Hindawi
Security and Communication Networks
Volume 2018, Article ID 9046064, 14 pages
https://doi.org/10.1155/2018/9046064

*Research Article*

# Efficient and Secure Biometric-Based User Authenticated Key Agreement Scheme with Anonymity

**Dongwoo Kang**[1], **Jaewook Jung**[1], **Hyoungshick Kim**[1], **Youngsook Lee**[2] and **Dongho Won**[1]

[1]*Department of Computer Engineering, Sungkyunkwan University, 2066 Seoburo, Suwon, Gyeonggido 440-746, Republic of Korea*
[2]*Department of Cyber Security, Howon University, 64 Howondae 3-gil, Impi-myeon, Gunsan-si, Jeonrabuk-do 54058, Republic of Korea*

Correspondence should be addressed to Dongho Won; dhwon@security.re.kr

Received 4 May 2017; Revised 4 November 2017; Accepted 13 May 2018; Published 20 June 2018

Academic Editor: Vincenzo Conti



At present, a number of users employ an authentication protocol so as to enjoy protected electronic transactions in wireless networks. In order to establish an efficient and robust the transaction system, numerous researches have been conducted relating to authentication protocols. Recently, Kaul and Awasthi presented an user authentication and key agreement scheme, arguing that their scheme is able to resist various types of attacks and preserve diverse security properties. However, this scheme possesses critical vulnerabilities. First, the scheme cannot prevent two kinds of attacks, including off-line password guessing attacks and user impersonation attacks. Second, user anonymity rule cannot be upheld. Third, session key can be compromised by an attacker. Fourth, there is high possibility that the time synchronization trouble occurs. Therefore, we suggest an upgraded version of the user authenticated key agreement method that provides enhanced security. Our security and performance analysis shows that compared, to other associated protocols, our method not only improves the security level but also ensures efficiency.

## 1. Introduction

The rapid evolution of mobile devices and the development of Information and Communication Technology (ICT) are providing convenience to our lives. This development has particularly affected the computer science environment, which has adhered to not only conventional but also inefficient ways. While the users enjoy simplicity and efficiency in their transaction systems, the issue of security has emerged as a major interest in both academic and industrial fields. In order to guarantee reliability among the communication parties, authentication protocol supports security when users access to foreign network.

Lamport [1] first proposed an authentication mechanism and, since then, many related studies have been carried out [2–5] to enhance efficiency and security. In 2004, Das et al. [6] presented an authentication mechanism using dynamic identity technique in order to avoid exposure of user's identity. However, Wang et al. [7] claimed that their mechanism [6] cannot guarantee mutual authentication and fails to secure against server spoofing attack, and they then presented a new version. In 2010, Khan et al. [8] proved that Wang et al.'s version [7] is imperfect because their scheme leads to anonymity problem and server internal attack. Khan et al. [8] also presented an upgraded authentication method so as to treat Wang et al.'s deficiency. However, An [9] and Chou et al. [10] then separately pointed out that Khan et al.'s method [8] has an anonymity problem and unsteady under the various attacks such as off-line password guessing attack and forgery attack, with each proposing an improved new scheme. In Chou et al.'s research [10], they did not only demonstrate the deficiencies of Khan et al.'s method [8] but also criticized Song's scheme [11] that it cannot guarantee to protect off-line password guessing attack. In 2013, Chang et al. [12] corrected Wang et al.'s [7] flaw to expose private data including user's identity in the process of messages transmitted and suggested enhanced mechanism. However, Kumari et al. [13] claimed that Chang

et al.'s mechanism [12] cannot guarantee protecting against off-line password guessing attack, user disguise attack, and server masquerading attack, and their scheme also cannot keep user's identity and mutual authentication property. Like their predecessor, this was also followed by Kumari et al.'s [13] proposal for enhanced authentication technique.

Recently, Kaul and Awasthi [14] proved that Kumari et al.'s proposal [13] fails to protect important security parameters and session key shared between communication parties. With compensating these defections, they presented their own authentication method [14], claiming it can resist different types of attacks. However, we discovered that they compromise several security flaws. Their scheme (i) cannot withstand off-line password guessing attack and user impersonation attacks, (ii) is unable to support user anonymity, (iii) cannot achieve session key security, and (iv) encounters time synchronization trouble. In this research, we explain how the previously stated attacks operate and present a more developed version.

The remainder of this paper is arranged as follows: Section 2 introduces preliminary knowledges. Kaul and Awasthi's authentication mechanism is described in Section 3. Section 4 demonstrates that vulnerabilities of Kaul and Awasthi's mechanism. Our proposed method with detailed explanation is provided in Section 5. Sections 6 and 7 deal with informal security analysis and formal security analysis, respectively. In Section 8, we analyze the performance of the proposed scheme and, lastly, Section 9 contains the conclusion to this paper.

## 2. Preliminary Knowledge

In this section, we will describe basic knowledge in terms of threat model and introduce bio-hash function [15], which is used in our proposed scheme.

*2.1. Threat Model.* This subsection describes the threat model. Based on previous researches [5–14, 16–19], we constructed several common assumptions, including the capabilities of an attacker.

(1) All existing smart cards have vulnerabilities because confidential information stored within them can be extracted by physically monitoring the power consumption [20], meaning that an attacker can read and acquire data stored on the smart card.

(2) An attacker can control the public channels between the user and the server, meaning that the attacker can intercept any messages that are transmitted via the public channel [16–18].

(3) An attacker can modify and resend the intercepted/eavesdropped message [16].

*2.2. Biohashing.* A user's biometric data is very sensitive information. Thus, when user identification is employed using biometric data, secure and accurate matching is needed. To address this concern, Jin et al. [15] suggested fingerprint-based function to identify user's legitimacy in 2004. According to prior research [15], bio-hash technique employs particular tokenized pseudo-random numbers to each of users measuring biometric feature arbitrarily onto twofold strands. Bio-hash function $H(\cdot)$ is a one-way function with a feature that the probability of denial of service can be reduced. To date, many authentication studies have been carried out [19, 21–23] based on the bio-hash technique. In order to improve security, our proposed scheme also adopts user's biometric information applied bio-hash function, and the details are as follows in Section 5.

Table 1: Notations.

| Notations | Description |
|---|---|
| $U_i$ | The user |
| $S$ | The server |
| $ID_i$ | Identity of $U_i$ |
| $PW_i$ | Password of $U_i$ |
| $B_i$ | Biometric information of $U_i$ |
| $x$ | Secret key of $S$ |
| $y$ | Secret number generated by $S$ |
| $y_i$ | Random number of $i$th $U_i$ generated by $S$ |
| $r_1, r_2$ | Random number generated by $U_i$ |
| $T, T_s$ | Time-stamp values |
| $h(\cdot)$ | One-way hash function |
| $H(\cdot)$ | Bio-hash function |
| $X\|\|Y$ | Concatenate operation |
| $\oplus$ | XOR operation |

## 3. Review of the Kaul and Awasthi's Scheme

In this section, we briefly review the Kaul and Awasthi's scheme [14] to examine the cryptanalysis on their scheme. It consists of the following phases: registration, login, authentication, and password change. Figure 1 describes the Kaul and Awasthi's scheme, and Table 1 displays the notations employed in the remainder of this paper.

*3.1. Registration Phase*

(1) $U_i$ inputs $ID_i$ and $PW_i$ and then $U_i$ generates a random number $b$ that is only kept to user $U_i$. $U_i$ computes $RPW_i = h(PW_i \parallel b)$ and sends a registration request $\langle ID_i, RPW_i \rangle$ to $S$ through a secure channel.

(2) $S$ generates a random number $y_i$ and computes $\alpha_i = h((ID_i \oplus x) \parallel y)$, $\beta_i = \alpha_i \oplus h(ID_i \oplus RPW_i)$, $\gamma_i = y_i \oplus h(\alpha_i \oplus RPW_i)$ and $\chi_i = h(ID_i \parallel RPW_i \parallel y_i \parallel \alpha_i)$.

(3) $S$ then issues a smart card with the parameters $\{\beta_i, h(\cdot), \gamma_i, \chi_i\}$ and sends it to $U_i$ through a secure channel.

(4) Upon receiving the smart card, $U_i$ computes $\eta_i = b \oplus h(ID_i \oplus PW_i)$ and enters the $\eta_i$ in its memory, and, finally, the smart card includes the information $\{\beta_i, h(\cdot), \gamma_i, \chi_i, \eta_i\}$.

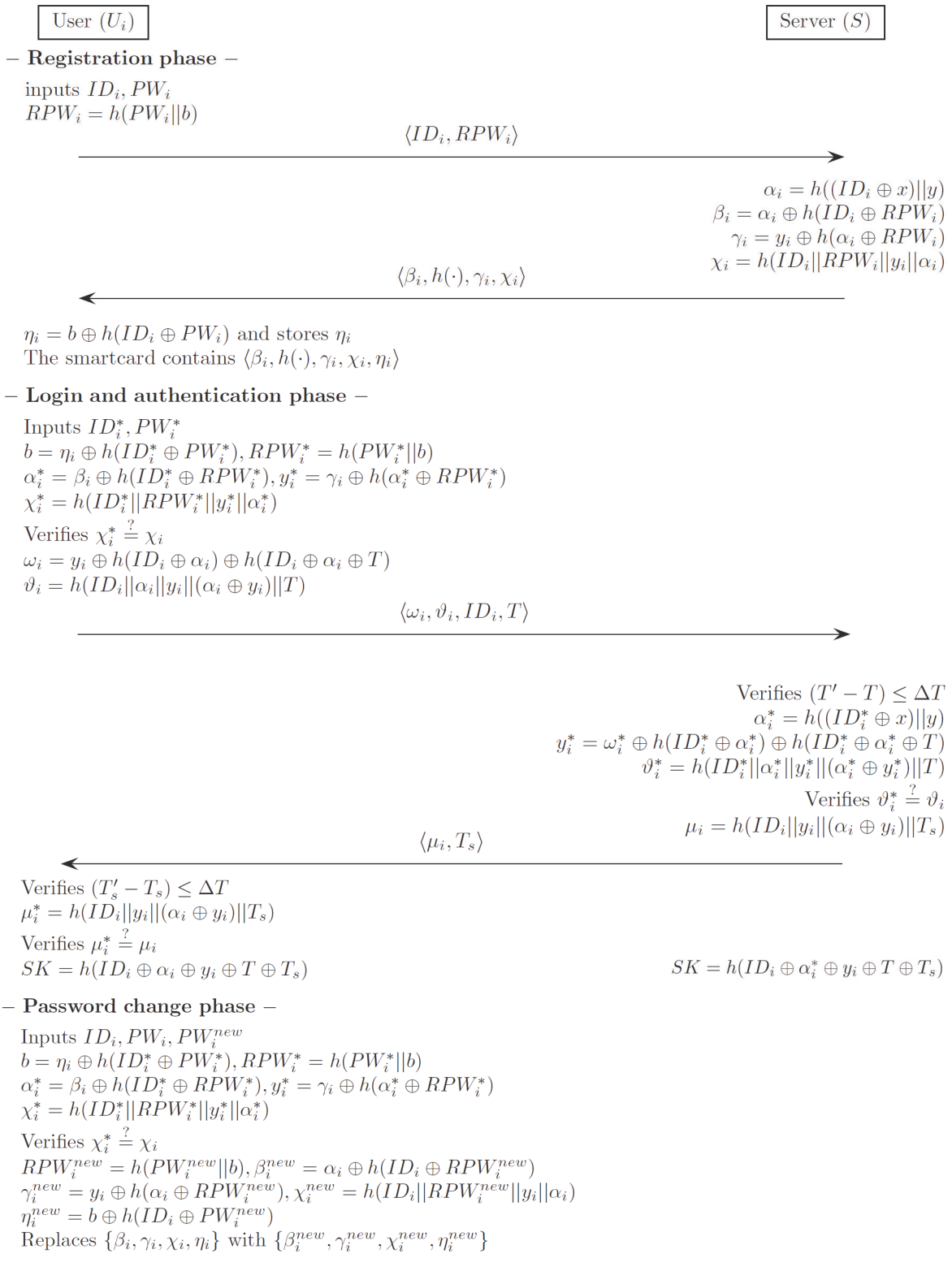


Figure 1: Kaul and Awasthi's scheme.

*3.2. Login Phase*

(1) $U_i$ inserts $U_i$'s smart card into a card reader and inputs his/her $ID_i^*$ and $PW_i^*$.

(2) Smart card computes $b = \eta_i \oplus h(ID_i^* \oplus PW_i^*)$, $RPW_i^* = h(PW_i^* \parallel b)$, $\alpha_i^* = \beta_i \oplus h(ID_i^* \oplus RPW_i^*)$, $y_i^* = \gamma_i \oplus h(\alpha_i^* \oplus RPW_i^*)$ and $\chi_i^* = h(ID_i^* \parallel RPW_i^* \parallel y_i^* \parallel \alpha_i^*)$. The smart card then compares $\chi_i^*$ with $\chi_i$. If this condition is satisfied, the smart card acknowledges the legitimacy of $U_i$ and proceeds with the next step. If not, this phase is terminated.

(3) Smart card computes $\omega_i = y_i \oplus h(ID_i \oplus \alpha_i) \oplus h(ID_i \oplus \alpha_i \oplus T)$ and $\vartheta_i = h(ID_i \parallel \alpha_i \parallel y_i \parallel (\alpha_i \oplus y_i) \parallel T)$.

(4) Finally, $U_i$ sends the login request $\langle \omega_i, \vartheta_i, ID_i, T \rangle$ to $S$ through a public network.

*3.3. Authentication Phase*

(1) $S$ verifies the time-stamp $T$ through $(T' - T) \leq \Delta T$. If it holds, $S$ proceeds with the next step. Otherwise, this phase is terminated.

(2) $S$ computes $\alpha_i^* = h((ID_i^* \oplus x) \parallel y)$, $y_i^* = \omega_i^* \oplus h(ID_i^* \oplus \alpha_i^*) \oplus h(ID_i^* \oplus \alpha_i^* \oplus T)$ and $\vartheta_i^* = h(ID_i^* \| \alpha_i^* \| y_i^* \| (\alpha_i^* \oplus y_i^*) \| T)$. $S$ then verifies whether $\vartheta_i^* = \vartheta_i$. If this comparison is satisfied, $S$ accepts the login request and proceeds with the next step. Otherwise, $S$ rejects the login request and this phase is terminated.

(3) $S$ computes $\mu_i = h(ID_i \parallel y_i \parallel (\alpha_i \oplus y_i) \parallel T_s)$ and sends an authentication request $\langle \mu_i, T_s \rangle$ to $U_i$ through a public network.

(4) $U_i$ verifies the time-stamp $T_s$ through $(T_s' - T_s) \leq \Delta T$. If it holds, $U_i$ proceeds with the next step. Otherwise, this phase is terminated.

(5) $U_i$ computes $\mu_i^* = h(ID_i \parallel y_i \parallel (\alpha_i \oplus y_i) \parallel T_s)$ and verifies whether $\mu_i^* = \mu_i$. If this comparison is satisfied, $U_i$ accepts the authentication request and proceeds with the next step. Otherwise, $U_i$ rejects the authentication request and this phase is terminated.

(6) Finally, $U_i$ computes a shared session key $SK = h(ID_i \oplus \alpha_i \oplus y_i \oplus T \oplus T_s)$ and $S$ also computes the same session key $SK$ successfully.

*3.4. Password Change Phase*

(1) $U_i$ inserts $U_i$'s smart card into a card reader and inputs $ID_i^*$, old password $PW_i^*$, and new password $PW_i^{new}$. The smart card computes $b = \eta_i \oplus h(ID_i^* \oplus PW_i^*)$ and $RPW_i^* = h(PW_i^* \parallel b)$.

(2) Smart card further computes $\alpha_i^* = \beta_i \oplus h(ID_i^* \oplus RPW_i^*)$, $y_i^* = \gamma_i \oplus h(\alpha_i^* \oplus RPW_i^*)$ and $\chi_i^* = h(ID_i^* \parallel RPW_i^* \parallel y_i^* \parallel \alpha_i^*)$. Smart card then verifies whether $\chi_i^* = \chi_i$. If this comparison is satisfied, smart card proceeds with the next step. Otherwise, this phase is terminated.

(3) Using the new password $PW_i^{new}$, smart card computes $RPW_i^{new} = h(PW_i^{new} \parallel b)$, $\beta_i^{new} = \alpha_i \oplus h(ID_i \oplus RPW_i^{new})$, $\gamma_i^{new} = y_i \oplus h(\alpha_i \oplus RPW_i^{new})$, $\chi_i^{new} = h(ID_i \parallel RPW_i^{new} \parallel y_i \parallel \alpha_i)$ and $\eta_i^{new} = b \oplus h(ID_i \oplus PW_i^{new})$.

(4) Smart card replaces $\{\beta_i, \gamma_i, \chi_i, \eta_i\}$ with the new parameters $\{\beta_i^{new}, \gamma_i^{new}, \chi_i^{new}, \eta_i^{new}\}$. Consequently, the smart card contains the information $\{\beta_i^{new}, h(\cdot), \gamma_i^{new}, \chi_i^{new}, \eta_i^{new}\}$.

## 4. Security Weaknesses of the Kaul and Awasthi's Scheme

In this section, we show that Kaul and Awasthi's scheme [14] possesses some security vulnerabilities. Based on the threat model as mentioned in Section 2.1, the following problems have been found and their detailed descriptions are given as follows.

*4.1. Lack of User's Anonymity.* In modern networks environments, user's sensitive information leakage such as identity or password can expedite an outside attacker to identify every specific user. In this case, user's privacy data is at risk of being exposed to a untrusted third party that disobey his/her will. Therefore, user anonymity is must be taken seriously as a satisfied property for user authentication scheme. However, in Kaul and Awasthi's scheme [14], an attacker can easily acquire the user's identity $ID_i$ through monitoring the public channels [16–18] because a user's identity $ID_i$ is transmitted in a plain text without any encryption during the login phase. Attacker also can abuse acquired user's identity to launch various types of attacks, leading to many malicious scenarios. For this reason, user anonymity cannot be preserved in Kaul and Awasthi's authentication scheme.

*4.2. Off-Line Password Guessing Attack.* This attack is the attempted identification of a password until the correct password is found due to the tendency of many users to create simple, brief passwords for the sake of convenience. For this reason, authentication schemes for all password-based users should be designed to prevent a guessing attack; however, Kaul and Awasthi's scheme has a weakness in this situation, and we therefore propose a scenario for an off-line password guessing attack. The following is a detailed description.

*Step 1.* After an attacker has stolen a smart card, the attacker can extract $\{\beta_i, h(\cdot), \gamma_i, \chi_i, \eta_i\}$ from the user's smart card.

*Step 2.* The attacker can use an eavesdropped login request $\langle \omega_i, \vartheta_i, ID_i, T \rangle$ from the public channel.

*Step 3.* The attacker selects a password candidate $PW_i^*$ and computes $RPW_i^* = h(PW_i^* \parallel b) = h(PW_i^* \parallel \eta_i \oplus h(ID_i \oplus PW_i^*))$.

*Step 4.* The attacker computes the following:

$$\chi_i^* = h(ID_i \parallel RPW_i^* \parallel y_i \parallel \alpha_i) = h(ID_i \parallel RPW_i^* \parallel \gamma_i \oplus h(\alpha_i \oplus RPW_i^*) \parallel \alpha_i) = h(ID_i \parallel RPW_i^* \parallel \gamma_i$$

$$\oplus h\left(\beta_i \oplus h\left(ID_i \oplus RPW_i^*\right) \oplus RPW_i^*\right) \parallel \beta_i \oplus h\left(ID_i \oplus RPW_i^*\right)) = h\left(ID_i \parallel h\left(PW_i^* \parallel \eta_i \oplus h\left(ID_i \oplus PW_i^*\right)\right) \parallel \gamma_i \oplus h\left(\beta_i \oplus h\left(ID_i \oplus h\left(PW_i^* \parallel \eta_i \oplus h\left(ID_i \oplus PW_i^*\right)\right)\right) \oplus h\left(PW_i^* \parallel \eta_i \oplus h\left(ID_i \oplus PW_i^*\right)\right)\right) \parallel \beta_i \oplus h\left(ID_i \oplus h\left(PW_i^* \parallel \eta_i \oplus h\left(ID_i \oplus PW_i^*\right)\right)\right)\right) \tag{1}$$

*Step 5.* The attacker iterates the comparison process until the computed result $\chi_i^*$ equals the extracted value $\chi_i$.

*Step 6.* If they corresponded with each other, $PW_i^*$ would be an accurate password.

Through the above description, we demonstrate that Kaul and Awasthi's scheme [14] does not guarantee to protect off-line password guessing attack.

*4.3. User Impersonation Attack.* Generally speaking, many password-based authentication schemes' security is based on knowledge of the password; therefore, if an attacker acquires an user's password, the attacker can impersonate a legitimate user. Unfortunately, Kaul and Awasthi's scheme has a weakness under this case. After obtaining the user's password $PW_i$, as described in Section 4.2, an attacker can successfully impersonate a legitimate user by performing the following steps.

*Step 1.* Attacker extracts $\{\beta_i, h(\cdot), \gamma_i, \chi_i, \eta_i\}$ after stolen smart card.

*Step 2.* The attacker gets $ID_i$ and $T$ in the eavesdropped request $\langle \omega_i, \vartheta_i, ID_i, T \rangle$.

*Step 3.* The attacker computes $b, RPW_i, \alpha_i, y_i$ using obtained $ID_i, PW_i, T$ and further computes $\omega_i, \vartheta_i$.

*Step 4.* The attacker constructs login request $\langle \omega_i, \vartheta_i, ID_i, T \rangle$ and sends it to $S$.

*Step 5.* Upon getting the login request, $S$ checks whether $T$ and $\vartheta_i$ are normal values or not.

*Step 6.* If the above checking process is done, $S$ assures that the received login request is a legal message.

In Step 6, it is obvious that $S$ can successfully verify $\vartheta_i$, since the values in attacker's login request are exactly equal to user's login request. Therefore, the attacker can successfully disguise a legitimate user in Kaul and Awasthi's scheme [14].

*4.4. Session Key Compromise.* In Kaul and Awasthi's scheme, if an attacker successfully guesses $U_i$'s password by off-line password guessing attack, the attacker can construct the session key $SK$ shared between user and server. First, the attacker can acquire $ID_i, T$ and $T_s$ after eavesdropping the login and authentication request. Then, the attacker can compute $\alpha_i$ and $y_i$ using obtained $PW_i$, which is previously compromised value through the Section 4.2. With combined these values, attacker can successfully establish $SK = h(ID_i \oplus \alpha_i \oplus y_i \oplus T \oplus T_s)$.

*4.5. Time Synchronize Problem.* It is time-stamp method that a hitherto commonly used method against replay attack. Kaul and Awasthi also mentioned a time-stamp method to prevent replay attack. However, this method may cause time synchronization problem between servers and users, since the current network system is large-scale wireless network composed of multitudinous users employing various devices contrary to past small scale network environment. Besides, it is inefficient to synchronize all system in real time. Nonce-based method applying random number is recommended instead of time-stamp method to settle synchronization problem [24]. Kaul and Awasthi's scheme is required to switch to nonce-based method to resolve time synchronization problem.

## 5. The Proposed Scheme

In this section, we suggest the refined version of authentication protocol to offer enhanced security by resolving Kaul and Awasthi's [14] vulnerabilities. In our proposed scheme, in order to conceal the user's identity, we employ dynamic identity technique that is combined form of $ID_i$ and random number. We also use biometrics information with Biohashing $H(\cdot)$ [15] to avoid off-line password guessing attack and impersonation attack. In addition, we apply nonce-based method to prevent replay attack instead of unsteady time-stamp method. Our proposed method also consists of four phases: registration, login, authentication, and password change. Figure 2 describes our proposed scheme, and the notations employed to the proposed scheme are displayed in Table 1.

*5.1. Registration Phase*

(1) $U_i$ inputs $ID_i$ and $PW_i$ and imprints his/her biometrics $B_i$. Then $U_i$ computes $RPW_i = h(PW_i \parallel H(B_i))$ and sends a registration request $\langle ID_i, RPW_i \rangle$ to server $S$ through a secure channel.

(2) $S$ computes $\alpha_i = h(ID_i \oplus x)$, $\beta_i = \alpha_i \oplus h(ID_i \parallel RPW_i)$, and $\gamma_i = h(\alpha_i \parallel RPW_i) \oplus x$.

(3) $S$ then issues a smart card with the parameters $\{\beta_i, h(\cdot), \gamma_i\}$ and sends it to $U_i$ through a secure channel.

(4) Upon receiving the smart card, $U_i$ computes $\eta_i = h(ID_i \parallel PW_i \parallel H(B_i))$ and enters the $\eta_i$ in its memory, and, finally, the smart card includes the information $\{\beta_i, h(\cdot), \gamma_i, \eta_i\}$.

*5.2. Login Phase*

(1) $U_i$ inserts $U_i$'s smart card into a card reader and inputs $ID_i$, $PW_i$ and also imprints biometric $B_i$. Smart card then computes $\eta_i^* = h(ID_i \parallel PW_i \parallel H(B_i))$ and compares it with the stored $\eta_i$ in the smart card. If this comparison is satisfied, the smart card acknowledges

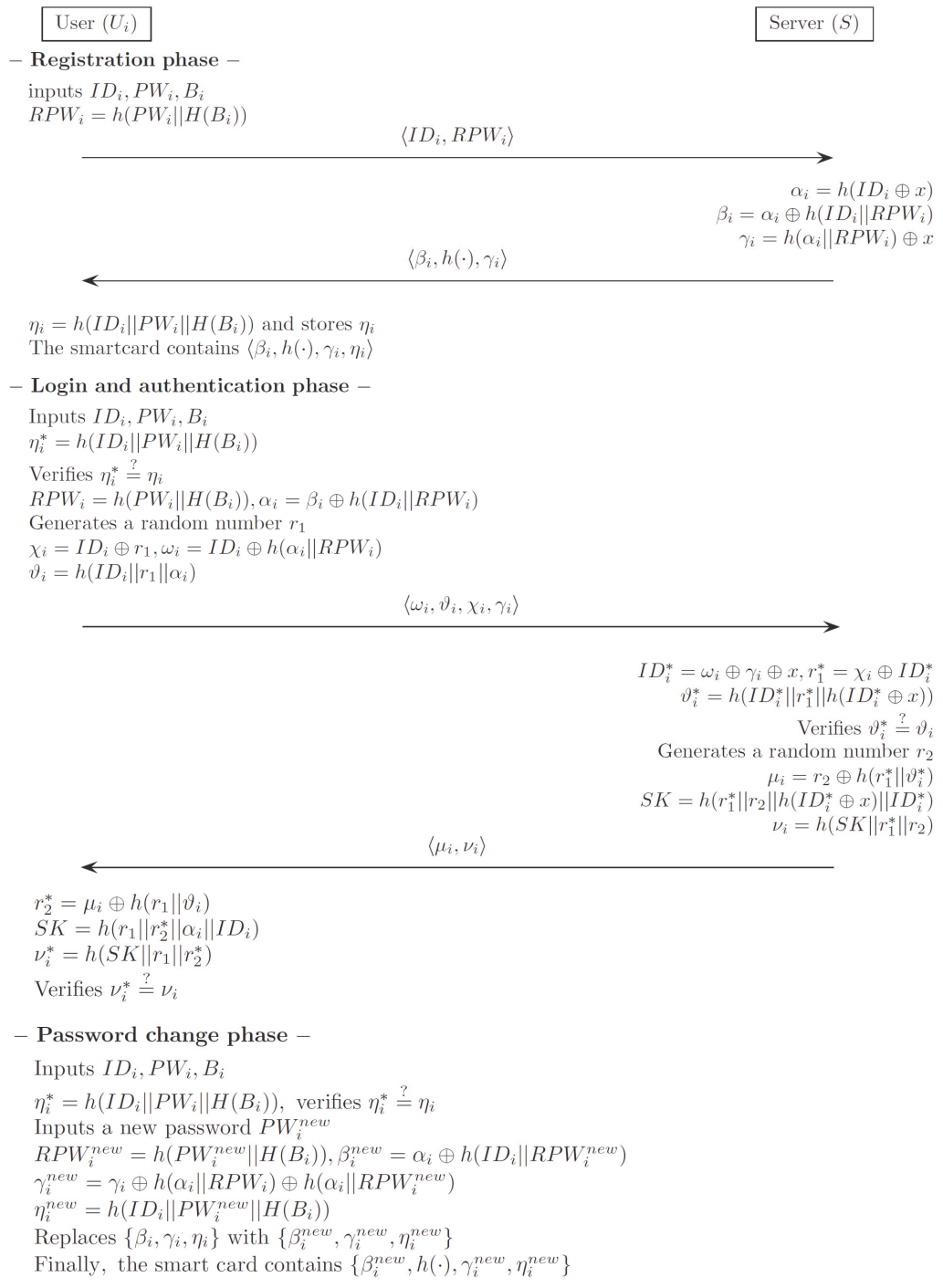


Figure 2: Our proposed scheme.

the legitimacy of the $U_i$ and proceeds with the next step. Otherwise, it terminates this phase.

(2) Smart card computes $RPW_i = h(PW_i \parallel H(B_i))$ and $\alpha_i = \beta_i \oplus h(ID_i \parallel RPW_i)$.

(3) Smart card selects a random number $r_1$ and computes $\chi_i = ID_i \oplus r_1, \omega_i = ID_i \oplus h(\alpha_i \parallel RPW_i)$ and $\vartheta_i = h(ID_i \parallel r_1 \parallel \alpha_i)$.

(4) Finally, $U_i$ sends the login request $\langle \omega_i, \vartheta_i, \chi_i, \gamma_i \rangle$ to $S$ through a public network.

### 5.3. Authentication Phase

(1) $S$ computes $ID_i^* = \omega_i \oplus \gamma_i \oplus x, r_1^* = \chi_i \oplus ID_i^*$ and $\vartheta_i^* = h(ID_i^* \parallel r_1^* \parallel h(ID_i^* \oplus x))$. $S$ then verifies whether $\vartheta_i^* = \vartheta_i$. If this comparison is satisfied, $S$ accepts the login

Table 2: Security comparison.

| Security attributes/Schemes | An [9] | Chou et al. [10] | Chang et al. [12] | Kumari et al. [13] | Kaul & Awasthi's [14] | Proposed scheme |
|---|---|---|---|---|---|---|
| User anonymity | × | × | × | × | × | √ |
| Privileged insider attack | √ | × | × | √ | √ | √ |
| Replay attack | √ | √ | √ | √ | √ | √ |
| Off-line password guessing attack | × | × | × | √ | × | √ |
| User impersonation attack | × | × | × | × | × | √ |
| Mutual authentication | × | × | × | × | √ | √ |
| Session key Compromise | × | × | $N/A$ | × | × | √ |
| Password verification process | × | × | × | √ | √ | √ |
| Convenient password change | $N/A$ | √ | × | √ | √ | √ |
| No time synchronization | × | × | × | × | × | √ |

request and proceeds with the next step. Otherwise, $S$ rejects the login request and this phase is terminated.

(2) $S$ selects a random number $r_2$ and computes $\mu_i = r_2 \oplus h(r_1^* \parallel \vartheta_i^*)$, $SK = h(r_1^* \parallel r_2 \parallel h(ID_i^* \oplus x) \parallel ID_i^*)$ and $\nu_i = h(SK \parallel r_1^* \parallel r_2)$.

(3) $S$ sends an authentication request $\langle \mu_i, \nu_i \rangle$ to $U_i$ through a public network.

(4) $U_i$ computes $r_2^* = \mu_i \oplus h(r_1 \parallel \vartheta_i)$, $SK = h(r_1 \parallel r_2^* \parallel \alpha_i \parallel ID_i)$ and $\nu_i^* = h(SK \parallel r_1 \parallel r_2^*)$.

(5) $U_i$ then verifies whether $\nu_i^* = \nu_i$. If this comparison is satisfied, $U_i$ accepts the authentication request and successfully authenticates $S$. Otherwise, $U_i$ rejects the authentication request and this phase is terminated.

*5.4. Password Change Phase*

(1) $U_i$ inserts $U_i$'s smart card into a card reader and inputs $ID_i$, $PW_i$ and also imprints biometric $B_i$. Smart card then computes $\eta_i^* = h(ID_i \parallel PW_i \parallel H(B_i))$ and compares it with the stored $\eta_i$ in the smart card. If this comparison is satisfied, the smart card acknowledges the legitimacy of $U_i$ and proceeds with the next step. Otherwise, it terminates this phase.

(2) $U_i$ inputs a new password $PW_i^{new}$, and smart card computes $RPW_i^{new} = h(PW_i^{new} \parallel H(B_i))$.

(3) Smart card further computes

$$\begin{aligned} \beta_i^{new} &= \alpha_i \oplus h\left(ID_i \parallel RPW_i^{new}\right) \\ \gamma_i^{new} &= \gamma_i \oplus h\left(\alpha_i \parallel RPW_i\right) \oplus h\left(\alpha_i \parallel RPW_i^{new}\right) \\ \eta_i^{new} &= h\left(ID_i \parallel PW_i^{new} \parallel H\left(B_i\right)\right) \end{aligned} \quad (2)$$

(4) Smart card replaces $\{\beta_i, \gamma_i, \eta_i\}$ with the new parameters $\{\beta_i^{new}, \gamma_i^{new}, \eta_i^{new}\}$. Consequently, the smart card contains the information $\{\beta_i^{new}, h(\cdot), \gamma_i^{new}, \eta_i^{new}\}$.

## 6. Security Analysis and Proof of the Proposed Scheme

In this section, we first analyze whether our proposed technique satisfies numerous security requirements. After that, we will apply Burrows-Abadi-Needham (BAN) logic [25] to validate that the generated session key is precisely distributed to user $U_i$ and server $S$.

*6.1. Security Analysis of Proposed Scheme.* We evaluate whether our proposal is secure against various attacks and satisfies various authentication requirements. In addition, comparative analysis of related schemes [9, 10, 12–14] is carried out, and the results are shown in Table 2.

*6.1.1. User Anonymity.* Our scheme protects the user's identity $ID_i$ sent by messages from the possible dangers of exposure in order to accomplish user anonymity. Even if an attacker captures $\chi_i$ by snatching login request $\langle \omega_i, \vartheta_i, \chi_i, \gamma_i \rangle$, it is unachievable to derive $ID_i$ since the attacker cannot acquire random number $r_1$.

*6.1.2. Privileged Insider Attack.* In our scheme, when $U_i$ sends a registration request $\langle ID_i, RPW_i \rangle$ to $S$, $PW_i$ is transmitted not as uncovered, but as a form of $RPW_i = h(PW_i \parallel H(B_i))$ with a value $H(B_i)$,to preclude insider attack. Thus, our scheme guarantees to hinder an insider attack.

*6.1.3. Replay Attack.* Assumes that an attacker steals the former login request $\langle \omega_i, \vartheta_i, \chi_i, \gamma_i \rangle$. Then, the attacker might try to pose as a valid user by sending this request in order to login the server. However, if the attacker sends prior login request, the server would apparently turn down the request because our scheme can find out the invalid random number through the comparison of $\vartheta_i$ value. In addition, in each session, our proposed scheme handles distinct random numbers. As a result, our scheme can provide safety against replay attack.

*6.1.4. Off-Line Password Guessing Attack.* From the stolen smart card and snatch the login request $\langle \omega_i, \vartheta_i, \chi_i, \gamma_i \rangle$, an

attacker can procure $\{\beta_i, h(\cdot), H(\cdot), \gamma_i, \eta_i\}$, and, using these values, the attacker may try to predict the precise password $PW_i$. However, the attacker cannot conjecture the $PW_i$ unless $ID_i$ and $H(B_i)$ are given. Furthermore, an user $U_i$ only knows $H(B_i)$ since it is a hashed biometric information. For this reason, our scheme guarantees to defend off-line password guessing attack.

*6.1.5. User Impersonation Attack.* In our scheme, an attacker should generate the values of $\omega_i$, $\vartheta_i$, $\chi_i$, and $\gamma_i$ after obtaining the value of $ID_i$ or a random number, $r_1$ to achieve impersonation attack. Still, as we mentioned above, it is impossible for an attacker to get the value of $ID_i$ or $r_1$. Thus, an attacker cannot create an appropriate login request to cheat $S$.

*6.1.6. Mutual Authentication.* In the authentication phase of our scheme, $U_i$ and $S$ can attest each other according to several procedures. To be specific, $S$ first confirms the login request by examining whether $\vartheta_i$ is accurate. $U_i$ also makes sure the authentication request by checking whether $\nu_i$ is correct. If all these verification processes are executed successfully, mutual authentication has succeeded properly.

*6.1.7. Session Key Compromise.* In our scheme, in order to compromise the session key $SK = h(r_1 \parallel r_2^* \parallel \alpha_i \parallel ID_i)$, an attacker should have the random numbers $r_1$ and $r_2$. Moreover, to acquire the random numbers, the attacker should know $ID_i$ in advance. However, the attacker has no way to derive the user's $ID_i$, all things considered. In this manner, our authentication mechanism guarantees session key security.

*6.1.8. Password Verification Process.* There is a feasibility that a user accidentally inputs an inaccurate password, but, for password verification procedure, a server will detect the wrong password after executing authentication phase [19]. Considering this kind of inefficient situation, our scheme evaluates the correctness of password by checking the value $\eta_i$ in an early login phase.

*6.1.9. Convenient Password Change.* In general, it is encouraged to implement verification process by itself when password adjustment occurs [24]. The performance of a security scheme can be enhanced through its own mechanism without communicating to server $S$. Our proposed scheme carries out extant password checking in self-verification process within smart card. After testing, calculated values $\{\beta_i^{new}, \gamma_i^{new}, \eta_i^{new}\}$ from new password will substitute the existed values in an efficient and appropriate way.

*6.1.10. No Time Synchronization.* In timestamp-based authentication protocols, when a user and a server transmit a packet, the clocks of all devices should be set accurately. For this reason, there is a strong likelihood that error occurs. On the contrary, our scheme handles random numbers $r_1$ and $r_2$ rather than time-stamp technique to avert this problem.

*6.2. Authentication Proof with BAN Logic.* We use Burrows-Abadi-Needham (BAN) logic [25] to demonstrate that the user and server participating in communication are each correctly assigned the session key $SK$. The basic symbols of BAN logic are as follows:

(i) $(F)_K$: formula $F$ is hashed with $K$.

(ii) $U \triangleleft F$: $U$ perceives formula $F$.

(iii) $< F >_K$: combine formula $F$ using $K$.

(iv) $\sharp(F)$: formula $F$ is fresh.

(v) $U \mid\sim F$: $U$ said formula $F$.

(vi) $U \mid\equiv F$: $U$ trusts formula $F$.

(vii) $U \Rightarrow F$: $U$ can manage formula $F$.

(viii) $U \overset{K}{\longleftrightarrow} S$: $U$ and $S$ assign a secret key $K$.

The ban logic also provides the following basic rules:

(1) Message meaning rule: $(U \mid\equiv U \overset{K}{\longleftrightarrow} S, U \triangleleft < F >_K)/(U \mid\equiv S \mid\sim F)$.

(2) Nonce verification rule: $(U \mid\equiv \#(F), U \mid\equiv S \mid\sim F)/(A \mid\equiv S \mid\equiv F)$.

(3) The believe rule: $(U \mid\equiv F, U \mid\equiv G)/(U \mid\equiv (F, G))$.

(4) Freshness conjuncatenation rule: $(A \mid\equiv \#(F))/(A \mid\equiv \#(F, G))$.

(5) Jurisdiction rule: $(U \mid\equiv S \mid\Rightarrow F, U \mid\equiv S \mid\equiv F)/(U \mid\equiv F)$.

Using ban logic, we will accomplish the following goals:

(i) Goal 1: $U \mid\equiv (U \overset{SK}{\longleftrightarrow} S)$.

(ii) Goal 2: $S \mid\equiv (U \overset{SK}{\longleftrightarrow} S)$.

The login and authentication messages used in our scheme can be translated into an ideal form shown as follows:

(i) Message 1: $U \rightarrow S$: $\langle ID_i \rangle_x, r_1, (ID_i, r_1, B_i)_x$.

(ii) Message 2: $S \rightarrow U$: $r_2, (ID_i, r_1, r_2, SK)_x$.

To proceed with the proof, we have defined the following assumptions:

(i) A1: $U \mid\equiv \sharp(r_1)$

(ii) A2: $S \mid\equiv \sharp(r_2)$

(iii) A3: $U \mid\equiv (U \overset{x}{\longleftrightarrow} S)$

(iv) A4: $S \mid\equiv (U \overset{x}{\longleftrightarrow} S)$

(v) A5: $U \mid\equiv S \mid\equiv (U \overset{x}{\longleftrightarrow} S)$

(vi) A6: $S \mid\equiv U \mid\equiv (U \overset{x}{\longleftrightarrow} S)$

Our verification procedure is as follows:

Based on Message 1, we could derive:

(i) S1: $S \triangleleft (ID_i, r_1, B_i)_x, r_1$

Based on assumption A4, we adapt the message meaning rule to derive:

(ii) S2: $S \mid\equiv U \mid\sim r_1$

Based on assumption A1 and the freshness conjuncatenation rule, we derive:

(iii) S3: $S \mid\equiv \sharp(ID_i, r_1, B_i)_x$

Based on S2, S3 and the nonce verification rule, we derive:

(iv) S4: $S \mid\equiv U \mid\equiv (ID_i, r_1, B_i)_x$

Based on A4, S4 and the jurisdiction rule, we derive:

(v) S5: $S \mid\equiv r_1$

Based on V5, assumption A2, and SK, we derive:

(vi) S6. $S \mid\equiv (U \overset{SK}{\longleftrightarrow} S)$ **(Goal 2.)**

Based on Message 2, we could derive:

(vii) S7: $U \triangleleft r_2, (ID_i, r_1, r_2, SK)_x$

Based on assumption A3, we adapt the message meaning rule to derive:

(viii) S8: $U \mid\equiv S \mid\sim r_2$

Based on assumption A2 and the freshness conjuncatenation rule, we derive:

(ix) S9: $U \mid\equiv \sharp(ID_i, r_1, r_2, SK)_x$

Based on S8, S9 and the nonce verification rule, we derive:

(x) S10: $U \mid\equiv S \mid\equiv (ID_i, r_1, r_2, SK)_x$

Based on A3, S10 and the jurisdiction rule, we derive:

(xi) S11: $U \mid\equiv r_2$

Based on V11, assumption A1, and SK, we derive:

(xii) S12. $U \mid\equiv (U \overset{SK}{\longleftrightarrow} S)$ **(Goal 1.)**

Based on (Goal.1) and (Goal.2), we can assure that our proposed scheme provides the mutual authentication and agreement of the session key $SK$, which is correctly distributed between $U_i$ and $S$.

## 7. Formal Security Proof Using AVISPA Tool

In this section, we demonstrate that our proposed scheme can resist both passive and active attacks by simulating its use in the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool [26].

*7.1. Overview of AVISPA Tool.* The AVISPA is a formal tool that is generally used to verify protocol security. The protocol specification is written in High Level Protocols Specification Language (HLPSL) [27] and is translated into the Intermediate Format (IF) by a Translator HLPSL2IF. The result is then treated as the input value of the different back-end procedures. In this paper, we first stipulate our authentication mechanism based on HLPSL and then derive the results of the simulation using two back-ends OFMC and CL-AtSe.

*7.2. Stipulating the Proposed Scheme.* This section supports descriptions of the specifications of our proposed scheme in HLPSL. We have assigned the fundamental roles for a user $U_i$ and a server $S$ for each phase, and we then stipulate the other roles for the session, environment, and goal. Box 1 shows that the role specification of the user $U_i$ for our proposed scheme.

During the registration phase, $U_i$ sends $\langle ID_i, RPW_i \rangle$ to $S$ through a secure channel using the $Snd()$ operation and a symmetric key $SKuisj$. The type declaration $channel(dy)$ expressed that the channel is effected by the Dolev-Yao threat model [28]. Under this threat model, the attacker can, some malignancy action such as intercept, eavesdrop on any messages by agents. The declaration $secret(\{IDi, PWi, BIOi\}, sub2, Ui)$ expressed that the personal private information $(ID_i, PW_i, Bio_i)$ is only known to $U_i$. $new()$ operation is used to generate a random number $r_1$ during the login and authentication phase. $U_i$ then computes $RPW_i = h(PW_i \parallel H(B_i))$, $\alpha_i = \beta_i \oplus h(ID_i \parallel RPW_i)$, $\chi_i = ID_i \oplus r_1$, $\omega_i = ID_i \oplus h(\alpha_i \parallel RPW_i)$, and $\vartheta_i = h(ID_i \parallel r_1 \parallel \alpha_i)$ and sends the login request $\langle \omega_i, \vartheta_i, \chi_i, \gamma_i \rangle$ to $S$ via a public network. The declaration $witness(Ui, Sj, user\_server\_r1, R1')$ expresses that $U_i$ has recently generated a random number $r_1$ for $S$ to communicate. $U_i$ finally receives the authentication message $\langle \mu_i, \nu_i \rangle$ from $S$ via a public channel.

The role specification of the server $S$ for our scheme is shown in Box 2. In the registration phase, $S$ receives the registration request message $\langle ID_i, RPW_i \rangle$ from $U_i$. After receiving the message, $S$ issues a smart card which contains parameters $\{\beta_i, h(\cdot), \gamma_i\}$ and sends it to $U_i$ using $Snd()$ operation and symmetric key $SKuisj$. During the login and authentication phase, $S$ receives the login request message $\langle \omega_i, \vartheta_i, \chi_i, \gamma_i \rangle$ from $U_i$. Then, similar to user $U_i$'s role, Server $S$ uses $new()$ operation to generate a random number $r_2$. The declaration $witness(Sj, Ui, server\_user\_r2, R2')$ expresses that $S$ has recently generated a random number $r_2$ for $U_i$ to communicate. The declaration $request(Ui, Sj, user\_server\_r1, R1')$ expresses that $S$ authenticates user $U_i$.

The rest segment in HLPSL, session, environment, and goal is described in Box 3. Each segment's specifics are as follows:

(i) Session segment: the participants in the communication, including the user and server, and other basic roles being instanced as concrete arguments

(ii) Environment segment: cover the global constant, session composition, and intruder knowledge

(iii) Goal segment: secrecy goals and authentication goals.

*7.3. Simulation Results.* This section describes the output results of the simulation conducted for our proposed scheme. The results of the simulation under the OFMC and CL-AtSe back-ends are shown in Box 4, which clearly shows that our scheme is SAFE under each back-end. Therefore, it is obvious that our proposed scheme can prevent passive and active attacks, including replay and man-in-the-middle attacks.

```
role user (Ui,Sj: agent,
           SKuisj: symmetric_key,
           %H : one-way/bio hash function
           H : hash_func,
           Snd,Rcv: channel(dy))
played_by Ui
def=
  local State : nat,
        IDi, PWi, RPWi, BIOi, X, Ai, Bi, Ci, R1, Di, Ei, Fi, R2, Gi, Vi : text
  const user_server_r1, server_user_r2,
        subs1, subs2 : protocol_id
  init  State := 0
  transition
%%%%% Registration phase
  1. State = 0 ∧ Rcv(start) =|>
     State' := 1 ∧ RPWi' := H(PWi.H(BIOi))
     ∧Snd({IDi.RPWi'}_SKuisj)
     ∧secret({X}, subs1, Sj)
     ∧secret({IDi, PWi, BIOi}, subs2, Ui)
  2. State = 1 ∧ Rcv({xor(H(xor(IDi, X)), H(IDi.H(PWi.H(BIOi)))).xor(H(xor(IDi, X).H(PWi.H(BIOi))), X)}_SKuisj) =|>
%%%%% Login phase
     State' := 2 ∧ R1' := new()
     ∧ RPWi' := H(PWi.H(BIOi))
     ∧ Ai' := xor(xor(H(xor(IDi, X)), H(IDi.H(PWi.H(BIOi)))), H(IDi.RPWi'))
     ∧ Ci' := xor(H(Ai'.H(PWi.H(BIOi))), X)
     ∧ Di' := xor(IDi, R1')
     ∧ Ei' := xor(IDi, H(Ai'.RPWi'))
     ∧ Fi' := H(IDi.R1'.Ai')
     ∧ Snd(Ci'.Di'.Ei'.Fi')
     ∧ witness(Ui, Sj, user_server_r1, R1')
%%%%% Verification phase
  3. State = 2 ∧ Rcv(xor(R2', H(R1'.H(IDi.R1'.Ai'))).H(H(R1'.R2'.H(xor(IDi, X)).IDi).R1'.R2')) =|>
     State' := 3 ∧ request(Sj, Ui, server_user_r2, R2')
end role
```

Box 1: Role specification in HLPSL for the user $U_i$.

## 8. Performance Analysis of the Proposed Scheme

In this section, we compare the execution time and cost of computation for our proposed technique with the technique [9, 10, 12–14]. In general, cost of computation in authentication protocol is analyzed by operations performed in each work within the protocols. Accordingly, the cost of computational analysis focuses on the activities performed by members, like users and servers. The following is a definition of the parameters for evaluation of the calculation cost:

(i) $T_H$ is the time of executing a one-way hashing function.

(ii) $T_\oplus$ is the time of executing a XOR operation.

(iii) $T_M$ is the time of executing a modular exponentiation.

Table 3 contains a summary of the calculation overhead comparison. The results show that An [9], Chou et al. [10], Chang et al. [12], Kumari et al. [13], Kaul and Awasthi [14], and our technique require the total computation cost overheads $12T_H + 14T_\oplus + 4T_M$, $36T_H + 31T_\oplus$, $24T_H + 14T_\oplus$, $21T_H + 24T_\oplus$, $28T_H + 36T_\oplus$, and $27T_H + 15T_\oplus$, respectively. On the basis of [29], we adopt that the actual execution time for the complexity notations is as follows: $T_H$=0.0005s and $T_M$=0.522s. Since the XOR operation time is short, $T_\oplus$ does not need to be considered. As shown in Table 3, we observed that the execution time of our proposed scheme requires only 0.0135s ($\approx 27 \times 0.0005$s), so it can be regarded as a negligible significance, whereas the execution time of An's scheme [9] using modular exponentiation operation required 2.094s ($\approx 12 \times 0.0005\text{s} + 4 \times 0.522\text{s}$), so this scheme turned out to be ineffective. Thus, we conclude that our proposed technology considers efficiency.

```
role server (Ui,Sj: agent,
             SKuisj: symmetric_key,
             H : hash_func,
             Snd,Rcv: channel(dy))
played_by Sj
def=
  local State : nat,
             IDi, PWi, RPWi, BIOi, X, Ai, Bi, Ci, R1, Di, Ei, Fi, R2, Gi, Vi : text
  const user_server_r1, server_user_r2,
             subs1, subs2 : protocol_id
  init   State := 0
  transition
%%%%% Registration phase
  1. State = 0 ∧ Rcv({ IDi.H(PWi.H(BIOi)) }_SKuisj) =|>
     State' := 1 ∧ secret({X}, subs1, Sj)
       ∧ secret({IDi, PWi, BIOi}, subs2, Ui)
       ∧ Snd({ xor(H(xor(IDi, X)), H(IDi.H(PWi.H(BIOi)))).xor(H(xor(IDi, X).H(PWi.H(BIOi))), X) }_SKuisj)
%%%%% Login phase
  2. State = 1 ∧ Rcv(xor(H(H(xor(IDi, X)).H(PWi.H(BIOi))), X).xor(IDi, R1').xor(IDi, H(H(xor(IDi,
X)).RPWi')).H(IDi.R1'.H(xor(IDi, X)))) =|>
%%%%% Verification phase
       State' := 2 ∧ R2' := new()
       ∧ IDi' := xor(xor(IDi, H(H(xor(IDi, X)).RPWi')), Ci', X)
       ∧ R1' := xor(xor(IDi, R1'), IDi)
       ∧ Fi' := H(IDi.R1'.H(xor(IDi, X)))
       ∧ Gi' := xor(R2', H(R1'.H(IDi.R1'.Ai')))
       ∧ Vi' := H(H(R1'.R2'.H(xor(IDi, X)).IDi).R1'.R2')
       ∧ Snd(Gi'.Vi')
       ∧ witness(Sj, Ui, server_user_r2, R2')
       ∧ request(Ui, Sj, user_server_r1, R1')
end role
```

Box 2: Role specification in HLPSL for the server $S$.

Table 3: Performance comparison.

| Phases/Schemes | An [9] | Chou et al. [10] | Chang et al. [12] | Kumari et al. [13] | Kaul & Awasthi [14] | Proposed scheme |
|---|---|---|---|---|---|---|
| Registration phase | $3T_H+2T_\oplus$ | $7T_H+5T_\oplus$ | $2T_H+1T_\oplus$ | $4T_H+5T_\oplus$ | $6T_H+3T_\oplus$ | $6T_H+3T_\oplus$ |
| Login phase | $2T_H+4T_\oplus$ | $5T_H+3T_\oplus$ | $4T_H+3T_\oplus$ | $5T_H+10T_\oplus$ | $8T_H+12T_\oplus$ | $6T_H+3T_\oplus$ |
| Authentication phase | $7T_H+8T_\oplus+4T_M$ | $17T_H+16T_\oplus$ | $6T_H+2T_\oplus$ | $7T_H+3T_\oplus$ | $6T_H+9T_\oplus$ | $8T_H+6T_\oplus$ |
| PW change phase | – | $7T_H+6T_\oplus$ | $12T_H+8T_\oplus$ | $5T_H+6T_\oplus$ | $8T_H+12T_\oplus$ | $7T_H+3T_\oplus$ |
| Total cost | $12T_H+14T_\oplus+4T_M$ | $36T_H+31T_\oplus$ | $24T_H+14T_\oplus$ | $21T_H+24T_\oplus$ | $28T_H+36T_\oplus$ | $27T_H+15T_\oplus$ |
| Execution time | 2.094 s | 0.018 s | 0.012 s | 0.0105 s | 0.014 s | 0.0135 s |
| Memory capacity | 640 bits | 896 bits | 384 bits | 768 bits | 640 bits | 512 bits |
| Exchange cost | 896 bits | 1080 bits | 768 bits | 896 bits | 768 bits | 768 bits |

```
role session(Ui, Sj: agent,
             SKuisj : symmetric_key,
             H : hash_func)
def=
local SI, SJ, RI, RJ: channel (dy)
  composition
             user(Ui, Sj, SKuisj, H, SI, RI)
             ∧ server(Ui, Sj, SKuisj, H, SJ, RJ)
end role

goal
  secrecy_of subs1
  secrecy_of subs2
  authentication_on user_server_r1
  authentication_on server_user_r2
end goal

environment()
role environment()
def=
    const ui, sj: agent,
             skuisj : symmetric_key,
             h : hash_func,
             %H : one-way/bio hash function
             idi, pwi, bioi, x, r1, r2: text,
             user_server_r1, server_user_r2, subs1, subs2 :
protocol_id
    intruder_knowledge = {ui, sj, h}
    composition
              session(ui, sj, skuisj, h)
              ∧ session(ui, sj, skuisj, h)
end role
```

Box 3: Role specification in HLPSL for the session, environment, and goal.

We also analyze the memory capacity of smart card and estimate the message exchange cost. Based on [13], we assume that output length of all values, such as $ID_i$, $PW_i$, $T, T_s$, $h(\cdot), H(\cdot)$, and random numbers, is 128 bits long. As shown in Table 3, in Kaul ans Awasthi's scheme [14], the memory capacity of smart card $\{\beta_i, h(\cdot), \gamma_i, \chi_i, \eta_i\}$ requires (5 × 128) = 640 bits. The communication message consists of two packets: login request $\langle \omega_i, \vartheta_i, ID_i, T \rangle$ and authentication request $\langle \mu_i, T_s \rangle$. Thus, message exchange cost of Kaul and Awasthi's scheme is (6 × 128) = 768 bits. In our proposed scheme, the smart card $\{\beta_i, h(\cdot), H(\cdot), \gamma_i, \eta_i\}$ requires (5 × 128) = 640 bits, and the messages composed of $\langle \omega_i, \vartheta_i, \chi_i, \gamma_i \rangle$ and $\langle \mu_i, \nu_i \rangle$ require (6 × 128) = 768 bits. In conclusion, the results show that the memory capacity and communication overhead of our proposed scheme are relatively better as those of the other schemes [9, 10, 13]. In addition, the proposed technology can defend against a variety of existing attacks. This is shown in Table 2.

## 9. Conclusions

In this paper, we demonstrate that Kaul and Awasthi's scheme includes some critical vulnerabilities, and we propose an extended version to overcome these defects. Our proposed scheme has been thoroughly estimated in terms of a variety of security features. In addition, the performance comparison for the proposed scheme in relation to the other studies has been analyzed, and we conclude that the proposed scheme properly considers the efficiency and robustness.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Authors' Contributions

Dongwoo Kang developed the idea of the proposed authentication scheme and carried out the security analysis. Jaewook Jung and Youngsook Lee conducted the performance analysis of the proposed idea and wrote the manuscript under the supervision of Professor Dongho Won and Hyoungshick Kim.

## Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
/home/jaewook/Desktop/span/testsuite/results/Security&com
munications.if
GOAL as specified
BACKEND OFMC

STATISTICS
  TIME 24 ms
  parseTime 0 ms
  visitedNodes: 22 nodes
  depth: 3 plies

 SUMMARY
  SAFE
 DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
 PROTOCOL
/home/jaewook/Desktop/span/testsuite/results/Security&com
munications.if
 GOAL
  As specified
 BACKEND
  CL-AtSe
 STATISTICS
  Analysed : 2 states
  Reachable : 0 state
  Translation: 0.02 seconds
  Computation: 0.00 seconds
```

Box 4: Simulation results under the OFMC and CL-AtSe back-ends.

(NRF) funded by the Ministry of Education (NRF-2010-0020210).

## References

[1] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[2] W.-H. Yang and S.-P. Shieh, "Password authentication schemes with smart cards," *Computers & Security*, vol. 18, no. 8, pp. 727–733, 1999.

[3] M.-S. Hwang and L.-H. Li, "A new remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28–30, 2000.

[4] H.-M. Sun, "An efficient remote use authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 958–961, 2000.

[5] H.-Y. Chien, J.-K. Jan, and Y.-M. Tseng, "An efficient and practical solution to remote authentication: smart card," *Computers & Security*, vol. 21, no. 4, pp. 372–375, 2002.

[6] M. L. Das, A. Saxena, and V. P. Gulati, "A dynamic ID-based remote user authentication scheme," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 629–631, 2004.

[7] Y.-Y. Wang, J.-Y. Liu, F.-X. Xiao, and J. Dan, "A more efficient and secure dynamic ID-based remote user authentication scheme," *Computer Communications*, vol. 32, no. 4, pp. 583–585, 2009.

[8] M. K. Khan, S.-K. Kim, and K. Alghathbar, "Cryptanalysis and security enhancement of a more efficient & secure dynamic ID-based remote user authentication scheme," *Computer Communications*, vol. 34, no. 3, pp. 305–309, 2011.

[9] Y.-H. An, "Security improvements of dynamic ID-based remote user authentication scheme with session key agreement," in *Proceedings of the 15th International Conference on Advanced Communication Technology: Smart Services with Internet of Things!, ICACT 2013*, pp. 1072–1076, January 2013.

[10] J. S. Chou, C. H. Huang, Y. S. Huang, and Y. Chen, "Efficient two-pass anonymous identity authentication using smart card," *IACR Cryptology ePrint Archive*, p. 402, 2013.

[11] R. Song, "Advanced smart card based password authentication protocol," *Computer Standards & Interfaces*, vol. 32, no. 5, pp. 321–325, 2010.

[12] Y.-F. Chang, W.-L. Tai, and H.-C. Chang, "Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 3430–3440, 2014.

[13] S. Kumari, M. K. Khan, and X. Li, "An improved remote user authentication scheme with key agreement," *Computers and Electrical Engineering*, vol. 40, no. 6, pp. 1997–2012, 2014.

[14] S. D. Kaul and A. K. Awasthi, "Security Enhancement of an Improved Remote User Authentication Scheme with Key Agreement," *Wireless Personal Communications*, vol. 89, no. 2, pp. 621–637, 2016.

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

[16] W. Jeon, J. Kim, J. Nam, Y. Lee, and D. Won, "An enhanced secure authentication scheme with anonymity for wireless environments," *IEICE Transactions on Communications*, vol. 95, no. 7, pp. 2505–2508, 2012.

[17] J. Nam, M. Kim, J. Paik, Y. Lee, and D. Won, "A provably-secure ECC-based authentication scheme for wireless sensor networks," *Sensors*, vol. 14, no. 11, pp. 21023–21044, 2014.

[18] S. Park, S. Kim, and D. Won, "ID-based group signature," *Electronics Letters*, vol. 33, no. 19, pp. 1616-1617, 1997.

[19] J. Moon, Y. Choi, J. Jung, and D. Won, "An improvement of robust biometrics-based authentication and key agreement scheme for multi-server environments using smart cards," *PLoS ONE*, vol. 10, no. 12, Article ID e0145263, 2015.

[20] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," *Advances in Cryptology CRYPTOΓ99*, pp. 388–397, 1999.

[21] J. Moon, Y. Choi, J. Kim, and D. Won, "An Improvement of Robust and Efficient Biometrics Based Password Authentication Scheme for Telecare Medicine Information Systems Using Extended Chaotic Maps," *Journal of Medical Systems*, vol. 40, no. 3, pp. 1–11, 2016.

[22] D. Mishra, A. Das, and S. Mukhopadhyay, "A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards," *Expert Systems with Applications*, vol. 41, no. 18, pp. 8129–8143, 2014.

[23] A. K. Das and A. Goswami, "A secure and efficient uniqueness-and-anonymity-preserving remote user authentication scheme for connected health care," *Journal of Medical Systems*, vol. 37, no. 3, pp. 1–16, 2013.

[24] B. Vaidya, J. H. Park, S.-S. Yeo, and J. J. P. C. Rodrigues, "Robust one-time password authentication scheme using smart card for home network environment," *Computer Communications*, vol. 34, no. 3, pp. 326–336, 2011.

[25] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 426, pp. 233–271, 1989.

[26] AVISPA, Automated validation of internet security protocols and applications, http://www.avispa-project.org/.

[27] D. von Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," in *Proceedings of APPSEM 2005 workshop*, pp. 1–17, 2005.

[28] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[29] Q. Jiang, J. Ma, G. Li, and L. Yang, "An efficient ticket based authentication protocol with unlinkability for wireless access networks," *Wireless Personal Communications*, vol. 77, no. 2, pp. 1489–1506, 2014.