MULTI-SERVER USER AUTHENTICATION SCHEME FOR PRIVACY PRESERVATION WITH FUZZY COMMITMENT

Stanley Mlato[1], Yesaya Gabriel[1], Prince Chirwa[1], and Hyunsung Kim[1,2]

[1]Dept. of Mathematical Sciences University of Malawi, Malawi
[2](Corresponding Author) Dept. of Cyber Security Kyungil University, Korea

ABSTRACT

The integration of artificial intelligence technology with a scalable Internet of Things (IoT) platform facilitates diverse smart communication services, allowing remote users to access services from anywhere at any time. The multi-server environment within IoT introduces a flexible security service model, enabling users to interact with any server through a single registration. To ensure secure and privacy preservation services for resources, an authentication scheme is essential. Zhao et al. recently introduced a user authentication scheme for the multi-server environment, utilizing passwords and smart cards, claiming resilience against well-known attacks. This paper conducts cryptanalysis on Zhao et al.'s scheme, focusing on denial of service and privacy attacks, revealing a lack of user-friendliness. Subsequently, we propose a new multi-server user authentication scheme for privacy preservation with fuzzy commitment over the IoT environment, addressing the shortcomings of Zhao et al.'s scheme. Formal security verification of the proposed scheme is conducted using the ProVerif simulation tool. Through both formal and informal security analyses, we demonstrate that the proposed scheme is resilient against various known attacks and those identified in Zhao et al.'s scheme.

KEYWORDS

Multi-server security, Privacy, Cryptanalysis, Fuzzy commitment, Authentication.

1. INTRODUCTION

With the substantial growth of available data resources in IoT environments, artificial intelligence (AI) technologies have garnered considerable attention in both research and industry [1-4]. The proliferation of data sources, users, and the volume of data required for establishing situational awareness and decision superiority for swift responses has increased significantly. The connectivity or integrations in a network expand exponentially in relation to the square of nodes within the network [5]. This remarkable growth and the resulting exponential complexity of technology have heightened the challenges in observing, orienting, deciding, and acting upon actionable data. Given that a multi-server environment can accommodate more connections and services, many companies are transitioning to such an environment [6-7]. Opting for a multi-server environment can offer substantial cost savings compared to continuously augmenting resources on a single server. This environment involves three distinct participants: remote users, service providers (S_j), and a registration center (*RC*), as illustrated in Fig. 1. The *RC* serves as the trusted third party, overseeing all S_j and users denoted by the red-colored lines in Fig. 1.

DOI: 10.5121/ijcnc.2024.16205

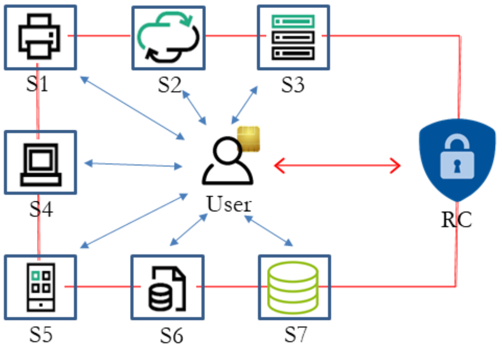


Figure 1. Multi-server network configuration

Given the sensitivity of security in any communication environment, substantial efforts have been dedicated to addressing security issues in the multi-server environment [8-10]. Among the foundational elements for ensuring secure communications, the user authentication scheme has received considerable attention. Tsaur introduced the initial user authentication scheme grounded in the integer factorization problem within the multi-server environment [11]. Nevertheless, a cryptanalysis by Kim et al. scrutinized Tsaur's scheme, specifically focusing on the offline password guessing attack under the assumption that the attacker could intercept the transmitted authentication messages [12]. Responding to this, Tsaur et al. conducted a cryptanalysis concentrated on an offline guessing attack and proposed a remedial scheme founded on the discrete logarithm problem [13]. Building upon the groundwork laid in [11-13], numerous user authentication schemes for the multi-server environment have been developed [14-21]. The first user authentication scheme utilizing passwords and smart cards in a multi-server environment, incorporating dynamic identity, was presented by Liao and Wang [16]. However, Hsiang and Shih identified security vulnerabilities in Liao and Wang's scheme, particularly concerning insider attacks, impersonation attacks, and forgery attacks [17]. Following this, Sood et al. highlighted security weaknesses in Hsiang and Shih's scheme, specifically against impersonation attacks, replay attacks, and stolen smart card attacks [18]. To address the issues in the preceding schemes, Yeh proposed an RSA-based authentication scheme, offering a formal security analysis based on the random oracle model [19]. Nonetheless, Truong et al. demonstrated that Yeh's scheme lacks mutual authentication and key agreement, proposing a corrective scheme for Yeh's approach [20]. Zhao et al. revealed that Truong et al.'s scheme is susceptible to impersonation attacks and offline password guessing attacks [21]. In response, Zhao et al. introduced an enhanced user authentication scheme, emphasizing its resilience against well-known attacks.

In this paper, we conduct a cryptanalysis of Zhao et al.'s user authentication scheme, identifying and discussing its vulnerabilities. Subsequently, we introduce a novel multi-server user authentication scheme with privacy preservation founded on fuzzy commitment, tailored to meet the security and privacy requirements of a multi-server environment. The primary contributions of our proposed scheme are outlined as follows: (1) achieving mutual authentication and session key agreement; (2) emphasizing privacy aspects, particularly anonymity and untraceability during the authentication phase; (3) enhancing user-friendliness by eliminating the involvement of other entities in the password update phase; (4) demonstrating resilience against not only well-known attacks but also potential threats specific to the multi-server environment.

The subsequent sections of this paper are structured as follows. Section 2 offers an overview of Zhao et al.'s authentication scheme. In Section 3, two security issues are identified within Zhao et al.'s scheme. Our proposed multi-server user authentication scheme is presented in Section 4, followed by a security analysis utilizing BAN logic and the ProVerif simulation tool in Section 5.

Section 6 delves into performance analysis. Lastly, Section 7 serves as the conclusion of the paper.

Table 1. Notations

Notation	Description
RC	Registration center
U_i	ith user
S_j	jth service provider
UID_i	Identity of U_i
SID_j	Identity of S_j
α	Master secret key of RC
$ASID_j$	Secret key of S_j
PW_i	Password of U_i
BIO_i	A set of minutiae points from the fingerprint of U_i
R_{Ki}, L_i, T_{Ki}, K_{CWi}	Random parameters for the fuzzy commitment of U_i
p, r_i, x_i, y_j	Random numbers
T_i, T_j	Timestamps
$K_{i,j}$, $K_{j,i}$	Session keys
$E_p(a, b)$	An elliptic curve
g	A group generator
$H_1(\cdot)$, $H_2(\cdot)$	Secure hash functions
$f(\cdot)$	Fuzzy commitment function
$\Psi_{enc}(\cdot)$, $\Psi_{dec}(\cdot)$	Encoding and decoding functions of error correction
$\cdot$	Scalar multiplication
$\oplus$	Exclusive or operation
$\parallel$	Concatenation operation
ΔT	Maximum allowed delay

2. Review of Zhao et al.'s Scheme

In this section, we briefly examine Zhao et al.'s user authentication scheme, utilizing passwords and smart cards within a multi-server environment [21]. The scheme proposed by Zhao et al. encompasses four distinct phases: initialization, registration, authentication, and password update. The notations employed in this paper are presented in Table 1.

2.1. Initialization Phase

To initialize the system, the registration center executes the following steps utilizing a security parameter K:

(1) Choose a prime number p with size K and then generate an elliptic curve $E_p(a, b)$ defined over Z_p^*, where a, $b \in Z_p$. Furthermore, produce a cyclic group G with prime order q from $E_p(a, b)$, and randomly pick a generator $g \in G$.

(2) Randomly sample an integer α from Z_p^*, and choose two cryptographically secure hash functions $H_1(\cdot) : \{0, 1\}^* \rightarrow \{0, 1\}^n$, $H_2(\cdot) : \{0, 1\}^* \rightarrow G$.

(3) Publish the system public parameters as $pp=\{E_p(a, b), g, H_1(\cdot), H_2(\cdot)\}$, which are available to all system users, and set $msk=\alpha$ as the master secret key.

2.2. Registration Phase

During this phase, each user U_i in the system registers with the RC to obtain a smart card. Additionally, each service provider S_j must register with the RC to establish a secure channel based on the acquired secret key, serving as a credential for S_j to demonstrate its legitimacy to system users. The registration process with the RC involves the following steps by S_j and RC:

(1) S_j selects a unique identity SID_j and sends it to RC via a secure channel.
(2) After receiving the registration request from U_i, RC computes $s_{i,j}=H_1(ASID_j \| UID_i) \oplus RPW_i$ for each, $j \in [m]$ and issues a smart card containing $\{\{s_{i,j}\}_{j=1,\ldots,m}, H_1(\cdot), H_2(\cdot), g\}$ to U_i.
(3) When U_i receives the smart card, he/she rewrites the random value r_i into the smart card and keeps it properly.

2.3. Authentication Phase

By executing this phase between U_i and S_j, they can verify each other's validity and establish a secure channel. The procedure unfolds as follows:

(1) U_i attaches his/her smart card to a card reader device and inputs UID_i and PW_i.
(2) U_i's smart card first computes $RPW_i'=H_1(PW_i \| UID_i \| r_i)$ and $s_{i,j}'=s_{i,j} \oplus RPW_i'$. Then, it randomly selects an integer $x_i \in Z_q^*$ and further calculates $X_i=x_i \cdot g$, $X_i'=X_i+ H_2\,(s_{i,j}')$ and $M_1=H_1(UID_i \| SID_j \| X_i' \| T_i)$, where T_i is the current timestamp. After that, the smart card forms and sends the authentication request message $\{UID_i, X_i', T_i, M_1\}$ to S_j.
(3) Upon the receipt of the message from U_i, S_j checks the validity of UID_i and T_i by verifying if $|T_j^c\text{-}\ T_i| \leq \Delta T$, where T_j^c is the current timestamp. If the validation is failed, the authentication request would be rejected. Moreover, S_j computes $M_1'=H_1(UID_i \| SID_j \| X_i' \| T_i)$. Then, it checks whether it holds that $M_1'=M_1$. If not, S_j terminates the authentication procedure; otherwise, it chooses a random integer $y_j \in Z_q^*$ and computes $s_{i,j}'=H_1(ASID_i \| UID_i)$, $X_i^*=X_i'\text{-}H_2(s_{i,j}')$, $Y_j=y_j \cdot g$, $Y_j'=X_i^*+Y_j$, $K_{j,i}=y_j \cdot X_i^*$ and $M_2=H_1(UID_i \| SID_j \| X_i' \| Y_j' \| K_{j,i} \| T_j)$, where T_j is the current timestamp. Subsequently, S_j sends the message $\{M_2, Y_j', T_j\}$ to U_i.
(4) After receiving the message from S_j, U_i's smart card first checks the validity of T_j by verifying if $|T_i{}^c\text{-}\ T_j| \leq \Delta T$, where T_j^c is the current timestamp. After that, it computes $Y_j^*=Y_j'\text{-}X_i$, $K_{i,j}=x_i \cdot Y_j^*$and $M_2'=H_1(UID_i \| SID_j \| X_i' \| Y_j' \| K_{i,j} \| T_j)$. Then, the smart card checks if $M_2' = M_2$. If the check is failed, the authentication procedure is terminated; otherwise, it successfully authenticates S_j and sends $M_3=H_1(X_i \| Y_j^* \| K_{i,j} \| T_i')$ where T_i' is the current timestamp.
(5) When receiving the message M_3 from U_i, S_j first checks the validity of T_i' by verifying if $|T_j^c\text{ - }T_i'| \leq \Delta T$, where T_j^c is the current timestamp. Then, S_j recomputes $M_3'=H_1(X_i^* \| Y_j \| K_{i,i} \| T_i')$ and checks if $M_3'=M_3$. If not, S_j terminates the authentication procedure; otherwise, U_i is successfully authenticated by S_j.

2.4. Password Update Phase

When U_i intends to modify their original password PW_i, the subsequent steps are undertaken:

(1) U_i randomly selects a service provider S_k, with whom U_i performs the authentication procedure.

(2) If both U_i and S_k pass through the authentication, then U_i selects a new password PW_i^{new} and lets the smart card compute $RPW_i^{new}=H_1(PW_i^{new}\|r_i)$, $s_{i,j}^{new}=s_{i,j}\oplus RPW_i\oplus RPW_i^{new}$ for j =1 to m.

(3) The smart card replaces $s_{i,j}$ with $s_{i,j}^{new}$ ($1 \le j \le m$).

3. Cryptanalysis on Zhao Et Al.'S Scheme

This section highlights the security shortcomings present in Zhao et al.'s user authentication scheme, contrary to their assertion that the scheme is immune to well-known attacks. The specific details of these deficiencies are outlined below.

3.1. Denial of Service Attack

A denial-of-service attack aims to disrupt the services of a host for intended users, rendering any server or network resource inaccessible to them. Owing to the absence of a design principle in Zhao et al.'s scheme, it fails to ensure availability even for legitimate users. The unavailability of the scheme can be characterized by either a) the failure to achieve the task claimed by the scheme or b) inconsistency in the parameters. In the case of Zhao et al.'s scheme, the second scenario applies.

To conduct a comprehensive analysis, it is crucial to examine both the registration phase and the authentication phase in Zhao et al.'s scheme. In the initial step of the registration phase, it defines RPW_i as follows:

(1) U_i selects a unique identity UID_i and a personal password PW_i easy to remember. Moreover, he/she randomly selects an integer $r_i \in Z_q^*$ and performs a computation of $RPW_i= H_1(PW_i\|r_i)$. Then, U_i sends the registration request message $\{UID_i, RPW_i\}$ to RC via a secure channel.

However, the subsequent second step of the authentication phase in Zhao et al.'s scheme utilizes RPW_i', which differs from the one employed in the registration phase.

(2) U_i's smart card first computes $RPW_i'=H_1(PW_i\|UID_i\|r_i)$ and $s_{i,j}'=s_{i,j}\oplus RPW_i'$. Then, it randomly selects an integer $x_i \in Z_q^*$ and further calculates $X_i=x_i\cdot g$, $X_i'= X_i+ H_2(s_{i,j}')$ and $M_1=H_1(UID_i\|SID_j\|X_i'\|T_i)$. After that, the smart card sends the message $\{UID_i, X_i', T_i, M_1\}$ to S_j.

Consequently, there exists parameter discrepancies between $RPW_i = H_1(PW_i\|r_i)$ and $RPW_i' = H_1(PW_i\|UID_i\|r_i)$, creating a scenario where the legitimate user may consistently face rejection by S_j during U_i's authentication process. This vulnerability makes Zhao et al.'s scheme susceptible to denial-of-service attack.

3.2. Privacy Attack

Privacy can be attained through the incorporation of anonymity and untraceability in an authentication scheme, which are crucial properties in contemporary communication schemes. In terms of privacy, a passive attacker has the ability to monitor an entity's communications without the entity's consent. After accumulating session messages, the attacker may launch privacy attacks on them.

For an in-depth analysis, it is essential to scrutinize the authentication phase in Zhao et al.'s scheme. It employs UID_i in the authentication message to S_j as follows:

(1) U_i attaches his/her smart card to a card reader and inputs UID_i and PW_i.

(2) U_i's smart card first computes $RPW_i'=H_1(PW_i \| UID_i \| r_i)$ and $s_{i,j}'=s_{i,j} \oplus RPW_i'$. Then, it randomly selects an integer $x_i \in Z_q^*$ and further calculates $X_i=x_i \cdot g$, $X_i'=X_i+H_2(s_{i,j}')$ and $M_1=H_1(UID_i \| SID_j \| X_i' \| T_i)$, where T_i is the current timestamp. After that, the smart card sends the message $\{UID_i, X_i', T_i, M_1\}$ to S_j.

Consequently, Zhao et al.'s scheme is susceptible to privacy attack.

3.3. No User Friendliness

The authentication scheme should not only ensure security and computational efficiency but also consider factors such as the burden imposed on the service provider during the password update phase, as excessive requirements can inconvenience the registered user. To enhance user-friendliness, users should have the flexibility to choose and change their passwords without the need for interaction with other entities in the network.

For a thorough examination, it is imperative to revisit the password update phase in Zhao et al.'s scheme. It necessitates S_k to facilitate the password update for U_i as follows:

(1) U_i randomly selects a service provider S_k, with whom he/she performs the authentication procedure.

Hence, the scheme proposed by Zhao et al. lacks user-friendliness in the password update phase.

4. Multi-Server User Authentication Scheme

To address the security vulnerabilities present in Zhao et al.'s scheme in [21], this section introduces a user authentication scheme with privacy preservation, incorporating fuzzy commitment as discussed in [22-24]. Before delving into the detailed scheme, it is essential to reiterate that Zhao et al.'s scheme exhibits security weaknesses against denial-of-service attack and the privacy attack, along with a deficiency in user-friendliness. The proposed scheme enables users to update their passwords without the need for assistance from other entities. Moreover, the authentication messages transmitted in the proposed scheme aim to enhance privacy by ensuring the anonymity and untraceability of the messages.

For the creation of a novel user authentication scheme, this section initiates by devising a fuzzy commitment leveraging the distinctive attributes of biometrics. Subsequently, we present a comprehensive design of the proposed multi-server user authentication scheme. Fig. 2 illustrates the overarching scheme, encompassing the initialization phase, registration phase, authentication phase, and password update phase.

4.1. Fuzzy Commitment

The fuzzy commitment employs error correction to rectify the attempted input of a biometric towards the committed value, provided the attempted input is in proximity to the committed value [22-24]. Otherwise, the correction process transforms it into a value entirely distinct from the committed one. A biometric template is generated through an irreversible transformation function, denoted as $f(\cdot)$. The measure of dissimilarity, termed as error, is calculated based on bitwise comparison between two biometrics. Error correction is effective only when the error's magnitude is below the capacity of the error-correcting code technique. The technique involves two primary steps: encoding, denoted as $\Psi_{enc}(\cdot)$, and decoding, denoted as $\Psi_{dec}(\cdot)$. Fuzzy commitment serves to conceal a secret under the security of a witness biometric. The secret can be unveiled using a witness only if it closely aligns with the witness utilized during the enrollment process [22].

In the proposed scheme, a randomly generated key, denoted as R_{Ki}, is employed and encoded as $K_{CWi}=\Psi_{enc}(R_{Ki})$. This encoded form, termed as a codeword, mimics an original biometric code. It is important to highlight that $\Psi_{dec}(\cdot)$ function can retrieve R_{Ki} from K_{CWi}, expressed as $R_{Ki}=\Psi_{enc}(K_{CWi})$. The transformation function $f(\cdot)$ utilizes a key, T_{Ki}, to convert the biometric data BIO_i into a biometric template C_i, defined as $C_i=f(BIO_i, T_{Ki})$. The computation of a locked data L_i involves performing bitwise exclusive OR operation between the template C_i and the codeword K_{CWi}, expressed as $L_i=C_i \oplus K_{CWi}$. The system takes careful measures to erase both the biometric template and the random secret from the fuzzy commitment. However, the system retains L_i and T_{Ki} for future utilization.

4.2. Initialization Phase

This phase is executed by the registration center (RC) to initialize the system based on a security parameter K, proceeding as follows:

(1) RC selects a prime number p of size K and proceeds to generate an elliptic curve $E_p(a, b)$ defined over Z_p^*, where both a and b belong to Z_p. Additionally, RC creates a cyclic group G with a prime order q derived from $E_p(a, b)$, and randomly selects a generator $g \in G$.
(2) RC randomly selects two integers, α and d, from Z_p^*, calculates $D=d\cdot g$, and opts for two cryptographically secure hash functions, $H_1(\cdot) : \{0, 1\}^* \rightarrow \{0, 1\}^n$ and $H_2(\cdot) : \{0, 1\}^* \rightarrow G$.
(3) RC publishes the system public parameters as $pp=\{E_p(a, b), g, D, H_1(\cdot), H_2(\cdot)\}$, making them accessible to all system users, while remaining α and d as the master secret keys.

4.3. Registration Phase

During this phase, individual system users U_i register with RC to acquire a smart card. Additionally, each service provider S_j undergoes registration with RC to secure a secret key, serving as S_j's credential to establish legitimacy with system users. The registration process between S_j and RC involves the following steps through a secure channel:

(1) S_j chooses a unique identity SID_j and sends it to RC.

Service provider S_j		Registration center RC
Selects SID_j	$\xrightarrow{\{SID_j\}}$	Chooses $d_{i,j}\in Z_p^*$
		Computes $D_{i,j}=d_{i,j}\cdot g$
	$\xleftarrow{\{ASID_j,d_{i,j}\}}$	$ASID_j=H_2(SID_j\|\alpha)$

(a) Server registration phase.

User U_i		Registration center RC
Selects UID_i, PW_i		
Chooses $r_i \in Z_q^*$		
Computes $RPW_i=H_1(PW_i\|r_i)$	$\xrightarrow{\{UID_i,RPW_i\}}$	For each $j \in [m]$, Computes $s_{i,j}=H_1(ASID_j\|UID_i)\oplus RPW_i$ Issues a smart card SC with

Inputs BIO_i	$\xleftarrow{\{SC\}}$	$\{\{s_{i,j},D_{i,j}\}_{j=1,\ldots,m},H_1(\cdot),H_2(\cdot),g,$
Chooses T_{Ki}, $R_{Ki}\in Z_q^*$		$f(\cdot),\Psi_{enc}(\cdot),\Psi_{dec}(\cdot)\}$
Computes $C_i=f(BIO_i, T_{Ki})$		
$K_{CWi}=\Psi_{enc}(R_{Ki})$		
$L_i=C_i\oplus K_{CWi}$		
$SH_i=RPW_i\oplus H_1(UID_i\Vert PW_i\Vert R_{Ki})$		
$SR_i=r_i\oplus H_1(UID_i\Vert PW_i\Vert R_{Ki})$		
Writes T_{Ki}, L_i, SH_i, SR_i into SC		

(b) User registration phase.

User U_i		Service provider S_j
Inputs UID_i', PW_i', BIO_i'		
Computes $C_i'=f(BIO_i', T_{Ki})$		
$R_{Ki}'=\Psi_{dec}(L_i\oplus C_i')$		
$r_i'=SR_i\oplus H_1(UID_i'\Vert PW_i'\Vert R_{Ki}')$		
$RPW_i'=SH_i\oplus H_1(UID_i'\Vert PW_i'\Vert R_{Ki}')$		
$RPW_i''=H_1(PW_i'\Vert r_i')$		
Validates RPW_i' ?= RPW_i''	$\xrightarrow{\{X_i,X_i',M_1,T_i\}}$	Verifies $\lvert T_j - T_i\rvert\le\Delta T$
Computes $s_{i,j}'=s_{i,j}\oplus RPW_i'$		Computes $UID_i'=X_i'\oplus X_i\cdot d_{i,j}$
Chooses $x_i\in Z_q^*$		$s_{i,j}''=H_1(ASID_j\Vert UID_i')$
Computes $X_i=x_i\cdot g$		$M_1'=H_1(UID_i'\Vert X_i\Vert X_i'\Vert s_{i,j}''\Vert T_i)$
$X_i'=x_i\cdot D_{i,j}\oplus UID_i$		Validates M_1' ?= M_1
$M_1=H_1(UID_i\Vert X_i\Vert X_i'\Vert s_{i,j}'\Vert T_i)$		Chooses $y_j\in Z_q^*$
		Computes $Y_j=y_j\cdot g$
		$Y_j'=H_2(s_{i,j}'')\oplus Y_j$
		$K_{j,i}=y_j\cdot X_i$
Validates $\lvert T_i'- T_j\rvert\le\Delta T$	$\xleftarrow{\{Y_j',M_2,T_j\}}$	$M_2=H_1(UID_i'\Vert X_i\Vert Y_j'\Vert Y_j\Vert K_{j,i}\Vert T_j)$
Computes $Y_j^*=Y_j'\oplus H_2(s_{i,j}')$		
$K_{i,j}=x_i\cdot Y_j^*$		
$M_2'=H_1(UID_i\Vert X_i\Vert Y_j'\Vert Y_j^*\Vert K_{i,j}\Vert T_j)$		
Validates M_2' ?= M_2	$\xrightarrow{\{M_3,T_i'\}}$	Validates $\lvert T_j^c- T_i'\rvert\le\Delta T$
Computes $M_3=H_1(X_i\Vert Y_j^*\Vert K_{i,j}\Vert T_i')$		Computes $M_3'=H_1(X_i\Vert Y_j\Vert K_{j,i}\Vert T_i')$
		Validates M_3' ?= M_3

(c) Authentication phase.

Figure 2. Multi-server user authentication scheme

(2) Upon receiving S_j's registration request, RC randomly selects an integer $d_{i,j}$ from Z_p^*, computes $D_{i,j}=d_{i,j}\cdot g$, and determines $ASID_j=H_2(SID_j\Vert\alpha)$. RC then sends the message $\{ASID_j, d_{i,j}\}$ to S_j.

(3) After receiving RC's response, S_j retains $ASID_j$ and $d_{i,j}$ as its secret key.

For U_i's registration with RC, they interactively conduct the following steps:

(1) U_i selects a unique identity UID_i and a password PW_i. Additionally, U_i generates a random integer $r_i\in Z_q^*$ and computes $RPW_i=H_1(PW_i\Vert r_i)$. U_i then sends the registration request message $\{UID_i, RPW_i\}$ to RC through the channel.

(2) Upon receiving U_i's registration request, RC computes $s_{i,j}=H_1(ASID_j\| UID_i)\oplus RPW_i$ for each $j \in [m]$ and issues a smart card containing $\{\{s_{i,j}, D_{i,j}\}_{j=1,\ldots,m}, H_1(\cdot), H_2(\cdot), g, f(\cdot), \Psi_{enc}(\cdot), \Psi_{dec}(\cdot)\}$ to U_i.

(3) After receiving the smart card, U_i imprints his/her fingerprint, obtains BIO_i, generates a random transformation parameter T_{Ki}, selects a random key R_{Ki}, and computes $K_{CWi}=\Psi_{enc}(R_{Ki})$, $C_i=f(BIO_i, T_{Ki})$, $L_i=C_i\oplus K_{CWi}$,$SH_i=RPW_i\oplus H_1(UID_i\|PW_i\|R_{Ki})$ and $SR_i=r_i\oplus H_1(UID_i\|PW_i\|R_{Ki})$. U_i then writes T_{Ki}, L_i, SH_i and SR_i into the smart card.

4.4. Authentication Phase

By executing this phase between U_i and S_j, they can verify the legitimacy of each entity and establish a secure channel with $K_{i,j}=K_{j,i}$. The process is carried out as follows:

(1) U_i inserts the smart card into the card reader, inputs UID_i' and PW_i', and scans his/her fingerprints to obtain BIO_i'.

(2) U_i's smart card calculates $C_i'=f(BIO_i', T_{Ki})$ and $R_{Ki}'=\Psi_{dec}(L_i\oplus C_i')$, retrieving $r_i'=SR_i\oplus H_1(UID_i'\|PW_i'\|R_{Ki}')$ and $RPW_i'=SH_i\oplus H_1(UID_i'\|PW_i'\|R_{Ki}')$. Subsequently, the smart card computes $RPW_i''=H_1(PW_i'\|r_i')$ and verifies if $RPW_i'=RPW_i''$. If validation fails, the ownership check is unsuccessful, and the smart card terminates the phase. Otherwise, it computes $s_{i,j}'=s_{i,j}\oplus RPW_i'$, selects a random integer $x_i \in Z_q^*$ and further calculates $X_i=x_i\cdot g$, $X_i'=x_i\cdot D_{i,j}\oplus UID_i$ and $M_1=H_1(UID_i\|X\ _i\|X_i'\|s_{i,j}'\|T_i)$, where T_i is the current timestamp. After that, the smart card sends a message $\{X_i, X_i', M_1, T_i\}$ to S_j.

(3) Upon receiving the message from U_i, S_j checks T_i by verifying if $|T_j - T_i| \leq \Delta T$ based on the current timestamp T_j. If not, the user's request is rejected. Moreover, S_j computes $UID_i'=X_i'\oplus X_i\cdot d_{i,j}$, $s_{i,j}''=H_1(ASID_j\|UID_i')$ and $M_1'=H_1(UID_i'\|X_i\|X_i'\|s_{i,j}''\| T_i)$. Then, S_j checks if $M_1'=M_1$. If not, S_j terminates the procedure. Otherwise, it chooses a random integer $y_j \in Z_q^*$ and computes $Y_j=y_j\cdot g$, $Y_j'=H_2(s_{i,j}'')\oplus Y_j$, $K_{j,i}=y_j\cdot X_i$ and $M_2=H_1(UID_i'\|X_i\|Y_j'\|Y_j\|K_{j,i}\|T_j)$, where T_j is the current timestamp. Subsequently, S_j sends a message $\{Y_j', M_2, T_j\}$ to U_i.

(4) After receiving the message from S_j, U_i's smart card checks T_j by verifying if $|T_i' - T_j| \leq \Delta T$, where T_i' is the current timestamp. After that, it computes $Y_j^*=Y_j'\oplus H_2(s_{i,j}')$, $K_{i,j}=x_i\cdot Y_j^*$ and $M_2'=H_1(UID_i\|X_i\|Y_j'\|Y_j^*\|K_{i,j}\|T_j)$. Then, the smart card checks if $M_2'=M_2$. If not, the procedure is terminated. Otherwise, it successfully authenticates S_j. After that, it computes $M_3=H_1(X_i\|Y_j^*\|K_{i,j}\|T_i')$ and sends a message $\{M_3, T_i'\}$ to S_j.

(5) When receiving the message from U_i, S_j checks T_i' by verifying if $|T_j^c - T_i'| \leq \Delta T$, where T_j^c is the current timestamp. Then, S_j computes $M_3'=H_1(X_i\|Y_j\|K_{j,i}\|T_i')$ and checks if $M_3'=M_3$. If not, S_j terminates the procedure. Otherwise, U_i is successfully authenticated by S_j.

4.5. Password Update Phase

When U_i wishes to update their password PW_i, the following steps are carried out:

(1) U_i inserts their smart card into the card reader, inputs UID_i' and PW_i', and scans their fingerprints to get BIO_i' with a new password PW_i^{new}.

(2) U_i's smart card computes $C_i'=f(BIO_i', T_{Ki})$ and $R_{Ki}'=\Psi_{dec}(L_i\oplus C_i')$ and retrieves $r_i'=SR_i\oplus H_1(UID_i'\|PW_i'\|R_{Ki}')$ and $RPW_i'=SH_i\oplus H_1(UID_i'\|PW_i'\|R_{Ki}')$. Afterward, the smart card computes $RPW_i''=H_1(PW_i'\|r_i')$ and verifies if $RPW_i'=RPW_i''$. If the validation fails, the ownership check is unsuccessful, and the smart card terminates the phase. Otherwise, U_i computes $RPW_i^{new}=H_1(PW_i^{new}\|r_i')$, $SH_i^{new}=RPW_i^{new}\oplus H_1(UID_i'\|PW_i^{new}\|R_{Ki}')$, $SR_i^{new}=r_i'\oplus H_1(UID_i'\|PW_i^{new}\|R_{Ki}')$ and $s_{i,j}^{new}=s_{i,j}\oplus RPW_i'\oplus RPW_i^{new}$ for j =1 to m.

(3) The smart card substitutes RPW_i, SH_i, SR_i and $s_{i,j}$($1 \leq j \leq m$) with RPW_i^{new}, SH_i^{new}, SR_i^{new} and $s_{i,j}^{new}$($1 \leq j \leq m$), respectively.

5. Security Analysis

In this section, we present security analysis of the proposed authentication scheme. Initially, we conduct BAN logic analysis, ProVerif validation, and informal security analysis to demonstrate the security and privacy of the proposed scheme [25-27].

5.1. BAN Logic Analysis

This section presents a formal analysis of the proposed scheme based on the [25]. BAN logic is commonly employed to verify the correctness of the security scheme. The correctness of the proposed scheme implies that upon successful completion, the two communicating parties, a user U_i and a service provider S_j, mutually possess a fresh session key $K_{ij}=K_{ji}$. The formal analysis of the proposed scheme encompasses the following steps:

(1) Transforming the messages in the scheme to their idealized form.
(2) Identifying the required assumptions for the initial system setup.
(3) Expressing the system's state after each message operation as logical assertions, attaching logical formulas to each message.
(4) Applying logical postulates to the assumptions and the assertions.

The notations utilized in the analysis are as follows:

- $Q |\equiv X$: Principal Q believes the statement X.
- $\#(X)$: Formula X is considered fresh.
- $Q|\Rightarrow X$: Principal Q has jurisdiction over statement X.
-$|\xrightarrow{D} Q$: Principal Q has a public key D.
- $Q \blacktriangleleft X$: Principal Q observes statement X.
- $Q|\sim X$: Principal Q previously stated statement X.
- (X, Y): Either formula X or Y constitutes one part of the formula (X, Y).
- $<P>_Q$: Formula P combined with formula Q.
- $Q \overset{SK}{\leftrightarrow} R$: Principals Q and R can use the shared session key, SK, to communicate with each other.

This SK is secure and will not be discovered by any other principals except Q and R.

In addition to this, the following rules are utilized to demonstrate that the proposed scheme ensures secure mutual authentication between U_i and S_j:

Rule 1. Message-meaning rule: $\frac{R |\equiv R \overset{Y}{\leftrightarrow} S,\ R \triangleleft <X>_Y}{R|\equiv S |\sim X}$
Rule 2. Nonce-verification rule: $\frac{R|\equiv \#(X),\ R |\equiv S |\sim X}{R|\equiv S |\equiv X}$
Rule 3. Jurisdiction rule: $\frac{R |\equiv S |\Rightarrow X,\ R |\equiv S |\equiv X}{R|\equiv X}$
Rule 4. Freshness-concatenation rule: $\frac{R |\equiv \#(X)}{R|\equiv \#(X,Y)}$

To analyze the security of mutual authentication between U_i and S_j in the proposed scheme, we aim to accomplish the following goals:

Goal 1: $U_i \mid\equiv (U_i \overset{K_{i,j}}{\longleftrightarrow} S_j)$
Goal 2: $S_j \mid\equiv (S_j \overset{K_{i,I}}{\longleftrightarrow} U_i)$
Goal 3: $U_i \mid\equiv S_j \mid\equiv (S_j \overset{K_{i,I}}{\longleftrightarrow} U_i)$
Goal 4: $S_j \mid\equiv U_i \mid\equiv (U_i \overset{K_{i,j}}{\longleftrightarrow} S_j)$

Idealized form: The messages exchanged between U_i and S_j in the proposed scheme are transformed into their idealized form as outlined below:

Message AM1. $U_i \rightarrow S_j$: $\{<X_i>, <X_i'>_{di,j}, <M_1>_{si,j}, <T_i>\}$
Message AM2. $S_j \rightarrow U_i$: $\{<Y_j'>_{si,j}, <M_2>_{Kj,i}, <T_j>\}$
Message AM3. $U_i \rightarrow S_j$: $\{<M_3>_{Ki,j}, <T_i'>\}$

Assumptions: The initial assumptions for the proposed scheme are outlined as follows:

A1: $U_i \mid\equiv \#(x_i, T_i, T_i')$
A2: $S_j \mid\equiv \#(y_j, T_j, T_j^c)$
A3: $U_i \mid\equiv (U_i \overset{s_{i,j}}{\longleftrightarrow} RC)$
A4: $RC \mid\equiv (RC \overset{s_{i,j}}{\longleftrightarrow} U_i)$
A5: $S_j \mid\equiv (S_j \overset{ASID_j}{\longleftrightarrow} RC)$
A6: $RC \mid\equiv (RC \overset{ASID_j}{\longleftrightarrow} S_j)$
A7: $S_j \mid\equiv (S_j \overset{d_{i,j}}{\longleftrightarrow} RC)$
A8: $RC \mid\equiv (RC \overset{d_{i,j}}{\longleftrightarrow} S_j)$

Proof:

Henceforth, we will demonstrate the achievement of the four specified goals to establish the security of the proposed authentication, utilizing the BAN logic rules and the provided assumptions.

Based on AM1, we could derive:

Step 1. $S_j \blacktriangleleft \{<X_i>, <X_i'>_{di,j}, <M_1>_{si,j}, <T_i>\}$
According to A3 and the message meaning rule, we could get:
Step 2. $S_j \mid\equiv U_i \mid\sim (<X_i>, <X_i'>_{di,j}, <M_1>_{si,j}, <T_i>)$
According to A1 and the freshness concatenation rule, we could get:
Step 3: $S_j \mid\equiv \#(<X_i>, <X_i'>_{di,j}, <M_1>_{si,j}, <T_i>)$
According to Steps 2 and 3 and the nonce verification rule, we could get:
Step 4. $S_j \mid\equiv U_i \mid\equiv (<X_i>, <X_i'>_{di,j}, <M_1>_{si,j}, <T_i>)$
According to Step 4, A2, A7 and the believe rule, we could get:
Step 5. $S_j \mid\equiv U_i \mid\equiv (U_i \overset{s_{i,j}}{\longleftrightarrow} S_j)$
According to the jurisdiction rule, we could get:
Step 6. $S_j \mid\equiv (S_j \overset{s_{i,j}}{\longleftrightarrow} U_i)$
Based on AM2, we could derive
Step 7. $U_i \blacktriangleleft \{<Y_j'>_{si,j}, <M_2>_{Kj,i}, <T_j>\}$
According to A7 and the message meaning rule, we could get:
Step 8. $U_i \mid\equiv S_j \mid\sim (<Y_j'>_{si,j}, <M_2>_{Kj,i}, <T_j>)$
According to A1 and the freshness concatenation rule, we could get:
Step 9: $U_i \mid\equiv \#(<Y_j'>_{si,j}, <M_2>_{Kj,i}, <T_j>)$
According to Steps 8 and 9 and the nonce verification rule, we could get:
Step 10. $U_i \mid\equiv S_j \mid\equiv (<Y_j'>_{si,j}, <M_2>_{Kj,i}, <T_j>)$
According to Step 10, A1, A3 and the believe rule, we could get:

Step 11. $U_i|\equiv S_j|\equiv(S_j \xleftrightarrow{s_{i,j}} U_i)$
According to the jurisdiction rule, we could get:
Step 12. $U_i|\equiv(U_i \xleftrightarrow{s_{i,j}} S_j)$
According to Steps 8, 9 and 10 and the nonce verification rule, we could get:
Step 13. $U_i|\equiv S_j|\equiv(S_j \xleftrightarrow{K_{j,i}} U_i)$ (**Goal 3**)
According to A3 and the jurisdiction rule, we could get:
Step 14. $U_i|\equiv(U_i \xleftrightarrow{K_{i,j}} S_j)$ (**Goal 1**)
Based on AM3, we could derive
Step 15. $S_j \blacktriangleleft \{<M_3>_{Ki,j}, <T_i'>\}$
According to A6 and the message meaning rule, we could get:
Step 16. $S_j|\equiv U_i|\sim(<M_3>_{Ki,j}, <T_i'>)$
According to A2 and the freshness concatenation rule, we could get:
Step 17: $S_j|\equiv \#(<M_3>_{Ki,j}, <T_i'>)$
According to Steps 16 and 17 and the nonce verification rule, we could get:
Step 18. $S_j|\equiv U_i|\equiv(<M_3>_{Ki,j}, <T_i'>)$
According to Step 18, A7 and the believe rule, we could get:
Step 19. $S_j|\equiv U_i|\equiv(U_i \xleftrightarrow{s_{i,j}} S_j)$
According to Steps 16, 17 and 18 and the nonce verification rule, we could get:
Step 20. $S_j|\equiv(S_j\ s_{i,j}\ U_i)$
According to Steps 18, 19 and 20, the nonce verification rule and the jurisdiction rule, we could get:
Step 21. $S_j|\equiv U_i|\equiv(U_i \xleftrightarrow{K_{i,j}} S_j)$ (**Goal 4**)
According to A7 and the jurisdiction rule, we could get:
Step 22. $S_j|\equiv(S_j \xleftrightarrow{K_{j,i}} U_i)$ (**Goal 2**)

In summary, based on Steps 14 and 22, it can be inferred that the proposed scheme effectively fulfills both goals (**Goals 1 and 2**). Consequently, U_i and S_j have confidence in securely sharing a common session key $K_{i,j}=K_{j,i}=x_i{\cdot}y_j{\cdot}g$.

5.2. ProVerif Validation

In this section, we conduct a formal security verification of the proposed scheme using the ProVerif tool within the Dolev-Yao threat model [26-27]. ProVerif is an automatic cryptographic scheme verifier. It is based on a representation of the scheme by Horn clauses [28]. ProVerif can prove the following properties: secrecy that the adversary cannot obtain the secret, authentication and more generally correspondence properties, strong secrecy that the adversary does not see the difference when the value of the secret changes and equivalences between processes that differ only by terms. The ProVerif code is structured into declarations, process macros, and main process.

The declaration part encompasses all the necessary definitions for the ProVerif, including variables, constants, functions, equations, events, transmission channels, and etc. The channel ch is used as a public communication channel between U_i and S_j:
free ch:channel.

In the context of the proposed scheme, g serves as a generator of an elliptic curve. The authentication scheme involves three participants: U_i, S_j and RC. UIDi and SIDj are the unique identifiers for U_i and S_j, respectively. The parameter “a” represents the secret key of RC. Sj utilizes a key pair, dij and Dij, denoting the private and public keys. Two crucial free names, svalueA and svalueB, play a pivotal role in verifying the session key related to Kij and Kji.

```
free Ui, Sj, RC: entity.
free g: bitstring.
free UIDi: bitstring.
free SIDj: bitstring.
free a: bitstring[private].
free dij: bitstring[private].
free Dij: bitstring.
free svalueA, svalueB: bitstring [private].
```

The secure one-way hash functions are denoted by H1() and H2(). Fuzzy commitment and error-correcting code encryption and decryption are facilitated by the functions f(), Wenc() and Wdec(). The XOR() function models the XOR operation, and Mul() represents multiplication in elliptic curve cryptography. The bit-concatenation function is denoted by Con().

```
fun H1(bitstring): bitstring.
fun H2(bitstring): bitstring.
fun f(bitstring, key): bitstring.
fun Wenc(bitstring): bitstring.
fun Wdec(bitstring): bitstring.
fun XOR(bitstring, bitstring): bitstring.
equation forall x: bitstring, y: bitstring;
XOR(XOR(x, y), y) = x.
fun Con(bitstring, bitstring): bitstring.
fun Mul(bitstring, bitstring): bitstring.fun h(bitstring):bitstring.
fun nontobit(nonce): bitstring [data,typeConverter].
fun bittokey(bitstring): key [data,typeConverter].
```

In the second part, actions for each entity are structured as follows.

Registration phase:
Message RM1: Ui–> RC:{UIDi, RPWi}
Message RM2: RC–> Ui:{SC}
Authentication phase:
Message AM1: Ui–> RC:{Xi, xXi, M1, Ti}
Message AM2: RC–> Ui:{xYj, M2, Tj}
Message AM3: Ui–> RC:{M3, xTi}

The registration phase must be executed through a secure channel. Consequently, our ProVerif simulation will exclusively focus on the authentication phase, which involves operations over the public channel ch. The ProVerif code processUi is structured as follows:

```
let (xCi: bitstring) = f(BIOi, TKi) in
let (xRKi: bitstring) = Wdec(XOR(Li, xCi)) in
let (xri: bitstring) = XOR(SRi, H1(Con(Con(UIDi, PWi), xRKi)) in
let (xRPWi: bitstring) = XOR(SHi, H1(Con(Con(UIDi, PWi), xRKi)) in
if xRPWi = RPWi then
    event SUbegin(Sj);
    let (xsij: bitstring) = XOR(sij, xRPWi) in
    new xi: nonce;
    new Ti: nonce;
    let (Xi: bitstring) = Mul(nontobit(xi), g) in
    let (xXi: bitstring) = XOR(Mul(nontobit(xi), Dij), UIDi) in
    let (M1: bitstring) = H1(Con(Con(Con(Con(UIDi, Xi), xXi), xsij), nontobit(Ti))) in
(*US1*) out(ch, (Xi, xXi, M1, Ti, true));
(*SU1*) in(ch, (xYj: bitstring, M2: bitstring, Tj: nonce));
new xTi: nonce;
let (xxYj: bitstring) = XOR(xYj, H2(xsij)) in
let (Kij: bitstring) = Mul(nontobit(xi), xxYj) in
let (xM2: bitstring) = H1(Con(Con(Con(Con(UIDi, Xi), xYj), Kij), nontobit(Tj))) in
if xM2 = M2 then
            let (M3: bitstring) = H1(Con(Con(Con(Xi, xxYj), Kij), nontobit(xTi))) in
            (*US2*) out(ch, (M3, xTi, true));
            event USend(Ui);
(* OK *)
            out(ch, Enc(svalueA, bittokey(Kij))).
```

The ProVerif codes processSj are designed as:

```
in(ch, (Xi: bitstring, xXi: bitstring, M1: bitstring, Ti: nonce));
new Tj: nonce;
let (xUIDi: bitstring) = XOR(xXi, Mul(Xi, dij)) in
let (xxsij: bitstring) = H1(Con(ASIDj, xUIDi)) in
let (xM1: bitstring) = H1(Con(Con(Con(Con(xUIDi, Xi), xXi), xxsij), nontobit(Ti))) in
if xM1 = M1 then
          event USbegin(Ui);
    new yj: nonce;
    let (Yj: bitstring) = Mul(nontobit(yj), g) in
    let (xYj: bitstring) = XOR(H2(xxsij), Yj) in
    let (Kji: bitstring) = Mul(nontobit(yj), Xi) in
let (M2: bitstring) = H1(Con(Con(Con(Con(Con(xUIDi, Xi), xYj), Yj), Kji), nontobit(Tj))) in
(*SU1*) out(ch, (xYj, M2, Tj, true));
(*US2*) in(ch, (M3: bitstring, xTi: nonce));
new xTj: nonce;
let (xM3: bitstring) = H1(Con(Con(Con(Xi, Yj), Kji), nontobit(xTi))) in
if xM3 = M3 then
           event SUend(Sj);
    (* OK *)
    out(ch, Enc(svalueB, bittokey(Kji))).
```

Given that the execution processes of multiple participants are concurrently modeled, an exclamation point (!) precedes each subprocess.

```
!processUi(UIDi, PWi, BIOi, RPWi, TKi, Li, SHi, SRi, sij)|!processSj(ASIDj)
```

We verify the confidentiality of the session key Kij=Kji through the queries made by the attacker. The ProVerif query codes are defined as follow.

```
query attacker(svalueA);
     attacker(svalueB).
```

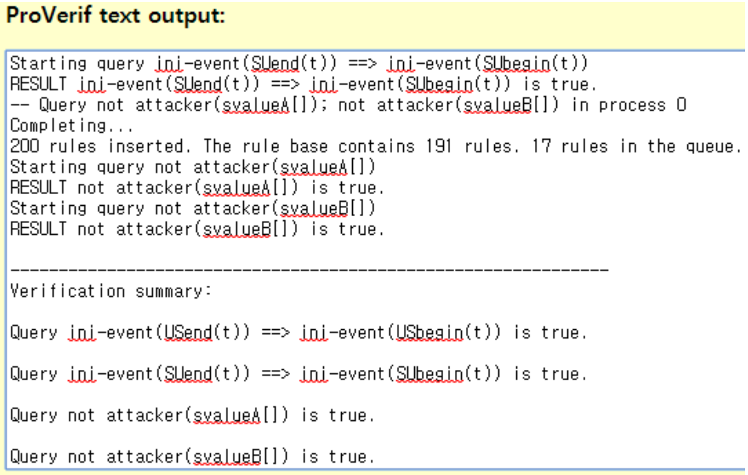
Fig. 3 illustrates that the expression attacker (svalueA)/attacker(svalueB) is false in the results of the attacker query. This indicates that the session key is secure, and the attacker cannot obtain it through any means. Four events are employed to assess the attainability of authentication.

```
event USbegin(entity).
event USend(entity).
event SUbegin(entity).
event SUend(entity).
```

We employ correspondence assertions to validate the authentication properties of two participants. In the formal proof, we establish two authentication correlations, USbegin(entity) and USend(entity). USbegin(entity) signifies the initiation of the record where Ui has executed the process with Sj. USend(entity) denotes the conclusion of the record where Ui concludes the process with Sj. The remaining events are analogous to these two. The reachabilities of events are verified through the following ProVerif queries.

```
query t: entity; inj-event(USend(t)) ==> inj-event(USbegin(t)).
query t: entity; inj-event(SUend(t)) ==> inj-event(SUbegin(t)).
```

Fig. 3 illustrates the results of the two corresponding queries, both yielding “true.” This indicates that the proposed scheme fulfills all the security requirements for authentication.



```
Starting query inj-event(SUend(t)) ==> inj-event(SUbegin(t))
RESULT inj-event(SUend(t)) ==> inj-event(SUbegin(t)) is true.
-- Query not attacker(svalueA[]); not attacker(svalueB[]) in process 0
Completing...
200 rules inserted. The rule base contains 191 rules. 17 rules in the queue.
Starting query not attacker(svalueA[])
RESULT not attacker(svalueA[]) is true.
Starting query not attacker(svalueB[])
RESULT not attacker(svalueB[]) is true.

--------------------------------------------------------------
Verification summary:

Query inj-event(USend(t)) ==> inj-event(USbegin(t)) is true.

Query inj-event(SUend(t)) ==> inj-event(SUbegin(t)) is true.

Query not attacker(svalueA[]) is true.

Query not attacker(svalueB[]) is true.
```

Figure 3. The result of ProVerif validation

5.3. Informal Security Analysis

This section adopts security analysis methodologies akin to those in [29-32]. Emphasizing a comprehensive evaluation of the proposed scheme’s security requirements, encompassing both passive and active attacks, it draws comparisons with Yeh’s scheme, Truong et al.’s scheme, and Zhao et al.’s scheme, as detailed in Table 2 [19-21].

Proposition 1. [S1]: The security of the proposed scheme against a service provider S_j masquerading attack is established.

Proof: Defined as an attack where an adversary assumes the identity of a registered service provider S_j and has the capability to intercept any transmitted message from previous sessions to the user U_i. In the proposed scheme, the attacker might attempt an attack by constructing AM2=$\{Y_j', M_2, T_j\}$ immediately after receiving AM1=$\{X_i, X_i', M_1, T_i\}$ from U_i. However, such an attack necessitates not only knowledge of the crucial secret key $s_{i,j}$ but also awareness of $d_{i,j}$ for S_j. Lacking this essential information, the attacker cannot construct the appropriate message AM2 corresponding to AM1. Therefore, it is affirmed that the proposed scheme effectively thwarts $S_j$masquerading attack.

Table 2. Comparison of security and privacy features

Features	Scheme in [19]	Scheme in [20]	Scheme in [21]	Proposed scheme
S1	Weak	Weak	Strong	Strong
S2	Weak	Weak	Strong	Strong
S3	Weak	Strong	Strong	Strong
S4	Weak	Weak	Strong	Strong
S5	Weak	Strong	Weak	Strong
P1	Not	Not	Not	Provide

S1: S_j masquerading attack, S2: U_i masquerading attack, S3: replay attack, S4: password guessing attack, S5: denial of service attack, P1: privacy

Proposition 2. [S2]: The proposed scheme is resilient against user masquerading attack.

Proof: In alignment with the definition of the attack involving S_j masquerading, an attacker aiming to masquerade as a legitimate user U_i must construct a valid authentication request message AM1=$\{X_i, X_i', M_1, T_i\}$ sent to S_j. The creation of a proper integrity value M_1 necessitates knowledge of $s_{i,j}$, a crucial secret key in the proposed scheme. However, this proves unattainable for the attacker, as acquiring the secret key of the targeted party is not within reach. Consequently, the attacker is unable to construct the message AM1 accurately. Thus, the proposed scheme effectively guards against user masquerading attack.

Proposition 3. [S3]: The proposed scheme effectively guards against replay attack.

Proof: In the context of a replay attack, an assailant might capture previously transmitted messages and attempt to reuse them during the execution of the proposed scheme. The aim is to deceive the recipient into believing that the transmitted message originates from a legitimate entity. To establish the resilience of the proposed scheme to this form of attack, we assume that the attacker possesses the ability to capture previous session messages within the proposed scheme and subsequently endeavors to transmit identical messages to the target entity. To counter this potential threat, the proposed scheme employs a challenge and response mechanism, complemented by a timestamp that is exclusively vaild for the session and cannot be reused. For instance, in preparing a message AM1=$\{X_i, X_i', M_1, T_i\}$, U_i utilizes a session random number x_i and a time stamp T_i. Similarly, when composing a message AM2=$\{Y_j', M_2, T_j\}$, S_j generates a random number y_j accompanied by a time stamp T_j. Another layer of defense against replay attack involves an integrity check on each message M_i. Specifically, either a timestamp-based mechanism or a challenge and response mechanism is incorporated to ensure the freshness of each message in the authentication scheme. By implementing these preventative measures, we can conclude that the proposed scheme is resilient to replay attack.

Proposition 4. [S4]: The proposed scheme is resilient against password guessing attack.

Proof: A password guessing attack involves numerous attempts to guess passwords or passphrases, with the hope of eventually guessing correctly. We assume that an attacker can pilfer a legitimate user U_i's smart card and extract the information stored on it $\{\{s_{i,j}, D_{i,j}\}_{j=1,\ldots,m}, H_1(\cdot), H_2(\cdot), g, \Psi_{enc}(\cdot), \Psi_{dec}(\cdot), T_{Ki}, L_i, SH_i, SR_i\}$ using power analysis methods [21]. To succeed in such an attack, the attacker would need simultaneous knowledge of UID_i, RPW_i and BIO_i simultaneously. This computational infeasibility arises from the one-way nature of the hash function, making it implausible for the attacker to guess these parameters simultaneously. Hence, the proposed scheme is effectively immune to password guessing attack.

Proposition 5. [S5]: The proposed scheme is resilient against denial of service attack.

Proof: Denial of service is an attack wherein tha assailant endeavors to render a scheme inaccessible to its designated users by intermittently or permanently disrupting the services of a host connected to the Internet. In the proposed scheme, there are two potential scenarios where a registered entity might face denial of service. In the first situation, if U_i unintentionally inputs incorrect credentials (UID_i, RPW_i and BIO_i) during the authentication phase, the smart card can verify U_i's credentials using the validation check of $RPW_i' \; ?= RPW_i''$. This ensures that an authentication request message AM1=$\{X_i, X_i', M_1, T_i\}$ can only be formed with the correct input credentials. Another potential scenario involves an adversary attempting to engage another entity by replaying messages, leading to potential denial or delay. However, each message in the proposed scheme undergoes freshness and validity checks, allowing legal entities to identify and thwart such attempts. Consequently, the proposed scheme is adept at mitigating denial of service attack.

Proposition 6. [P1]: The proposed scheme ensures privacy.

Proof: The privacy analysis considers both of anonymity and untraceability. An entity in the scheme achieves anonymity if its identity remains unknown to other entities. Untraceability is maintained if an attacker cannot discern any links between messages communicated across sessions by monitoring the communication. In the proposed scheme, two messages pertain to user identity: $\{X_i, X_i', M_1, T_i\}$ and $\{Y_j', M_2, T_j\}$. Regarding anonymity, the attacker attempting to extract UID_i from $X_i'=x_i \cdot D_{i,j} \oplus UID_i$, $M_1 = H_1(UID_i \| X_i \| X_i' \| s_{i,j}' \| T_i)$ and $M_2 = H_1(UID_i' \| X_i \| Y_j' \| Y_j \| K_{j,i} \| T_j)$ faces computational infeasibility due to the elliptic curve discrete logarithm problem and the one-wayness of the hash function. For a trace attack, the attacker must find common factors among messages intercepted from various legal entities in the scheme, even after intercepting two or more session messages. However, the proposed scheme prevents tracing U_i and S_j by using session fresh random numbers with timestamps, in addition to addressing the one-way feature of the hash function. Therefore, the proposed scheme ensures both anonymity and untraceability.

6. Performance Analysis

This section presents an evaluation of computational and communication overheads associated with the proposed scheme, comparing it with three related schemes [19-21]. The proposed scheme comprises four phases: initialization, registration, authentication, and password update. The focus of the performance analysis is on the authentication phase, as it is the most frequently utilized phase in an open channel.

To assess the performance of the proposed scheme, we adopt a scale introduced by Wu et al. utilizing MIRACL [31-32]. The proposed scheme involves three fundamental operations: XOR, hash and scalar multiplication. Given that XOR operations are relatively lightweight and negligible compared to the others, our analysis will focus on hash and scalar multiplication operations. The costs for hash (t_h) and scalar multiplication (t_m) operations are 0.005174 ms and 0.427576 ms, respectively.

6.1. Computation Overhead

Table 3 provides a comparison of the computational overhead comparison between the proposed scheme and the related schemes in [19-21].

The results in Table 3 indicate that the proposed scheme incurs approximately 67% less overhead than Truong et al.'s and Zhao et al.'s schemes, while exhibiting similar overhead to Yeh's scheme. It is noteworthy that the proposed authentication scheme achieves superior functional properties, ensuring the security and privacy necessary for a multi-server environment. For the functional properties, the proposed scheme is carefully designed by incorporates fuzzy commitment and to enable users to update their passwords without the need for assistance from other entities.

Table 3. Comparison of computation overhead

Entity	Scheme in [19]	Scheme in [20]	Scheme in [21]	Proposed scheme
U_i	$4t_h + 2t_m$ (0.875848 ms)	$5t_h + 2t_m$ (0.881022 ms)	$6t_h + 2t_m$ (0.886196 ms)	$8t_h + 3t_m$ (1.32312 ms)
S_j	$4t_h + 4t_m$ (1.731 ms)	$6t_h + 2t_m$ (0.886196 ms)	$6t_h + 2t_m$ (0.886196 ms)	$5t_h + 3t_m$ (1.308598 ms)
Total	$8t_h + 6t_m$ (2.606848 ms)	$11t_h + 4t_m$ (1.767218 ms)	$12t_h + 4t_m$ (1.772392 ms)	$13t_h + 6t_m$ (2.632718 ms)

6.2. Communication Overhead

We evaluate communication overhead based on the bit length of messages. For identity, random number, timestamp, one-way hash function, and secret key, we consider a length of 128 bits each. Additionally, the length of ECC operation is set at 160 bits. Table 4 provides a comparison of communication costs between the proposed scheme and the related schemes in [19-21].

Table 4. Comparison of communication overhead

Entity	Scheme in [19]	Scheme in [20]	Scheme in [21]	Proposed scheme
U_i	2*128+1*160	4*128+1*160	5*128+1*160	4*128+2*160
S_j	1*128+1*160	1*128+1*160	2*128+1*160	2*128+1*160
Total	3*128+2*160 (704 bits)	5*128+2*160 (960 bits)	7*128+2*160 (1,216 bits)	6*128+3*160 (1,248 bits)

The communication message lengths needed in the authentication schemes of Yeh, Truong et al., Zhao et al. and the proposed scheme are 704 bits, 960 bits, 1,216 bits and 1,248 bits, respectively. Consequently, the proposed scheme exhibits comparable overhead to Zhao et al.'s scheme but incurs more overhead than Yeh scheme and Truong et al.'s scheme, attributed to its enhanced functionality, security and privacy features.

7. CONCLUSION

This paper introduces a multi-server user authentication scheme with privacy preservation founded on fuzzy commitment. In contrast to Zhao et al.'s authentication scheme, which exhibits vulnerabilities to denial of service and privacy attacks while lacking user friendliness, the proposed scheme prioritizes anonymity and untraceability-critical features in contemporary network environments. The formal security analysis utilizes BAN logic and the ProVerif tool, offering comprehensive verification of the proposed scheme. Additionally, a performance analysis is conducted to underscore the reliability and efficiency of the proposed scheme. Although the computational and communication costs are slightly higher compared to related schemes, they are justified by the additional functionality, enhanced security, and privacy provisions offered by the proposed scheme.

ACKNOWLEDGEMENTS

The corresponding author is Hyunsung Kim. This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A1B04032598).

REFERENCES

[1] Chiphiko, B. A. & Kim, H. (2023). Machine to Machine Authenticated Key Agreement with Forward Secrecy for Internet of Things. International Journal of Computer Networks & Communications, *15*(6), 27-53.

[2] Trabelsi, R., Fersi, G. & Jmaiel, M. (2023). Access control in Internet of Things: A survey. *Computers & Security*, *135*, 103472.

[3] Aski, V. J., Dhaka, V. S., Parashar, A., Kumar, S. & Rida, I. (2023). Internet of Things in healthcare: A survey on protocol standards, enabling technologies, WBAN architectures and open issues. *Physical Communication*, *60*, 102103.

[4] Manocha, A., Sood, S. K. & Bhatia, M. (2024). IoT-Dew Computing-Inspired Real-Time Monitoring of Indoor Environment for Irregular Health Prediction. *IEEE Transactions on Engineering Management*, *71*, 1669-1682.

[5] Metcalfe, B. (2013). Metcalfe's Law after 40 Years of Ethernet. *Computers*, *46*(1), 26-31.

[6] Marin, A. & Rossi, S. (2017). Fair workload distribution for multi-server systems with pulling strategies. *Performance evaluation*, *113*, 26-41.

[7] Haq, I., Wang, J., Zhu, Y & Maqbool, S. (2020). A survey of authenticated key agreement protocols for multi-server architecture. *Journal of Information Security and Applications*, *55*, 102639.

[8] Wang, D., Zhang, X., Zhang, Z. & Wang, P. (2020). Understanding security failures of multi-factor authentication schemes for multi-server environments. *Computers & Security* , *88*, 101619.

[9] Ijemaru, G. K., Adeyanju, I. A., Olusuyi, K. O., Ofusori, T. J., Ngharamike, E. T. & Sobowale, A. A. (2018). Security Challenges of Wireless Communications Networks: A Survey. *International Journal of Applied Engineering Research*, *13*(8), 5680-5692.

[10] HaddaPajouh, H., Dehghantanha, A., Parizi, R. M. Aledhari, M. & Karimipour, H. (2019). A survey on internet of things security: Requirements, challenges, and solutions. *Internet of Things*, 100129.

[11] Tsaur, W. (2001). A Flexible User Authentication Scheme for Multi-server Internet Services. *Lecture Notes in Computer Science*, *2093*, 174-183.

[12] Kim, S., Lim, S. & Won, D. (2002). Cryptanalysis of flexible remote password authentication scheme of ICN'01. *Electronics Letters*, *38*(24), 1519-1520.

[13] Tsaur, W., Wu, C. & Lee, W. (2005). An enhanced user authentication scheme for multi-server Internet services. *Applied Mathematics and Computation*, *170*(1), 258-266.

[14] Wu, F., Xu, L. & Li, X. (2018). A New Chaotic Map-Based Authentication and Key Agreement Scheme with User Anonymity for Multi-server Environment. *Lecture Notes in Electrical Engineering*, *464*, 335-344.

[15] Nkhoma, S. K., Ali, P., Eneya, L. & Kim, H. (2018). Unlinkable User Authenticated Key Agreement for Multi-Gateway Wireless Sensor Networks. *Current Analysis on Communication Engineering*, *1*, 20-30.

[16] Liao, Y. P. & Wang, S. S. (2009). A secure dynamic ID based remote user authentication scheme for multi-server environment. *Computer Standards & Interface*. *31*(1), 24-29.

[17] Hsiang, H. C., &Shih, W. K. (2009). Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. *Computer Standards & Interfaces*, *31*(6), 1118-1123.

[18] Sood, S. K., Sarje, A. K. & Singh, K. (2011). A secure dynamic identity based authentication protocol for multi-server architecture. *Journal of Network and Computer Applications*, *34*(2), 609-618.

[19] Yeh, K. H. (2014). A Provably Secure Multi-server Based Authentication Scheme. *Wireless Personal Communications*, *19*(3), 1621-1624.

[20] Truong, T. T., Tran, M., Duong, A. & Echizen, I. (2017). Provable Identity Based User Authentication Scheme on ECC in Multi-server Environment. *Wireless Personal Communications*, *95*, 2785-2801.

[21] Zhao, Y., Li, S., Jiang, L. (2018). Secure and Efficient User Authentication Scheme Based on Password and Smart Card for Multiserver Environment. *Security and Communication Networks*, 9178941.

[22] Juels, A. & Wattenberg, M. (1999). A Fuzzy Commitment Scheme. In *Proc. of ACM Conference on Computer and Communications Security*, 28-36.

[23] Choi, D., Seo, S., Oh, Y. & Kang, Y. (2019). Two-factor Fuzzy Commitment for Unmanned IoT Devices Security. *IEEE Internet of Things Journal*, *6*(1), 335-348.

[24] Kapito, B. (2020). *IoT$_{MAKA}$-Privacy Preserving Machine Authenticated Key Agreement for Internet of Things*. University of Malawi Master's Degree Thesis, Malawi.

[25] Burrows, M., Abadi, M. & Needham, R. (1990). A logic of authentication. *ACM Transactions on Computer Systems*, *8*(1), 18-36.

[26] Blanchet, B. (2001). An efficient cryptographic protocol verifier based on prolog rules. In *Proc. of the 14th IEEE Workshop on Computer Security Foundations*, 82-96.

[27] Dolev, D. & Yao, A. C. (1983). On the security of public key protocols. *IEEE Transactions on Information Theory*, *IT-29*(2), 198-208.

[28] Chandra, A. K. & Harel, D. (1985). Horn clause queries and generalizations. *The Journal of Logic Programming*, *2*(1), 1-15.

[29] Kim, H. (2014). Freshness-Preserving Non-Interactive Hierarchical Key Agreement Protocol over WHMS. *Sensors*, *14*, 23742-23747.

[30] Messerges, T. S., Dabbish, E. A. & Sloan, R. H. (2002). Examining smart-card security under the threat of power analysis attacks. *IEEE Transactions on Computers*, *51*(5), 541-552.

[31] Wu, F., Xu, L., Kumari, S., Li, X., Das, A. K., Khan, M. K., Karuppiah, M. & Baliyan, R. (2016). A novel and provably secure authentication and key agreement scheme with user anonymity for global mobility networks. *Secure Communication Networks*, *9*, 3527-3542.

[32] MIRACL, https://www.miracl.com. accessed at Jan. 20 (2024).