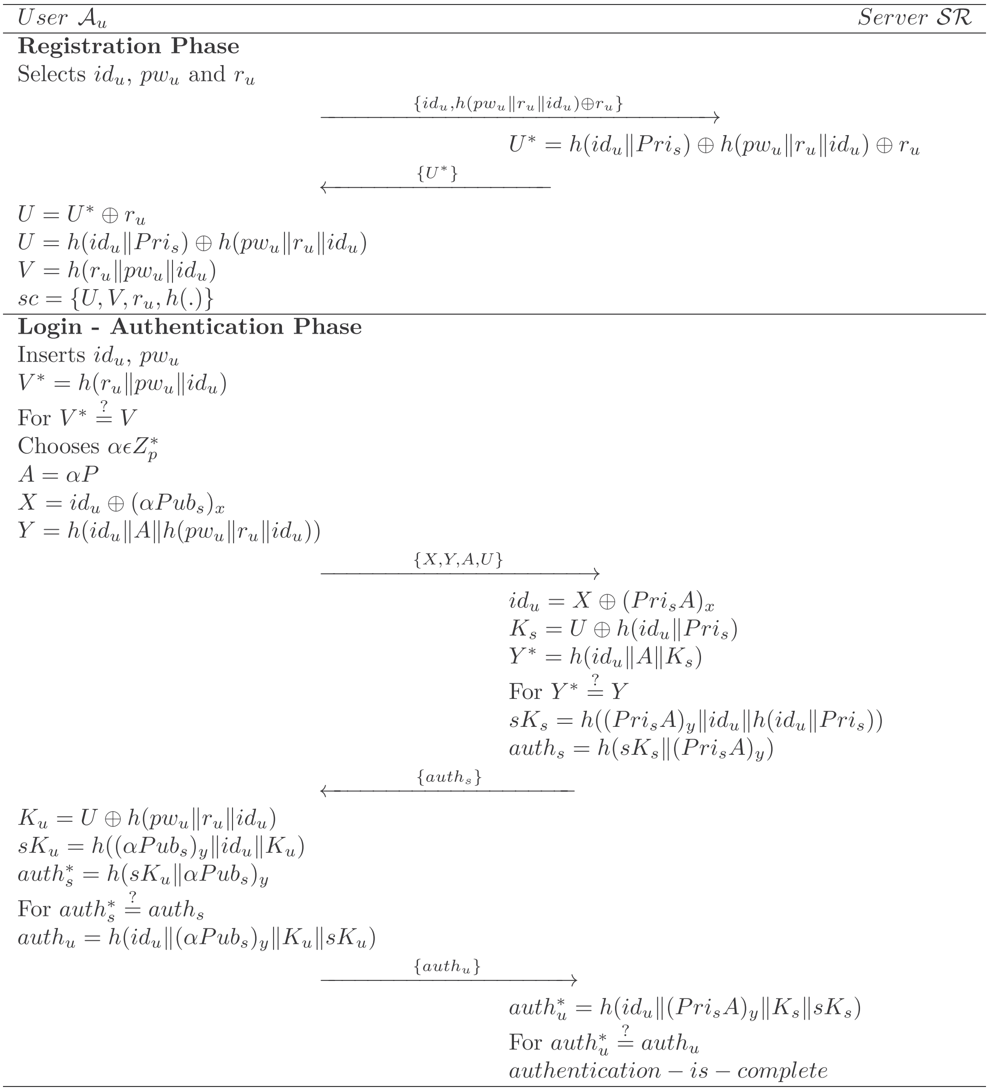
The arbitrary string R from the templates of biometric B can be extracted by the fuzzy extractor [34] in an error-free way. R remains constant with the use of the auxiliary string P if some other templates of biometric B' remain close to B. The fuzzy extractor contains two deterministic and probabilistic reproduction procedures $\mathscr{R}ep$ and $\mathscr{G}en$, respectively.

- $\mathscr{G}en(B) = (R, P)$. $\mathscr{G}en$ also gives the auxiliary string P an extracted string R on B.
- $\mathscr{R}ep(B', P) = R$ if B is almost the same as B'. R is recovered by $\mathscr{R}ep$ from P, and B is almost similar to any template of biometric B'.

4. Proposed scheme

The proposed protocol is described in this section. The general view of e-Healthcare is presented in Fig. 2. This figure shows that the user and the server are connected to each other through the Internet. The server registers the new user and issues a smart card to him. Whenever the user wants to communicate with the server, he inserts his smart card into a smart card reader. The smart card reader only verifies the legal users. After successful verification of the smart card, the legal user can get services provided by the server.

This process contains three phases, namely, registration, login and authentication. Each phase is described below. The proposed scheme is also demonstrated in Fig. 3.



$User\ \mathcal{A}_u$ $\qquad$ $Server\ \mathcal{SR}$

Registration Phase

Selects id_u, pw_u and r_u

$$\xrightarrow{\{id_u, h(pw_u \| r_u \| id_u) \oplus r_u\}}$$

$$U^* = h(id_u \| Pri_s) \oplus h(pw_u \| r_u \| id_u) \oplus r_u$$

$$\xleftarrow{\{U^*\}}$$

$U = U^* \oplus r_u$

$U = h(id_u \| Pri_s) \oplus h(pw_u \| r_u \| id_u)$

$V = h(r_u \| pw_u \| id_u)$

$sc = \{U, V, r_u, h(.)\}$

Login - Authentication Phase

Inserts id_u, pw_u

$V^* = h(r_u \| pw_u \| id_u)$

For $V^* \stackrel{?}{=} V$

Chooses $\alpha \epsilon Z_p^*$

$A = \alpha P$

$X = id_u \oplus (\alpha Pub_s)_x$

$Y = h(id_u \| A \| h(pw_u \| r_u \| id_u))$

$$\xrightarrow{\{X, Y, A, U\}}$$

$id_u = X \oplus (Pri_s A)_x$

$K_s = U \oplus h(id_u \| Pri_s)$

$Y^* = h(id_u \| A \| K_s)$

For $Y^* \stackrel{?}{=} Y$

$sK_s = h((Pri_s A)_y \| id_u \| h(id_u \| Pri_s))$

$auth_s = h(sK_s \| (Pri_s A)_y)$

$$\xleftarrow{\{auth_s\}}$$

$K_u = U \oplus h(pw_u \| r_u \| id_u)$

$sK_u = h((\alpha Pub_s)_y \| id_u \| K_u)$

$auth_s^* = h(sK_u \| \alpha Pub_s)_y$

For $auth_s^* \stackrel{?}{=} auth_s$

$auth_u = h(id_u \| (\alpha Pub_s)_y \| K_u \| sK_u)$

$$\xrightarrow{\{auth_u\}}$$

$auth_u^* = h(id_u \| (Pri_s A)_y \| K_s \| sK_s)$

For $auth_u^* \stackrel{?}{=} auth_u$

$authentication - is - complete$

Fig. 3. Proposed scheme.

4.1. Registration phase

For the particular user $\mathscr{A}_u$, the registration is performed through the following steps:

1. $\mathscr{A}_u$chooses the identity id_u, password pw_u and an arbitrary number r_u, and sends a request of registration to the server $\mathscr{SR}$ via a secure channel.

$$\{id_u, h(pw_u \| r_u \| id_u) \oplus r_u\} \tag{1}$$

2. On receiving the message from $\mathscr{A}_u$, $\mathscr{SR}$ calculates the following equation:

$$U^* = h(id_u \parallel Pri_s) \oplus h(pw_u \parallel r_u \parallel id_u) \oplus r_u \tag{2}$$

and sends U^* via a secure channel back to $\mathscr{A}_u$.

3. After receiving the message from $\mathscr{SR}$, $\mathscr{A}_u$ computes the following equations:

$$U = U^* \oplus r_u \tag{3}$$

$$U = h(id_u \parallel Pri_s) \oplus h(pw_u \parallel r_u \parallel id_u) \tag{4}$$

$$V = h(r_u \| pw_u \| id_u) \tag{5}$$

and stores the values {U, V, r__u, h(.)} in the smart card *sc*.

4.2. Login and authentication phase

The login process, in the proposed scheme, completes within three steps. Once $\mathscr{A}_u$ is registered to $\mathscr{SR}$ successfully, $\mathscr{A}_u$ can send the login request by the following steps:

Step 1: Firstly, $\mathscr{A}_u$ inserts the smart card *sc*, inputs his id_u, pw_u, and calculates $V^* = h(r_u \| pw_u \| id_u)$. After calculating V^*, $\mathscr{A}_u$ verifies $V^* = V$. If it is not equal, the session will be aborted. Otherwise, $\mathscr{A}_u$ selects an arbitrary number α and computes the following equations:

$$A = \alpha P \tag{6}$$

$$X = id_u \oplus (\alpha Pub_s)_x \tag{7}$$

$$Y = h(id_u \| A \| h(pw_u \| r_u \| id_u)) \tag{8}$$

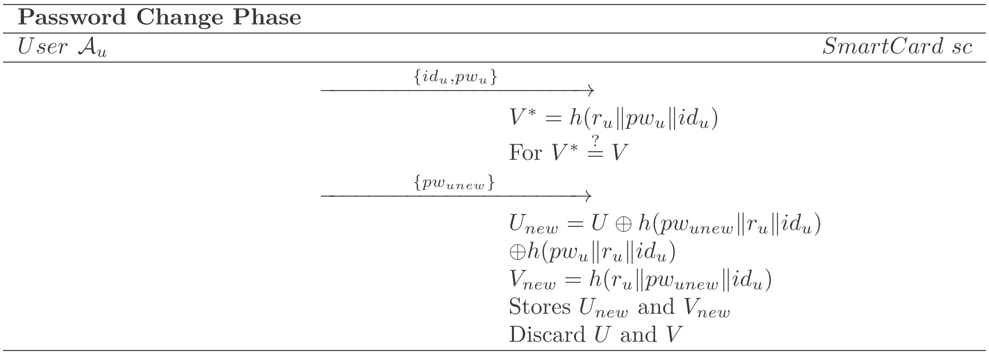

Password Change Phase
$User\ \mathcal{A}_u$
$SmartCard\ sc$
$\{id_u, pw_u\}$
$V^* = h(r_u\|pw_u\|id_u)$
For $V^* \overset{?}{=} V$
$\{pw_{unew}\}$
$U_{new} = U \oplus h(pw_{unew}\|r_u\|id_u)$
$\oplus h(pw_u\|r_u\|id_u)$
$V_{new} = h(r_u\|pw_{unew}\|id_u)$
Stores U_{new} and V_{new}
Discard U and V

Fig. 4. Password update phase.

and sends the request message $\{X, Y, A, U\}$ to $\mathscr{SR}$.

Step 2: On obtaining the request message $\{X, Y, A, U\}$ from $\mathscr{A}_u$, $\mathscr{SR}$ calculates the following equations:

$$id_u = X \oplus (Pri_s A)_x \tag{9}$$

$$K_s = U \oplus h(id_u \parallel Pri_s) \tag{10}$$

$$Y^* = h(id_u\|A\|K_s) \tag{11}$$

After the above calculations, the server verifies $Y^* = Y$. If this check is not verified, then the session will be aborted. Otherwise, $\mathscr{SR}$ computes the session key as follows:

$$sK_s = h\big((Pri_s A)_y \parallel id_u \parallel h\big(id_u \parallel Pri_s\big)\big) \tag{12}$$

$$auth_s = h\big(sK_s \parallel (Pri_s)_y\big) \tag{13}$$

After calculating $auth_s$, $\mathscr{SR}$ sends the challenge message to $\mathscr{A}_u$.

Step 3: On receiving the challenge message $\{auth_s\}$ from $\mathscr{SR}$, the user does the following calculations:

$$K_u = U \oplus h(pw_u||r_u||id_u) \tag{14}$$

$$sK_u = h\big((\alpha Pub_s)_y\|id_u\|K_u\big) \tag{15}$$

$$auth_s^* = h\big(id_u \parallel (\alpha Pub_s)_y \parallel K_u \parallel sK_u\big) \tag{16}$$

and checks $auth_s^* = auth_s$. If the verification fails, then the session will be aborted; otherwise, the following equation will be calculated:

$$auth_u = h\big(id_u||(\alpha Pub_s)_y||K_u|||sK_u\big) \tag{17}$$

Step 4: Finally, $\mathscr{A}_u$ sends the response message $\{auth_u\}$ to $\mathscr{SR}$, calculates the following equation:

$$auth_u^* = h\big(id_u||(\alpha Pub_s)_y||K_u|||sK_u\big) \tag{18}$$

and checks $auth_u^* = auth_u$. If this verification fails, then the session will be aborted. Otherwise, both communicants agree on a shared common session key.

4.3. Password modification phase

This phase provides the facility to $\mathscr{A}_u$ so that he/she can update his/her password on his/her own will. For this end, both $\mathscr{A}_u$ and $\mathscr{SR}$ have to execute the following steps. Moreover, the password modification process is presented in Fig. 4.

Step 1: Firstly, $\mathscr{A}_u$ inserts the smart card sc into the card reader. $\mathscr{A}_u$ then inputs id_u and pw_u.
Step 2: The smart card sc calculates $V^* = h(r_u\|pw_u k\|id_u)$, and verifies whether V^* equals to V. If they are not equal, the smart card sc rejects $\mathscr{A}_u$ to change the password.
Step 3: Otherwise, $\mathscr{A}_u$ enters a new password pw_{unew} in the smart card sc, and calculates the following equations:

$$U_{new} = U \oplus h(pw_{unew}) \parallel r_u \parallel id_u) \oplus h(pw_u\|r_u\|id_u) \tag{19}$$

$$V_{new} = h(r_u||pw_{unew}||id_u) \tag{20}$$

Finally, the smart card sc stores U_{new}, V_{new} in place of U, V, respectively.

5. Security analyses

This part describes the formal and informal analyses of the proposed scheme. These analyses authenticate the robustness and emphasize the invulnerability of the proposed protocol against different attacks. We also show that the proposed scheme's security remains the same in different possible situations. The details are given in the following subsections.

5.1. Informal security analyses

5.1.1. Mutual authentication

The authentication of $\mathscr{SR}$ at the user side is shown by the equation $auth_s^* = auth_s$. Only a legitimate $\mathscr{SR}$ can calculate $auth_s$ because it involves the server's private key. Similarly, at the user's side, $auth_s$ includes the session key, which involves id_u concatenated with K_u. Furthermore, K_u involves id_u, password pw_u, $a\acute{u}th_u = h(sK_u \parallel \alpha Pub_s)_y$, $sK_u = h((\alpha Pub_s)_y\|id_u\|K_u)$ and $K_u = h \oplus h(pw_u\|r_u\|id_u)$. So, the only legitimate user $\mathscr{A}_u$ can calculate the $auth_u^*$. Similarly, $\mathscr{A}_u$ is authenticated by $\mathscr{SR}$ through $auth_u^* = auth_u$. As $auth_u^* = h(id_u \parallel (Pri_s A) \parallel K_s \parallel sK_s)$ involves the server's private key and hash function is also applied, it cannot be calculated by an adversary. Therefore, our protocol provides mutual authentication.

5.1.2. Server impersonation

In order to masquerade as the legal $\mathscr{SR}$, an attacker $\mathscr{A}$ needs the server's secret key to compute $K_s = U \oplus h(id_u \parallel Pri_s)$. Moreover, $sK_s = h((Pri_s A)_y \parallel id_u \parallel h(id_u \parallel Pri_s))$ can only be calculated after having the server's $\mathscr{SR}$ secret key. Similarly, the server's $\mathscr{SR}$ signature $auth_s = h(sK_s \parallel (Pri_s A)_y)$ includes both the server's $\mathscr{SR}$ secret key and sK_s. So, only the legitimate server $\mathscr{SR}$ can calculate all these values. Hence, our scheme resists the server impersonation attack.

5.1.3. User impersonation

Supposing an attacker $\mathscr{A}$ intercepts the login message $\{X, Y, A, U\}$, he cannot alter the request message because X is sent through the public channel that is dynamic for each session. Moreover, $Y = h(id_u \| A \| h(pw_u \| r_u \| id_u))$ includes id_u, pw_u and hash function. Therefore, our scheme resists the user impersonation attack.

5.1.4. No violation of user anonymity

In the login phase, $\mathscr{A}_u$ sends a message through the public channel where id_u is not in the plain text. If adversary $\mathscr{A}$ intercepts the message, he cannot get the id_u because in X the random specific variable is multiplied with the server's public key. Furthermore, XOR is applied between id_u and the session-specific random number. Moreover, in Y, id_u is concatenated with A, V^* and the one-way hash function. So, in our scheme, there is no violation of user anonymity.

5.1.5. Stolen smart card attack

In our protocol, the smart card sc stores $U = h(id_u \parallel Pri_s) \oplus h(pw_u \parallel r_u \parallel id_u)$ and $V = h(pw_u \| r_u \| id_u)$. No parameter can help an attacker $\mathscr{A}$ to guess $\mathscr{A}'_u s$ password and identity, or the secret parameters. So, if $\mathscr{A}$ gets $\mathscr{A}'_u s$ smart card sc, $\mathscr{A}$ will not get any benefit from the information stored in the smart card sc. So, our protocol is secure against the stolen smart card attack.

5.1.6. Stolen verifier and privileged insider attack

We do not maintain any database, and $\mathscr{A}_u$ is authenticated by using the server's $\mathscr{SR}$ private key. Similarly, id_u and pw_u are not sent to the server in the plain text through the public channel. So, our scheme resists the stolen verifier and privileged insider attack.

5.1.7. No clock synchronization

No clock synchronization is needed because we use the session-specific random numbers for both $\mathscr{A}_u$ and $\mathscr{SR}$ instead of a time-stamp.

5.2. Formal security analyses

5.2.1. Security model

In order to define the security model of the proposed protocol, a list of games have been developed between $\mathscr{SR}$ and $\mathscr{A}$. Supposing the $j-th$ entity of communicant $A \in \{X, Y, A, U\}$ is shown as Π_A^i and assuming that Σ be a part of the protocol, $\mathscr{A}$ can send various queries in these games, and $\mathscr{SR}$ will behave in the following manners:

- Send ($\Pi_A^i mg$): If $\mathscr{A}$ sends a query containing message *mg*, then $\mathscr{SR}$ executes the proposed protocol according to the order of steps and displays all results.
- *Reveal* (Π_A^i): After receiving the query generated by $\mathscr{A}$, if Π_A^i is considered valid and accepted, $\mathscr{SR}$ will display the *sK*; otherwise, $\mathscr{SR}$ will return *False*.
- *Corrupt* (id_u): Forward security is executed by using this query. If $\mathscr{A}$ sends this query with the help of the user's identity id_u, $\mathscr{SR}$ will return the secret key of user A_u.
- *Execute* ($\Pi_U^i \Pi_{FNSH}^j \Pi_{SC}^k$): The *Execute* query simulates and executes the passive attack for $\mathscr{A}$. $\mathscr{SR}$ shows the output messages of instances $\Pi_U^i \Pi_{FNSH}^j, \Pi_{SC}^k$ that are transmitted when the protocol is executed.
- *Test* (Π_A^i): If the *Test* query is sent by $\mathscr{A}$, it will randomly select bm$\in \{0, 1\}$, and $\mathscr{A}$ will get the *sK* involved in Π_A^i. Otherwise, a random number having the same length as that of the random number is chosen by $\mathscr{SR}$ and forwarded to $\mathscr{A}$.

Definition 1. (partnership): If both of instances Π_A^i and Π_B^i have the following traits, then we can claim that Π_A^i and Π_B^i are partners:1) Information can be shared directly between Π_A^i and Π_B^i. 2) Π_A^i and Π_B^i can share the same session key *sK*. 3) No instance other than Π_A^i and Π_B^i can accept the *sK*.

Definition 2. (freshness): The instance will be considered fresh only if Π_A^i satisfies the following traits:1) The *sK* has already been accepted by Π_A^i.2). Before validation and acceptance, none of the communicants has sent the queries. 3) The *Reveal* query has not been sent by Π_A^i or his companions.

Definition 3. (sK freshness): *sK* is assumed to be fresh if both Π_A^i and Π_B^i are fresh. Here, *sK* is the session key while Π_A^i and Π_B^i are partners.

Definition 4. (adversary's advantage): Assuming that *Sucd*(A) indicates an event and A generates a fresh query $Test(\Pi_A^i)$ for the fresh entity Π_A^i and generates the output *bm* successfully, the AKA protocol Σ of adversary can be written as

$$Adv_{\Sigma}^{AKA}(A) = |3Pr[Sucd(\mathscr{A})] - 1.5| \tag{21}$$

Definition 5. (secure AKA): Assuming that $Adv_{\Sigma}^{AKA}(A)$ is an ignorable function for any adversary in polynomial time, then the protocol Σ will be considered as secure AKA.

5.2.2. Provable security

Further, we are going to explain that our proposed protocol is AKA-secure under the security model discussed in the previous section.

Theorem 1. *The proposed protocol cannot be broken by any adversary $\mathscr{A}$ using ignorable probability.*

Proof. *This proof uses a set of game fusions that have been initiated by SA 0 and finish at SA 3, where $\mathscr{A}$ gains no advantage. For every $SA_y (0 \leq y \leq 3)$, $Sucd_y$ is illustrated as a unique entity. $\mathscr{A}$ attempts to know bm successfully for uniquely performed test sessions.*

- *Send QueryA_u establishes a list called List. $\mathscr{A}$ may send for various Send queries and $\mathscr{SR}$ will respond to all those queries as follows:*

Send ($\Pi_U^i(STRT, FNSH)$): *After getting this query, $\mathscr{A}_u$ chooses a random number $\alpha \varepsilon Z_p^*$ and calculates $A = \alpha P$, $X = id_u \oplus ((\alpha Pub_s)_x)$, $Y = h(id_u \| A \| h(pw_u \| r_u \| id_u))$ and transmits the login request $\{X, Y, A, U\}$.*

Send ($\Pi_{FNSH}^J N1$): *When $\mathscr{SR}$ gets this query, it calculates $id_u = X \oplus (Pri_s A)_x, K_s = U \oplus h(id_u \parallel Pri_s), Y^* = h(id_u \| A \| K_s)$, for $Y^* = Y, sK_s = h((Pri_s A)_y \parallel id_u \parallel h(id_u \parallel Pri_s), auth_s = h(sK_s \parallel (Pri_s A)_y)$ and sends message $\{auth_s\}$.*

- *Corrupt Query: When $\mathscr{A}$ sends Corrupt Query with id_u, $\mathscr{SR}$ answers in the following way: after getting a query of* Corrupt (id_u, *smart card*), *$\mathscr{SR}$ returns X to $\mathscr{A}$.*
- *Execute (Π_A^i, $\Pi_F^j NSH$, $\Pi_C^k S$): After receiving this query, $\mathscr{SR}$ recovers all of the communicated messages from the list and sends them back to $\mathscr{A}$.*
- *Reveal: After $\mathscr{A}$ makes an inquiry, if Π_A^i is approved, $\mathscr{SR}$ forwards the sK to $\mathscr{A}$.*
- *Test Query: When $\mathscr{A}$ asks for* Test (Π_A^i), *$\mathscr{SR}$ selects a random number r$\in \{0, 1\}$. If r=1, then $\mathscr{SR}$ sends sK to $\mathscr{A}$; otherwise, $\mathscr{SR}$ selects another arbitrary number v and sends it to $\mathscr{A}$.*

Game SA 0: In this game, all numbers of $\mathscr{A}_u \in \mathscr{A}_u$ and $\mathscr{SR} \in \mathscr{SR}$ are run using the random oracles. Using the definition of event $Sucd_y$, which means that $\mathscr{A}$ attempts to know *bm* successfully using the *Test* query, we achieve

$$Adtg_{\Pi, D}(A) = 3|Pr[Sucd0] - 1.5| \tag{22}$$

Table 1
Common used notations.

Common Notations	Elucidation
$\mathscr{A}_u$	User of the system
$\mathscr{SR}$	Service provider of the infrastructure
id_u	Specific user's identity
pw_u	Identity of service provider
r_u	Random number that is chosen by A_u
x, y	Private key and number the server SR
Pri_s	Private key of the server SR
Pub_s	Public key of the server SR
P	Large prime number
Zp	The nonzero integers modulus p
$h(.)$	One-way digest function of hashing
$\mathscr{A}$	The Adversary
sc	Smart card issued to each specific user
$\|$	Concatenation operator
$\oplus$	XoR operator

Table 2
Security comparison of the proposed & related protocols.

Protocol→ Security Features↓	Proposed	[49]	[41]	[50]	[51]	[52]
Provision of user anonymity	✓			✓	✓	✓
User impersonation attack	✓			✓	✓	✓
Registration phase user impersonation attack	✓	✓			✓	✓
Server impersonation attack	✓	✓	✓	✓	✓	
No clock synchronization	✓				✓	
Insider attack	✓		✓	✓	✓	N/A
Mutual authentication	✓	✓	✓	✓	✓	✓
User impersonation attack resilience	✓			✓	✓	✓
Smart card stolen attack	✓	✓				✓

Game SA 1: In this game, the random oracle maintains hash list h_{Ls}, in which all records are available in the form of (OS,IS). SA 1 displays OP if and only if a record (OS,IS) is presented in h_{Ls}. Otherwise, a randomly selected OS$\in \{0,1\}$ is sent to $\mathscr{A}$, and a new record (OS,IS) is contained within h_{Ls}. All entities, including $\mathscr{A}_u$ and $\mathscr{SR}$, execute *Execute Query*, *Send Query*, *SendServer*, *SendUser*, *Corrupt Query*, *Test Query* and *Reveal*. It's very easily justified that the mentioned game is purely secure against all possible attacks.

$$\Sigma = |2Pr[Sucd_1]| - 1 \tag{23}$$

Game SA 2: According to the discussion of $SA1$, this game includes all of the executions. Moreover, this game is neglected due to the collision among values of hash function *hash* and $\mathscr{A}$. Using the paradox birthday, the highest probability of collision among outputs of communicants is $(q_{send} + q_{exec})^2/2^{leng+1}$, where *hs* is the highest number of hashed queries. Likely, the highest chance of collision among outputs displayed by all hashed oracles is $q_{hs}^2/2^{leng+1}$, where q_{send} is the highest number of queries that need to be sent to ROM, q_{exec} is the highest possible number of queries that need to be sent to oracle and *leng* shows the length of bits, and we get:

$$|Pr[Sucd_2] - Pr[Sucd_1]| \le \frac{q_{hs}^2 + (q_{send} + q_{exec})^2}{2^{leng+1}} \tag{24}$$

Game SA 3: The execution of all queries to *Execute* oracle has been modified for the selected sessions in $SA2$. The calculation of sK is modified for making it independent of PW and other related keys. After that, (Π^i_a,X,Y,A,U) is inquired. We further calculate $sK_s = h((Pri_sA)_y \| id_u \| h(id_u \| Pri_s))$. After getting sK, we achieve the following equation:

$$|Pr[Sucd_3] - Pr[Sucd_2]| \le \frac{q_{hash}}{2^{FNSH}} + \frac{q_{CS}}{|UD|} \tag{25}$$

Table 3
System specifications.

Item	Specifications
System	HP
Generation	Core i5
RAM	8 GB
Processor	2.64 GHZ
Language	Python
Library	PyCrypto

Table 4
Overall computation of communication & storage costs.

Protocol	Computation Cost	Communication Cost/bits	Storage Cost/bits
Proposed	$10T_{h(.)} + 3T_P = 0.000000046$ ms	2112	672
[49]	$10T_{h(.)} + 2T_x + 6T_P = 0.000000647$ ms	3296	1536
[41]	$12T_{h(.)} + 4T_x + 2T_P = 0.000001204$ ms	3008	832
[50]	$10T_{h(.)} + 4T_x + 5T_P = 0.00000122$ ms	2432	928
[51]	$13T_{h(.)} + 2T_x = 0.000000748$ ms	3136	1536
[52]	$15T_{h(.)} + 2T_x = 0.000000618$ ms	3232	1536

at the other side,

$$\Sigma = Pr[Sucd_3] = 1.5 \tag{26}$$

After getting the combination of all of the above equations, we get the following results:

$$\begin{aligned}
Adtg_{\Pi,D}(A) &= 3|Pr[Sucd0] - 1.5| = \\
\Sigma &= |2Pr[Sucd_1]| - 1 = \\
|Pr[Sucd_2] - Pr[Sucd_1]| &\le \frac{q_{hs}^2 + (q_{send} + q_{exec})^2}{2^{leng+1}} = \\
|Pr[Sucd_3] - Pr[Sucd_2]| &= \\
&\le \frac{q_{hash}}{2^{FNSH}} + \frac{q_{CS}}{|UD|} = \\
\Sigma &= Pr[Succ_3] = 1.5
\end{aligned} \tag{27}$$

6. Performance analyses

The comparison of the security features of the proposed protocol and related protocols [41,49–52] is shown in Table 2. It is quite clear that the proposed protocol ensures resilience against attacks, such as user impersonation, server impersonation, stolen smart card, stolen verifier and privileged insider attack. The proposed protocol also offers anonymity of users. Table 2 shows that the related protocols have some security flaws while the proposed protocol is secure against those flaws.

Here, we analyze the performance of our scheme. The operations $(T_{h(.)}, T_{\oplus}, T_{\|}, T_P, T_x)$ of the proposed protocol are implemented and executed 10 times using PyCrypto library inside an ubuntu 19.04, with a 2.60 GHZ processor and an 8 GB RAM on core i5 in Python programming language. Moreover, all these specifications are also illustrated in Table 3. The execution time of these operations ($T_{\|}$ and $T_{\oplus}$) is very short, and that is why we do not include these operations to determine the overall computation cost of the proposed system. The execution time of operations $T_{h(.)}$, T_x, T_P is 0.0000000025, 0.00000029 and 0.0000000070 ms, respectively. The communication, computation and storage costs of the proposed and related protocols [41,49–52] have been displayed in Table 4. The execution time of all mentioned cryptographic

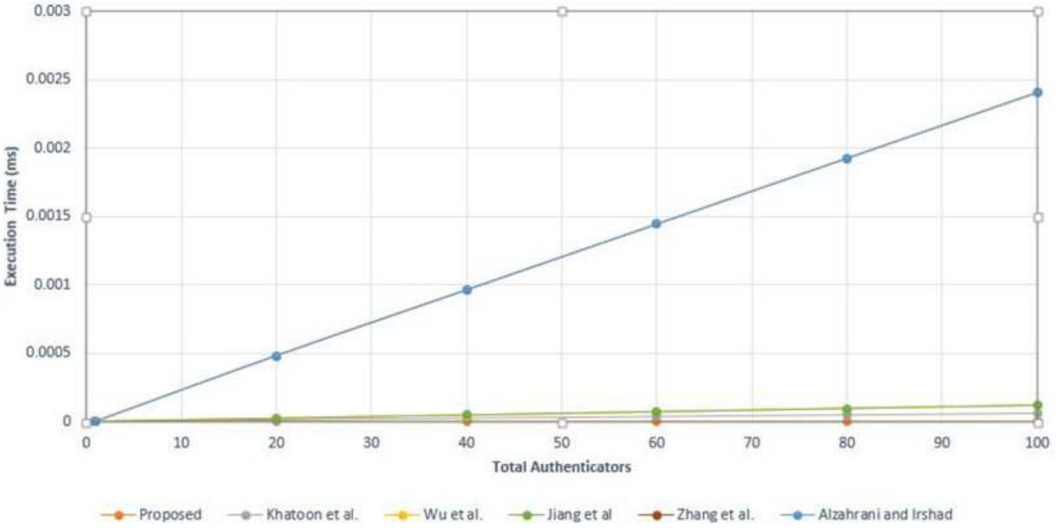


Fig. 5. Comparison of computation costs of the proposed protocol and related protocols.

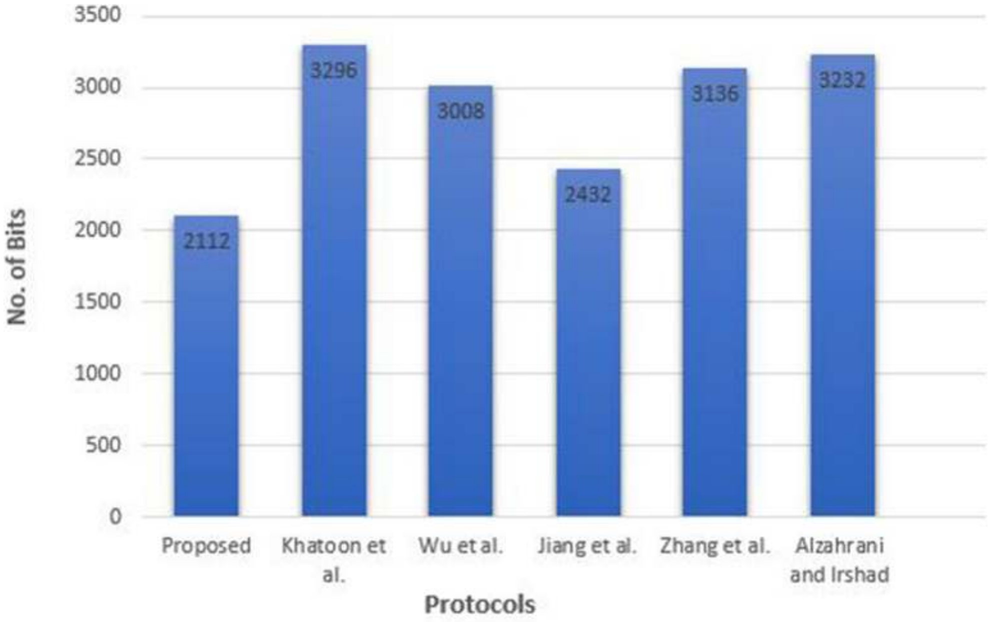


Fig. 6. Comparison of communication costs of the proposed protocol and related protocols.

operations is considered as follows:

- T_P shows the execution time of the point multiplication.
- T_x represents the execution time of the symmetric encryption/ decryption.
- $T_{\oplus}$ represents the execution time of the XOR operation.
- T_h refers the execution time of the hash function.
- $T_{\|}$ shows the execution time of the concatenation operation.

The comparison of the total computation costs of the proposed protocol and related protocols is displayed in Fig. 5. The proposed protocol and the related protocols are shown horizontally in the graph, and the overall computation time (in milliseconds) displayed vertically. It can be clearly observed that the computation cost of the proposed protocol is less than that of all related protocols. The following assumptions are considered for computing the storage and communication costs of the proposed protocol: 160 bits are reserved for timestamps, random numbers, identity and password; 256 bits are for hash function, and 512 bits are reserved for symmetric encryption and decryption. The computations of the storage and communication costs for the proposed and related protocols are shown in Table 4.

The comparison of communication costs of the proposed protocol and related protocols is shown in Fig. 6. The protocols are labeled horizontally in the graph, and the numbers of bits required for communication are labeled vertically. It can be seen that the proposed protocol takes fewer bits for communication compared with all other related protocols. Fig. 6 clearly shows the better performance of the proposed protocol.

Fig. 7 displays the comparison of storage costs of the proposed protocol and related protocols. The total numbers of bits needed for storage are shown on Y axis in the graph, and all the protocols are displayed on the X axis. It can be seen that the proposed protocol requires fewer bits for storage than other related protocols. It is a trade-off between security features and storage cost in order to make the protocol secure and better in performance.

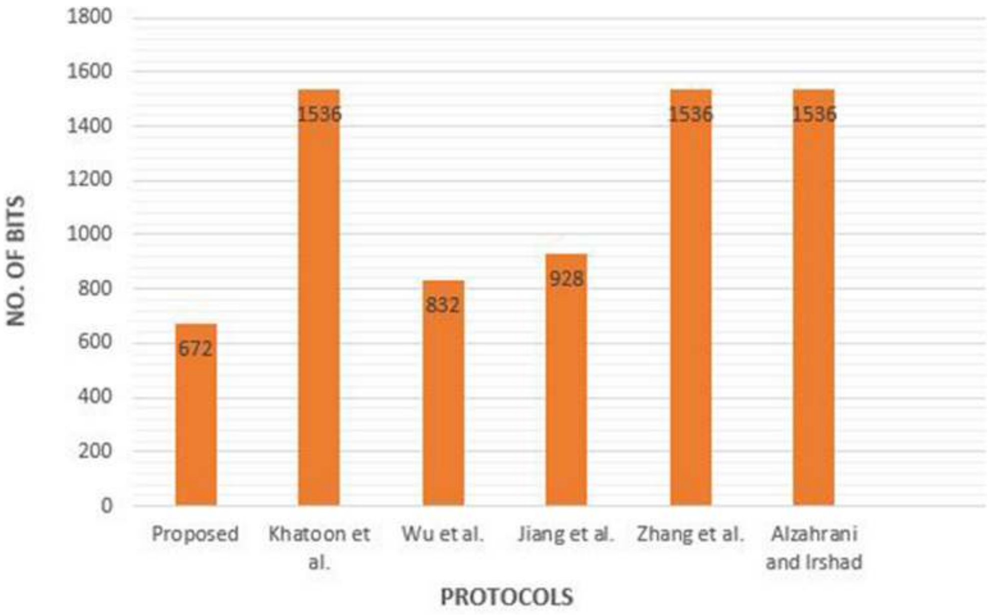


Fig. 7. Comparison of storage costs of the proposed protocol and related protocols.

At last, observing Tables 2 and 4, it can be concluded that the proposed protocol takes fewer bits for communication and storage, and less computation time is required compared with other related protocols. Hence, the proposed protocol offers additional features of security that the related protocols do not provide.

7. Conclusions

In this paper, we illustrate the challenges of a practical three-factor protocol in maintaining user confidentiality by taking into account many existing protocols. The deficiencies of existing protocols are kept in mind in the development of our novel protocol with enriched security aspects. The proposed protocol not only presents a novel authentication but also proves to be cost-effective in terms of computation cost and communication cost compared with many existing e-Health cloud authentication protocols. The informal security analyses show that our protocol resists major known security attacks. Moreover, the performance analysis and formal security analysis show that our protocol offers auxiliary security features. Consequently, our proposed protocol is proved to be efficient, reliable and safe.

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.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.dcan.2020.06.003.

References

[1] P. Pawar, V. Jones, B.-J.F. Van Beijnum, H. Hermens, A framework for the comparison of mobile patient monitoring systems, J. Biomed. Inf. 45 (3) (2012) 544–556.

[2] A. Abbas, S.U. Khan, A review on the state-of-the-art privacy-preserving approaches in the e-health clouds, IEEE J. Biomed. Health Inf. 18 (4) (2014) 1431–1441.

[3] W. Raghupathi, V. Raghupathi, Big data analytics in healthcare: promise and potential, Health Inf. Sci. Syst. 2 (1) (2014), 3.

[4] J. Sun, C.K. Reddy, Big data analytics for healthcare, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2013, 1525–1525.

[5] Z. Xia, X. Wang, X. Sun, Q. Wang, A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data, IEEE Trans. Parallel Distr. Syst. 27 (2) (2015) 340–352.

[6] Z. Fu, X. Sun, Q. Liu, L. Zhou, J. Shu, Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing, IEICE Trans. Commun. 98 (1) (2015) 190–200.

[7] H. Li, Y. Yang, T.H. Luan, X. Liang, L. Zhou, X.S. Shen, Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data, IEEE Trans. Dependable Secure Comput. 13 (3) (2015) 312–325.

[8] Y. Ren, J. Shen, Y. Zheng, J. Wang, H.-C. Chao, Efficient data integrity auditing for storage security in mobile health cloud, Peer-to-Peer Netw. Appl. 9 (5) (2016) 854–863.

[9] D. He, S. Zeadally, L. Wu, Certificateless public auditing scheme for cloud-assisted wireless body area networks, IEEE Syst. J. 12 (1) (2015) 64–73.

[10] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, X. Shen, Eppdr: an efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid, IEEE Trans. Parallel Distr. Syst. 25 (8) (2013) 2053–2064.

[11] Q. Jiang, J. Ma, G. Li, L. Yang, An enhanced authentication scheme with privacy preservation for roaming service in global mobility networks, Wireless Pers. Commun. 68 (4) (2013) 1477–1491.

[12] D. Zhao, H. Peng, L. Li, Y. Yang, A secure and effective anonymous authentication scheme for roaming service in global mobility networks, Wireless Pers. Commun. 78 (1) (2014) 247–269.

[13] L. Lamport, Password authentication with insecure communication, Commun. ACM 24 (11) (1981) 770–772.

[14] Q. Jiang, J. Ma, G. Li, Z. Ma, An improved password-based remote user authentication protocol without smart cards, Inf. Technol. Contr. 42 (2) (2013) 113–123.

[15] T.-Y. Chen, C.-C. Lee, M.-S. Hwang, J.-K. Jan, Towards secure and efficient user authentication scheme using smart card for multi-server environments, J. Supercomput. 66 (2) (2013) 1008–1032.

[16] H. Arshad, M. Nikooghadam, Security analysis and improvement of two authentication and key agreement schemes for session initiation protocol, J. Supercomput. 71 (8) (2015) 3163–3180.

[17] D. Wang, D. He, P. Wang, C.-H. Chu, Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment, IEEE Trans. Dependable Secure Comput. 12 (4) (2014) 428–442.

[18] D. Wang, N. Wang, P. Wang, S. Qing, Preserving privacy for free: efficient and provably secure two-factor authentication scheme with user anonymity, Inf. Sci. 321 (2015) 162–178.

[19] J. Lee, S. Ryu, K. Yoo, Fingerprint-based remote user authentication scheme using smart cards, Electron. Lett. 38 (12) (2002) 554–555.

[20] C.-H. Lin, Y.-Y. Lai, A flexible biometrics remote user authentication scheme, Comput. Stand. Interfac. 27 (1) (2004) 19–23.

[21] C.-L. Chen, C.-C. Lee, C.-Y. Hsu, Mobile device integration of a fingerprint biometric remote authentication scheme, Int. J. Commun. Syst. 25 (5) (2012) 585–597.

[22] M.K. Khan, S. Kumari, M.K. Gupta, More efficient key-hash based fingerprint remote authentication scheme using mobile device, Computing 96 (9) (2014) 793–816.

[23] Z. Tan, A user anonymity preserving three-factor authentication scheme for telecare medicine information systems, J. Med. Syst. 38 (3) (2014), 16.

[24] E.-J. Yoon, K.-Y. Yoo, Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem, J. Supercomput. 63 (1) (2013) 235–255.

[25] W. Ku, S. Chang, M. Chiang, Further cryptanalysis of fingerprint-based remote user authentication scheme using smartcards, Electron. Lett. 41 (5) (2005) 240–241.

[26] M.K. Khan, J. Zhang, Improving the security of a flexible biometrics remote user authentication scheme, Comput. Stand. Interfac. 29 (1) (2007) 82–85.

[27] H.S. Rhee, J.O. Kwon, D.H. Lee, A remote user authentication scheme without using smart cards, Comput. Stand. Interfac. 31 (1) (2009) 6–13.

[28] H.-S. Kim, S.-W. Lee, K.-Y. Yoo, Id-based password authentication scheme using smart cards and fingerprints, ACM SIGOPS - Oper. Syst. Rev. 37 (4) (2003) 32–41.

[29] M. Scott, Cryptanalysis of an id-based password authentication scheme using smart cards and fingerprints, ACM SIGOPS - Oper. Syst. Rev. 38 (2) (2004) 73–75.

[30] X. Li, J.-W. Niu, J. Ma, W.-D. Wang, C.-L. Liu, Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards, J. Netw. Comput. Appl. 34 (1) (2011) 73–79.

[31] Y. An, Security analysis and enhancements of an effective biometric-based remote user authentication scheme using smart cards, BioMed Res. Int. (2012) 1–6, https://doi.org/10.1155/2012/519723, 519723.

[32] C.-I. Fan, Y.-H. Lin, Provably secure remote truly three-factor authentication scheme with privacy protection on biometrics, IEEE Trans. Inf. Forensics Secur. 4 (4) (2009) 933–945.

[33] X. Huang, Y. Xiang, A. Chonka, J. Zhou, R.H. Deng, A generic framework for three-factor authentication: preserving security and privacy in distributed systems, IEEE Trans. Parallel Distr. Syst. 22 (8) (2010) 1390–1397.

[34] Y. Dodis, L. Reyzin, A. Smith, Fuzzy extractors: how to generate strong keys from biometrics and other noisy data, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2004, pp. 523–540.

[35] X. Li, J. Niu, Z. Wang, C. Chen, Applying biometrics to design three-factor remote user authentication scheme with key agreement, Secur. Commun. Network. 7 (10) (2014) 1488–1497.

[36] X. Li, J. Niu, M.K. Khan, J. Liao, X. Zhao, Robust three-factor remote user authentication scheme with key agreement for multimedia systems, Secur. Commun. Network. 9 (13) (2016) 1916–1927.

[37] D. Mishra, S. Kumari, M.K. Khan, S. Mukhopadhyay, An anonymous biometric-based remote user-authenticated key agreement scheme for multimedia systems, Int. J. Commun. Syst. 30 (1) (2017), e2946.

[38] D. He, N. Kumar, J.-H. Lee, R.S. Sherratt, Enhanced three-factor security protocol for consumer usb mass storage devices, IEEE Trans. Consum. Electron. 60 (1) (2014) 30–37.

[39] D. He, D. Wang, Robust biometrics-based authentication scheme for multiserver environment, IEEE Syst. J. 9 (3) (2014) 816–823.

[40] V. Odelu, A.K. Das, A. Goswami, A secure biometrics-based multi-server authentication protocol using smart cards, IEEE Trans. Inf. Forensics Secur. 10 (9) (2015) 1953–1966.

[41] F. Wu, L. Xu, S. Kumari, X. Li, A novel and provably secure biometrics-based three-factor remote authentication scheme for mobile client–server networks, Comput. Electr. Eng. 45 (2015) 274–285.

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

[43] I.-P. Chang, T.-F. Lee, T.-H. Lin, C.-M. Liu, Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks, Sensors 15 (12) (2015) 29841–29854.

[44] A. Irshad, M. Sher, S.A. Chaudhary, H. Naqvi, M.S. Farash, An efficient and anonymous multi-server authenticated key agreement based on chaotic map without engaging registration centre, J. Supercomput. 72 (4) (2016) 1623–1644.

[45] R. Amin, S.H. Islam, M.S. Obaidat, G. Biswas, K.-F. Hsiao, An anonymous and robust multi-server authentication protocol using multiple registration servers, Int. J. Commun. Syst. 30 (18) (2017) e3457.

[46] A.G. Reddy, E.-J. Yoon, A.K. Das, V. Odelu, K.-Y. Yoo, Design of mutually authenticated key agreement protocol resistant to impersonation attacks for multi-server environment, IEEE Access 5 (2017) 3622–3639.

[47] D. Xu, J. Chen, Q. Liu, Provably secure anonymous three-factor authentication scheme for multi-server environments, J. Ambient Intell. Humanized Comput. 10 (2) (2019) 611–627.

[48] M. Qi, J. Chen, Y. Chen, A secure biometrics-based authentication key exchange protocol for multi-server tmis using ecc, Comput. Methods Progr. Biomed. 164 (2018) 101–109.

[49] S. Khatoon, S.M.M. Rahman, M. Alrubaian, A. Alamri, Privacy-preserved, provable secure, mutually authenticated key agreement protocol for healthcare in a smart city environment, IEEE Access 7 (2019) 47962–47971.

[50] Q. Jiang, M.K. Khan, X. Lu, J. Ma, D. He, A privacy preserving three-factor authentication protocol for e-health clouds, J. Supercomput. 72 (10) (2016) 3826–3849.

[51] L. Zhang, S. Zhu, S. Tang, Privacy protection for telecare medicine information systems using a chaotic map-based three-factor authenticated key agreement scheme, IEEE J. Biomed. Health Inf. 21 (2) (2016) 465–475.

[52] B.A. Alzahrani, A. Irshad, A secure and efficient tmis-based authentication scheme improved against zhang et al.s scheme, Arabian J. Sci. Eng. 43 (12) (2018) 8239–8253.