Hindawi
Security and Communication Networks
Volume 2017, Article ID 1378128, 13 pages
https://doi.org/10.1155/2017/1378128

Research Article

Improving an Anonymous and Provably Secure Authentication Protocol for a Mobile User

Jongho Moon,[1] Youngsook Lee,[2] Jiye Kim,[3] and Dongho Won[4]

[1]*Department of Electrical and Computer Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do 16419, Republic of Korea*
[2]*Department of Cyber Security, Howon University, 64 Howondae 3-gil, Impi-myeon, Gunsan-si, Jeonrabuk-do 54058, Republic of Korea*
[3]*Department of Mobile Internet, Daelim University College, 29 Imgok-ro, Dongan-gu, Anyang-si, Gyeonggi-do 13916, Republic of Korea*
[4]*Department of Computer Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do 16419, Republic of Korea*

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

Received 4 May 2017; Accepted 16 August 2017; Published 27 September 2017

Academic Editor: Hongxin Hu

Recently many authentication protocols using an extended chaotic map were suggested for a mobile user. Many researchers demonstrated that authentication protocol needs to provide key agreement, mutual authentication, and user anonymity between mobile user and server and resilience to many possible attacks. In this paper, we cautiously analyzed chaotic-map-based authentication scheme and proved that it is still insecure to off-line identity guessing, user and server impersonation, and on-line identity guessing attacks. To address these vulnerabilities, we proposed an improved protocol based on an extended chaotic map and a fuzzy extractor. We proved the security of the proposed protocol using a random oracle and AVISPA (Automated Validation of Internet Security Protocols and Applications) tool. Furthermore, we present an informal security analysis to make sure that the improved protocol is invulnerable to possible attacks. The proposed protocol is also computationally efficient when compared to other previous protocols.

1. Introduction

Given recent developments in mobile telecommunications and the rapid spread of mobile devices, there is a growing importance of wireless and wired networking services that utilize bygone and current positional information from users carrying mobile devices with location tracking capabilities [1]. Remote user authentication schemes typically verify registered credentials using stored databases. Since Lamport [2] presented the first authentication scheme based on passwords in 1981, various remote user authentication schemes [3, 4] based on passwords have been proposed. However, since a server under a password-based remote user authentication protocol needs to store a verification table, which stores the password to determine the credentials of a remote user, the server arranges for extra storage for the verification table. Furthermore, several studies have shown that password-based remote user authentication protocols are insecure against some attacks, including off-line password guessing or stolen smart card attacks [5–7]. The problem with password-based authentication scheme is that it can be easily stolen or lost and making it difficult to remember on a regular basis. For these reasons, many researchers have presented new remote user authentication protocols that use biometrics. A major characteristic of biometrics is it uniqueness. Other advantage is that it cannot be guessed or stolen. Biological characteristics have been used in numerous remote user authentication schemes [8–13].

To design a secure authentication scheme, some cryptographic algorithms are also used, such as an RSA cryptosystem [14, 15], elliptic curve cryptography [16, 17], hash function [18, 19], and chaos-based cryptography [20–22].

Recently, many chaos-based authentication protocols have been suggested. Xiao et al. [23] first presented a user authentication protocol using a chaotic map and claimed that their protocol is useful and suitable for serviceable implementations. Unfortunately, many attacks were demonstrated by Han [31]. To overcome these vulnerabilities in [23], Han et al. [24] presented an enhanced user authentication protocol using chaos and asserted that their protocol resists all possible attacks. After that, Niu and Wang [32] proved that Han et al.'s protocol is vulnerable against an insider attack. Furthermore, Yoon [33] demonstrated that Niu and Wang's protocol does not resist a denial-of-service (DoS) attack. After that, Xue and Hong [34] proposed an improved authentication and key agreement protocol using a chaotic map to improve the security to some possible attacks. Unfortunately, Tan [35] found that Xue and Hong's protocol does not resist a man-in-the-middle attack. Lee et al. [25] presented an improved chaotic map-based authentication protocol, and He et al. [29] proved that Lee et al.'s protocol does not resist DoS and insider attacks. To enhance the functionality and security, Lin [26] proposed a new authentication and key agreement protocol using a chaotic map and dynamic identity. Unfortunately, Islam et al. [27] found that Lin's protocol cannot resist well-known attacks, and proposed an enhanced authentication protocol. However, we found that Islam et al.'s protocol is still insecure against off-line identity guessing, impersonation, and on-line identity guessing attacks.

The remainder of this paper is organized as follows. We briefly introduce the Chebyshev chaotic maps, threat assumptions, and fuzzy extractor that we adopt in the proposed protocol in Section 2. In Sections 3 and 4, we, respectively, review and cryptanalyze Islam et al.'s protocol. In Section 5, we propose an improved authentication and key agreement protocol for a mobile user. In Section 6, we present a security analysis of the proposed protocol. Section 7 explains the functionality and performance analyses comparing the proposed protocol to previous protocols. The conclusions are presented in Section 8.

1.1. Our Contribution. To address the security vulnerabilities in Islam et al.'s authentication protocol and obtain the required performance, we propose a security-improved scheme. The primary contribution of this paper are described below.

(i) First, we prove that Islam et al.'s protocol is still vulnerable to some attacks, and we show how an adversary can impersonate a legitimate user or server.

(ii) Second, we suggest an improved biometrics-based authentication and key agreement protocol on Islam et al.'s protocol. The improved protocol is designed to be secure to well-known attacks.

(iii) Third, we analyze that the proposed protocol has better robustness and a lower computational cost with a performance analysis.

2. Preliminaries

We briefly introduce the Chebyshev chaotic maps [28, 36], threat assumptions, and fuzzy extractor.

2.1. Chebyshev Chaotic Maps. The Chebyshev polynomial $T_k(v)$ is a v polynomial of degree k.

Definition 1. Let k be a whole number and w be a real number from the round $[-1, 1]$; the Chebyshev polynomial of degree k is then defined as $T_k(v) = \cos(k \cdot \arccos(v))$.

Definition 2 (CMDLP). Given the two parameters $v, w \in Z_n^*$, the Chaotic Maps Discrete Logarithm Problem is whether integer k can be found such that $w = T_k(v)$. The probability of $\mathscr{E}$ being able to address the CMDLP is defined as $\Pr[\mathscr{E}(v, w) = k : k \in Z_n^*,\ w = T_k(v) \bmod n]$.

Definition 3 (CMDHP). Given the three elements v, $T_j(v)$, and $T_k(v)$, the Chaotic Maps Diffie-Hellman Problem is whether $T_{jk}(v)$ can be computed such that $T_{jk}(v) = T_j(T_k(v)) = T_k(T_j(v))$.

2.2. Threat Assumptions. We introduce some threat model [37, 38] and consider constructing the threat assumptions described as follows:

(i) Adversary $\mathscr{E}$ can be both a user or server. Any registered mobile user can act as an adversary.

(ii) $\mathscr{E}$ can intercept all messages in a public channel, thereby capturing any message exchanged between a user or server.

(iii) $\mathscr{E}$ has the ability to modify, reroute, or delete the captured message.

(iv) Stored parameters can be extracted from the mobile device.

2.3. Fuzzy Extractor. In this subsection, we describe the basis for a biometric-based fuzzy extractor that converts biometric information data into a random value. Based on [39–41], the fuzzy extractor is operated through two procedures (Gen, Rep), demonstrated as

(i) $\mathrm{Gen}(\mathrm{BIO}) \rightarrow \langle \alpha, \beta \rangle$,

(ii) $\mathrm{Rep}(\mathrm{BIO}^*, \beta) = \alpha$ if BIO^* is reasonably close to BIO.

Gen is a probabilistic generation function for which the biometrics BIO returns an "extracted" string $\alpha \in \{0,1\}^k$ and auxiliary string $\beta \in \{0,1\}^*$, and Rep is a deterministic reproduction function that enables the recovery of α from β and any vector BIO^* close to BIO. Detailed information of the fuzzy extractor can be found in [42].

3. Review of Islam et al.'s Protocol

We review Islam et al.'s protocol. Their protocol consists of registration, login, verification, and password change phases and uses an extended chaotic maps. The term $T_k(a)$ is the chaotic map computation that is calculated with respect to "mod n" and $a \in (-\infty, +\infty)$. The notations of this paper are illustrated in the Notations.

3.1. Registration Phase

(i) User U_i selects the identity ID_i and password PW_i and inputs these values into the mobile devices MD_i. MD_i then chooses a random number t, calculates $W_i = \mathrm{PW}_i \oplus t$, and sends $\langle \mathrm{ID}_i, W_i \rangle$ to server S over an insecure channel.

(ii) Upon receiving $\langle \mathrm{ID}_i, W_i \rangle$, server S computes $H_i = h(s, \mathrm{ID}_i)$ and $n_i = h(W_i, \mathrm{ID}_i) \oplus (H_i, T_s(H_i))$ and sends $\langle n_i \rangle$ to user U_i by using a secure channel.

(iii) Upon receiving $\langle n_i \rangle$, MD_i retrieves $N_i = n_i \oplus h(W_i, \mathrm{ID}_i) \oplus h(\mathrm{ID}_i, \mathrm{PW}_i)$, $(H_i, T_s(H_i)) = N_i \oplus h(\mathrm{ID}_i, \mathrm{PW}_i)$, and $X_i = h(h(\mathrm{ID}_i, \mathrm{PW}_i) \parallel (H_i \parallel T_s(H_i))$ and stores $\langle N_i, X_i \rangle$ into MD_i.

3.2. Login Phase

(i) User U_i enters ID_i and PW_i into MD_i.

(ii) MD_i computes $(H_i \parallel T_s(H_i)) = N_i \oplus h(\mathrm{ID}_i, \mathrm{PW}_i)$ and $X_i' = h(h(\mathrm{ID}_i, \mathrm{PW}_i) \parallel (H_i \parallel T_s(H_i))$. MD_i then checks whether X_i' is equal to X_i. If this holds, MD_i executes the following stage; otherwise, MD_i rejects the login request.

(iii) MD_i chooses a random number k and then computes $Z_i = T_k(T_s(H_i))$ and $\mathrm{CID}_i = \mathrm{ID}_i \oplus (H_i \parallel T_1 \parallel Z_i)$, where $C_i = T_k(H_i)$, $R_i = H_i \oplus Z_i$, $V_i = h(\mathrm{CID}_i, Z_i, H_i, R_i, T_1)$, and T_1 is the current timestamp. MD_i sends $\langle \mathrm{CID}_i, C_i, V_i, R_i, T_1 \rangle$ to server S by using a public channel.

3.3. Verification Phase

(i) When receiving the request message $\langle \mathrm{CID}_i, C_i, V_i, R_i, T_1 \rangle$ from user U_i, server S verifies freshness of timestamp T_1 and terminates the session if $(T_2 - T_1) \leq \Delta T$ is false; otherwise, server S continues the next stage.

(ii) S computes $Z_i = T_s(C_i)$, $H_i = R_i \oplus Z_i$, $\mathrm{ID}_i = \mathrm{CID}_i \oplus (H_i \parallel T_1 \parallel Z_i)$, and $V_i' = h(\mathrm{CID}_i, Z_i, H_i, R_i, T_1)$. S then rejects the session if $V_i' \neq V_i$; otherwise, server S continues the following stage.

(iii) S randomly chooses a number l and computes the session key $\lambda = h(H_i, T_1, T_2, T_l(C_i))$, and $V_s = h(\lambda, H_i, T_1, T_2)$. S then sends the response messages $\langle V_s, T_2, T_l(H_i) \rangle$ over an insecure channel.

(iv) After receiving the response message $\langle V_s, T_2, T_l(H_i) \rangle$ from server S at time T_3, MD_i checks the freshness of T_2 and terminates the session if $(T_3 - T_2) \leq \Delta T$ is false; otherwise, MD_ie then computes $\lambda = h(H_i, T_1, T_2, T_k(T_l(H_i)))$, and $V_s' = h(\lambda, H_i, T_1, T_2)$. MD_i next checks whether $V_s' \stackrel{?}{=} V_s$. If this holds, MD_i accepts λ as the session key and authenticates server S; otherwise, MD_i rejects the session.

3.4. Password Change Phase

(i) User U_i inputs ID_i and PW_i into the mobile device MD_i.

(ii) MD_i computes $(H_i \parallel T_s(H_i)) = N_i \oplus h(\mathrm{ID}_i, \mathrm{PW}_i)$ and $X_i' = h(h(\mathrm{ID}_i, \mathrm{PW}_i) \parallel (H_i \parallel T_s(H_i)))$. MD_i then checks whether X_i' is the same to X_i. If this holds, the mobile device asks the new identity and password to U_i; otherwise, MD_i rejects the password change request.

(iii) U_i inputs a new ID_i^* and PW_i^* into MD_i. MD_i then computes $N_i^* = N_i \oplus h(\mathrm{ID}_i, \mathrm{PW}_i) \oplus h(\mathrm{ID}_i^*, \mathrm{PW}_i^*)$ and $X_i^* = h(h(\mathrm{ID}_i^*, \mathrm{PW}_i^*) \parallel (H_i \parallel T_s(H_i)))$ and replaces $\langle N_i, X_i \rangle$ by $\langle N_i^*, X_i^* \rangle$ into MD_i.

4. Cryptanalysis of Islam et al.'s Protocol

We cryptanalyze the security problems in Islam et al.'s protocol [27]. Islam et al. analyzed the protocol by Lin et al. and improved it to support an improved security functionality. However, we found that Islam et al.'s protocol was vulnerable to some possible attacks. These attacks are based on the threat assumptions that an adversary $\mathscr{E}$ was entirely monitored through the public channel connecting U_i and S in the login and verification phases and that $\mathscr{E}$ obtained the mobile device. Therefore, $\mathscr{E}$ can insert, modify, eavesdrop on, or delete any message transmitted over a public network. We now reveal further details of these problems.

4.1. Violation of the Identity. Let $\mathscr{E}$ be an active adversary who is a legitimate user and owns a mobile device to extract information $\langle N_{\mathscr{E}}, X_{\mathscr{E}} \rangle$ and suppose that an adversary $\mathscr{E}$ eavesdrops on the communication messages $\langle \mathrm{CID}_i, C_i, V_i, R_i, T_1, V_s, T_2, T_l(H_i) \rangle$ between user U_i and server S. $\mathscr{E}$ can then easily obtain the identity of user U_i. The details are described as follows:

(i) Adversary $\mathscr{E}$ calculates $(H_{\mathscr{E}} \parallel T_s(H_{\mathscr{E}})) = N_{\mathscr{E}} \oplus h(\mathrm{ID}_{\mathscr{E}}, \mathrm{PW}_{\mathscr{E}})$.

(ii) Using [43], the adversary computes $s' = (\arccos(T_s(H_{\mathscr{E}})) + 2k'\pi)/\arccos(H_{\mathscr{E}}), \forall k \in Z$.

(iii) $\mathscr{E}$ can then compute $Z_i' = T_{s'}(C_i)$, $H_i' = R_i \oplus Z_i$, and $\mathrm{ID}_i = \mathrm{CID}_i \oplus (H_i' \parallel T_1 \parallel Z_i')$.

4.2. On-Line Identity Guessing and User Impersonation Attack. Let $\mathscr{E}$ be an active adversary who is a legitimate user and owns a mobile device to extract information $\langle N_{\mathscr{E}}, X_{\mathscr{E}} \rangle$. $\mathscr{E}$ can then easily guess the identity of any user U_i and impersonate U_i as follows.

(i) Adversary $\mathscr{E}$ computes $(H_{\mathscr{E}} \parallel T_s(H_{\mathscr{E}})) = N_{\mathscr{E}} \oplus h(\mathrm{ID}_{\mathscr{E}}, \mathrm{PW}_{\mathscr{E}})$.

(ii) $\mathscr{E}$ generates a random number k, computes $Z_{\mathscr{E}} = T_k(T_s(H_{\mathscr{E}}))$, guesses any identity ID_i, and then computes $\mathrm{CID}_i = \mathrm{ID}_i \oplus (H_{\mathscr{E}} \parallel T_1 \parallel Z_{\mathscr{E}})$, where $C_{\mathscr{E}} = T_k(H_{\mathscr{E}})$, $R_{\mathscr{E}} = H_{\mathscr{E}} \oplus Z_{\mathscr{E}}$, $V_i = h(\mathrm{CID}_i, Z_{\mathscr{E}}, H_{\mathscr{E}}, R_{\mathscr{E}}, T_1)$, and T_1 is the current time stamp. MD_i sends $\langle \mathrm{CID}_i, C_{\mathscr{E}}, V_i, R_{\mathscr{E}}, T_1 \rangle$ to server S over an insecure network.

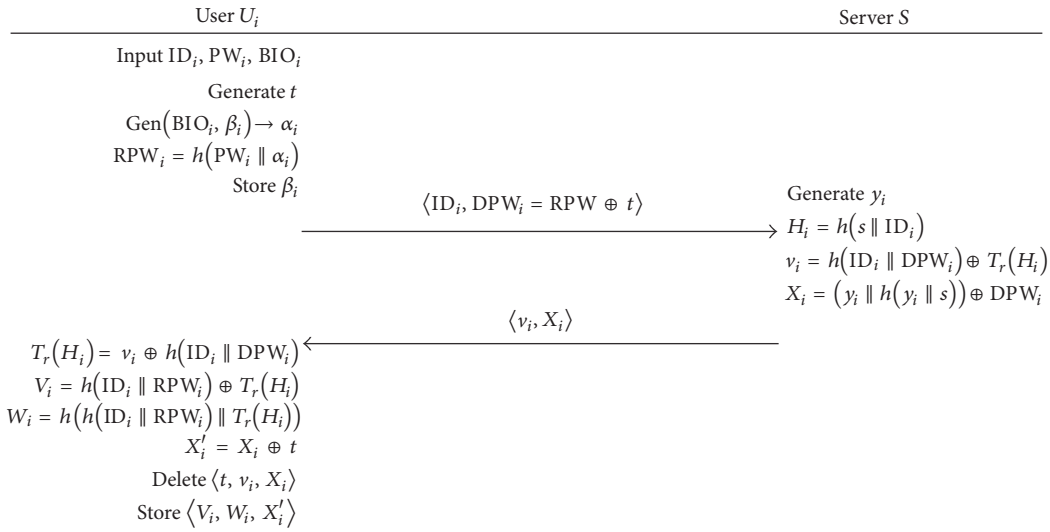

Figure 1: Registration phase of the proposed scheme.

(iii) Upon receiving the login request message $\langle \mathrm{CID}_i, C_{\mathscr{E}}, V_i, R_{\mathscr{E}}, T_1 \rangle$ from the adversary $\mathscr{E}$, server S verifies the freshness of the timestamp T_1 and terminates the session if $(T_2 - T_1) \leq \Delta T$ is false; otherwise, server S continues the next stage.

(iv) S computes $Z_{\mathscr{E}} = T_s(C_{\mathscr{E}})$, $H_{\mathscr{E}} = R_{\mathscr{E}} \oplus Z_{\mathscr{E}}$, $\mathrm{ID}_i = \mathrm{CID}_i \oplus (H_{\mathscr{E}} \| T_1 \| Z_{\mathscr{E}})$, and $V_i' = h(\mathrm{CID}_i, Z_{\mathscr{E}}, H_{\mathscr{E}}, R_{\mathscr{E}}, T_1)$. S then rejects the session if $V_i' \neq V_i$; otherwise, server S continues the following stage.

(v) S randomly chooses a number l and computes the session key $\lambda = h(H_{\mathscr{E}}, T_1, T_2, T_l(C))$, and $V_s = h(\lambda, H_{\mathscr{E}}, T_1, T_2)$. S then sends the response messages $\langle V_s, T_2, T_l(H_{\mathscr{E}}) \rangle$ over an insecure channel.

(vi) After receiving the response messages $\langle V_s, T_2, T_l(H_{\mathscr{E}}) \rangle$ from server S at time T_3, the mobile device checks the freshness of T_2 and terminates the session if $(T_3 - T_2) \leq \Delta T$ is false; otherwise, MD_i then computes $\lambda = h(H_{\mathscr{E}}, T_1, T_2, T_k(T_l(H_{\mathscr{E}})))$. Finally, $\mathscr{E}$ and S "successfully" conclude on the session key λ. However, server S faultily decides that he/she is communicating with user U_i.

4.3. Server Impersonation Attack. Let $\mathscr{E}$ be an active adversary who is a legitimate user and owns a mobile device to extract information $\langle N_{\mathscr{E}}, X_{\mathscr{E}} \rangle$. $\mathscr{E}$ can then easily impersonate S as follows.

(i) Adversary $\mathscr{E}$ computes $(H_{\mathscr{E}} \| T_s(H_{\mathscr{E}})) = N_{\mathscr{E}} \oplus h(\mathrm{ID}_{\mathscr{E}}, \mathrm{PW}_{\mathscr{E}})$.

(ii) Using [43], the adversary computes $s' = (\arccos(T_s(H_{\mathscr{E}})) + 2k'\pi)/\arccos(H_{\mathscr{E}}), \forall k \in Z$.

(iii) When receiving the login request message $\langle \mathrm{CID}_i, C_i, V_i, R_i, T_1 \rangle$ from user U_i, $\mathscr{E}$ computes $Z_i' = T_{s'}(C_i)$ and $H_i = R_i \oplus Z_i'$.

(iv) Adversary $\mathscr{E}$ randomly chooses a number l and computes the session key $\lambda = h(H_i, T_1, T_2, T_l(C_i))$, and $V_s = h(\lambda, H_i, T_1, T_2)$. The $\mathscr{E}$ then sends the response messages $\langle V_s, T_2, T_l(H_i) \rangle$ to user U_i over an insecure channel.

(v) After receiving the response message $\langle V_s, T_2, T_l(H_i) \rangle$ from adversary $\mathscr{E}$ at time T_3, the mobile device checks the freshness of T_2 and terminates the session if $(T_3 - T_2) \leq \Delta T$ is false; otherwise, MD_i then computes $\lambda = h(H_i, T_1, T_2, T_k(T_l(H_i)))$, and $V_s' = h(\lambda, H_i, T_1, T_2)$. The mobile device next checks whether $V_s' = V_s$. If this holds, the mobile device accepts λ as the session key. However, server S faultily decides that he/she is communicating with U_i.

4.4. Violation of the Session Key. Assume that any adversary $\mathscr{E}$ eavesdrops on the communication messages $\langle \mathrm{CID}_i, C_i, V_i, R_i, T_1, V_s, T_2, T_l(H_i) \rangle$ between user U_i and server S. $\mathscr{E}$ can then easily calculate the session key between U_i and S.

(i) $\mathscr{E}$ calculates $(H_{\mathscr{E}} \| T_s(H_{\mathscr{E}})) = N_{\mathscr{E}} \oplus h(\mathrm{ID}_{\mathscr{E}}, \mathrm{PW}_{\mathscr{E}})$.

(ii) Using [43], the adversary computes $s' = (\arccos(T_s(H_{\mathscr{E}})) + 2k'\pi)/\arccos(H_{\mathscr{E}}), \forall k \in Z$.

(iii) $\mathscr{E}$ can compute $Z_i' = T_{s'}(C_i)$ and $H_i = R_i \oplus Z_i'$.

(iv) Using [43], the adversary computes $k' = (\arccos(C_i) + 2k'\pi)/\arccos(H_i), \forall k \in Z$.

(v) $\mathscr{E}$ can then compute the session key $\lambda = h(H_i, T_1, T_2, T_{k'}(T_l(H_i)))$.

5. The Proposed Protocol

We will propose an improved biometric-based authentication protocol using the fuzzy extractor. The proposed protocol is also two members, user U_i and server S, and consists of four phases such as registration, login, verification, and password change. Figures 1 and 2 are the registration and login and verification phases of the proposed scheme.

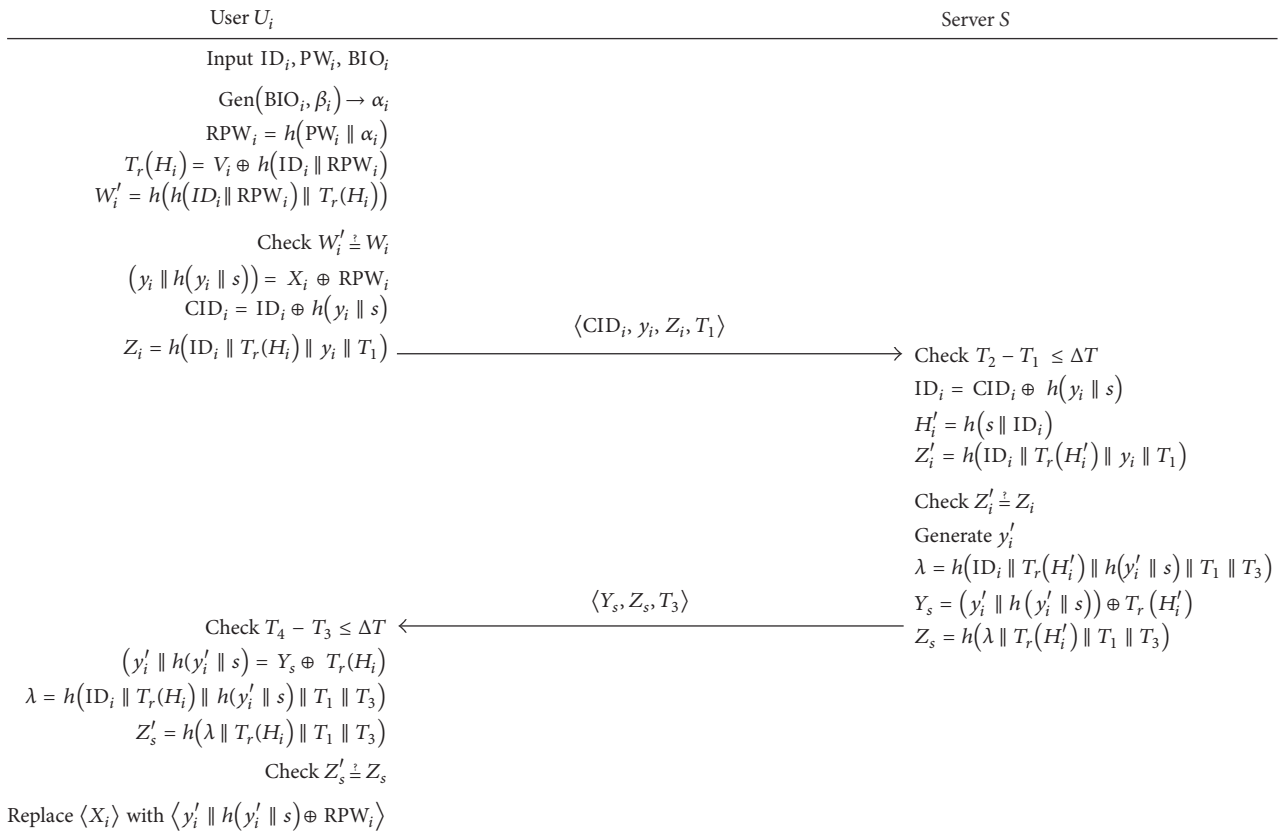

Figure 2: Login and verification phases of the proposed protocol.

5.1. Registration Phase

(i) U_i gives one's biometrics BIO_i at the mobile device MD_i. The MD_i then scans BIO_i, pulls out two random strings (α_i, β_i) from the computation $\mathrm{Gen}(\mathrm{BIO}_i) \rightarrow (\alpha_i, \beta_i)$, and stores β_i in storage. U_i enters the identity ID_i and password PW_i, and MD_i then calculates $\mathrm{RPW}_i = h(\mathrm{PW}_i \| \alpha_i)$. Finally, MD_i generates a random number t, stores t in the storage, and sends user registration request message $\langle \mathrm{ID}_i, \mathrm{DPW}_i = \mathrm{RPW}_i \oplus t \rangle$ to server S by using a secure communication channel.

(ii) Upon receiving the request message for registration, S randomly chooses a number y_i and calculates $H_i = h(s \| \mathrm{ID}_i)$, $v_i = h(\mathrm{ID}_i \| \mathrm{DPW}_i) \oplus T_r(H_i)$, and $X_i = (y_i \| h(y_i \| s)) \oplus \mathrm{DPW}_i$, where r is a fixed random positive integer and s is the master key of server S.

(iii) S sends $\langle v_i, X_i \rangle$ to the MD_i.

(iv) After receiving the registration response message $\langle v_i, X_i \rangle$, MD_i computes $T_r(H_i) = v_i \oplus h(\mathrm{ID}_i \| \mathrm{DPW}_i)$, $V_i = h(\mathrm{ID}_i \| \mathrm{RPW}_i) \oplus T_r(H_i)$, $W_i = h(h(\mathrm{ID}_i \| \mathrm{RPW}_i) \| T_r(H_i))$, and $X_i' = X_i \oplus t = (y_i \| h(y_i \| s)) \oplus \mathrm{RPW}_i$ and stores $\langle V_i, W_i, X_i' \rangle$ into storage after deleting t, v_i, and X_i.

5.2. Login Phase

(i) U_i enters ID_i and PW_i and gives BIO_i^* into the mobile device MD_i.

(ii) MD_i scans BIO_i^* and recovers α_i from the computation $\mathrm{Rep}(\mathrm{BIO}_i^*, \beta_i) \rightarrow \alpha_i$.

(iii) MD_i then computes $\mathrm{RPW}_i = h(\mathrm{PW}_i \| \alpha_i)$, $T_r(H_i) = V_i \oplus h(\mathrm{ID}_i \| \mathrm{RPW}_i)$, and $W_i' = h(h(\mathrm{ID}_i \| \mathrm{RPW}_i) \| T_r(H_i))$, and checks whether W_i' is the same to the stored W_i. If this holds, MD_i performs the next stage; otherwise, MD_i rejects the login request.

(iv) MD_i calculates $(y_i \| h(y_i \| s)) = X_i \oplus \mathrm{RPW}_i$, $\mathrm{CID}_i = \mathrm{ID}_i \oplus h(y_i \| s)$, and $Z_i = h(\mathrm{ID}_i \| T_r(H_i) \| y_i \| T_1)$, where T_1 is the current timestamp.

(v) Finally, MD_i sends the request message $\langle \mathrm{CID}_i, y_i, Z_i, T_1 \rangle$ for login to server S.

5.3. Verification Phase

(i) When receiving the request message $\langle \mathrm{CID}_i, y_i, Z_i, T_1 \rangle$ from MD_i, server S checks whether $T_2 - T_1 \leq \Delta T$ is valid, where ΔT is the minimum acceptable time interval and T_2 is the actual arrival time of login request. If this holds, S continues to proceed to the next stage; otherwise, S rejects the request.

(ii) S then calculates $\mathrm{ID}_i = \mathrm{CID}_i \oplus h(y_i \parallel s)$, $H_i = h(s \parallel \mathrm{ID}_i)$, and $Z_i' = h(\mathrm{ID}_i \parallel T_r(H_i) \parallel y_i \parallel T_1)$ and checks whether Z_i' is the same to the received Z_i. If this holds, the S continues to proceed to the next stage; otherwise, S terminates this session.

(iii) S randomly chooses a number y_i' and calculates the session key $\lambda = h(\mathrm{ID}_i \parallel T_r(H_i) \parallel h(y_i' \parallel s) \parallel T_1 \parallel T_3)$, $Y_s = (y_i' \parallel h(y_i' \parallel s)) \oplus T_r(H_i)$, and $Z_s = h(\lambda \parallel T_r(H_i) \parallel T_1 \parallel T_3)$. S then sends the login response message $\langle Y_s, Z_s, T_3 \rangle$ where T_3 is the current timestamp.

(iv) After receiving the response message $\langle Y_s, Z_s, T_3 \rangle$ from server S, MD_i checks whether $T_4 - T_3 \leq \Delta T$ is valid, where ΔT is the minimum acceptable time interval and T_4 is the actual arrival time of response message. If this holds, MD_i continues to the next stage; otherwise, MD_i terminates this session.

(v) MD_i computes $y_i' \parallel h(y_i' \parallel s) = Y_s \oplus T_r(H_i)$ and the session key $\lambda = h(\mathrm{ID}_i \parallel T_r(H_i) \parallel h(y_i' \parallel s) \parallel T_1 \parallel T_3)$ and $Z_s' = h(\lambda \parallel T_r(H_i) \parallel T_1 \parallel T_3)$ and verifies whether Z_s' is the same to the received Z_s. If this holds, MD_i continues to the next stage; otherwise, MD_i terminates current session.

(vi) Finally, MD_i replaces $\langle X_i \rangle$ by $\langle (y_i' \parallel h(y_i' \parallel s)) \oplus \mathrm{RPW}_i \rangle$ into storage.

5.4. Password Change Phase

(i) User U_i inputs ID_i and PW_i and gives BIO_i^* into the mobile device MD_i.

(ii) MD_i scans BIO_i^* and recovers α_i from the computation $\mathrm{Rep}(\mathrm{BIO}_i^*, \beta_i) \rightarrow \alpha_i$.

(iii) MD_i then computes $\mathrm{RPW}_i = h(\mathrm{PW}_i \parallel \alpha_i)$, $T_r(H_i) = V_i \oplus h(\mathrm{ID}_i \parallel \mathrm{RPW}_i)$, and $W_i' = h(h(\mathrm{ID}_i \parallel \mathrm{RPW}_i) \parallel T_r(H_i))$ and checks whether W_i' is the same to the stored W_i. If this holds, MD_i performs the next stage; otherwise, MD_i rejects the password change request.

(iv) U_i inputs a new password PW_i^* into MD_i. MD_i then computes $\mathrm{RPW}_i^* = h(\mathrm{PW}_i^* \parallel \alpha_i)$, $V_i^* = h(\mathrm{ID}_i \parallel \mathrm{RPW}_i^*) \oplus T_r(H_i)$, $W_i^* = h(h(\mathrm{ID}_i \parallel \mathrm{RPW}_i^*) \parallel T_r(H_i))$, and $X_i^* = X_i \oplus \mathrm{RPW}_i \oplus \mathrm{RPW}_i^*$.

(v) Finally, MD_i replaces $\langle V_i, W_i, X_i \rangle$ by $\langle V_i^*, W_i^*, X_i^* \rangle$ into storage.

6. Security Analysis of the Improved Protocol

The proposed protocol, which retains the advantages of Islam et al.'s protocol, is demonstrated, and it can resist some possible attacks and supports all security properties. The analysis of the improved protocol was organized with the threat assumptions made in Preliminaries.

6.1. Formal Security Analysis. A random oracle-based formal analysis is demonstrated here, and its security is shown. First, the following hash function is defined [44]:

Definition 4. A collision-resistance and one-way hash function $h : \{0,1\}^* \rightarrow \{0,1\}^k$ receives an input as a binary string of arbitrary length $v \in \{0,1\}^*$, returns a binary string of fixed length $h(v) \in \{0,1\}^k$, and gratifies the following conditions:

(i) Given $w \in W$, it is computationally impracticable to find a $v \in V$ such that $w = h(v)$.

(ii) Given $v \in V$, it is computationally impracticable to find another $v' \neq v \in V$, such that $h(v') = h(v)$.

(iii) It is computationally impracticable to find a pair $(v', v) \in V' \times V$, with $v' \neq v$, such that $h(v') = h(v)$.

Theorem 5. *According to the assumptions if hash function $h(\cdot)$ similarly acts like an random oracle, then the improved protocol is clearly secure to an adversary $\mathscr{E}$ to protect sensitive information, including identity ID_i, semigroup property $T_r(H_i)$, common session key λ, and master secret key s.*

Proof. Formal proof of the proposed protocol is similar in [40, 45], and it uses the oracle to construct $\mathscr{E}$, which will have the ability to extract ID_i, $T_r(H_i)$, λ, and s. □

Reveal. Random oracle can extract input value a from hash value $n = h(a)$ without failing. Adversary $\mathscr{E}$ now executes the experimental algorithm shown in Algorithm 1, $\mathrm{EXP}_{\mathrm{HASH,A}}^{\mathrm{BBSMK}}$ for the proposed scheme as BBSMK, for example. Let us then define the probability of success for $\mathrm{EXP}_{\mathrm{HASH,A}}^{\mathrm{BBSMK}}$ as $\mathrm{Success}_{\mathrm{HASH,A}}^{\mathrm{BBSMK}} = |\Pr[\mathrm{EXP}_{\mathrm{HASH,A}}^{\mathrm{BBSMK}} = 1] - 1|$, where $\Pr(\cdot)$ means the probability of $\mathrm{EXP}_{\mathrm{HASH,A}}^{\mathrm{BBSMK}}$. The advantage function for this algorithm then defines $\mathrm{Adv}_{\mathrm{HASH,A}}^{\mathrm{BBSMK}}(t, q_R) = \max_{\mathrm{Success}}$, where t and q_R are the execution time and number of queries. We then discuss the algorithm in Algorithm 1 for $\mathscr{E}$. If $\mathscr{E}$ has the capability to address the problem of hash function given in Definition 4, then he/she can immediately retrieve ID_i, $T_r(H_i)$, λ, and s. In that case, $\mathscr{E}$ will detect the complete connections between U_i and S; however, the inversion of the input from a given hash result is not possible computationally; that is, $\mathrm{Adv}_{\mathrm{HASH,A}}^{\mathrm{BBSMK}}(t) \leq \epsilon$, for all $\epsilon > 0$. Thus, $\mathrm{Adv}_{\mathrm{HASH,A}}^{\mathrm{BBSMK}}(t, q_R) \leq \epsilon$, since $\mathrm{Adv}_{\mathrm{HASH,A}}^{\mathrm{BBSMK}}(t, q_R)$ depends on $\mathrm{Adv}_{\mathrm{HASH,A}}^{\mathrm{BBSMK}}(t)$. In conclusion, there is no method for $\mathscr{E}$ to detect the complete connections between U_i and S, and the proposed protocol is distinctly invulnerable to an adversary $\mathscr{E}$ to retrieve $(\mathrm{ID}_i, T_r(H_i), \lambda, s)$.

6.2. Simulation Result Using AVISPA. We perform to simulate the improved protocol for formal analysis using the widely accepted AVISPA. The main contribution of the simulation is to prove that the improved protocol is invulnerable to man-in-the-middle and replay attacks. AVISPA tool consists of four back-ends: (1) On-the-Fly Model Checker (OFMC); (2) Constraint-Logic-Based Attack Searcher; (3) SAT-Based Model Checker; and (4) Tree Automata Based on Automatic Approximations for the Analysis of Security Protocols. In the AVISPA, the protocol is implemented in High-Level Protocol Specification Language (HLPSL) [44], which is based on the roles: the basic roles for representing each entity role and composition roles for representing the scenarios of the basic

(1) Eavesdrop the login request message $\{CID_i, y_i, Z_i, T_1\}$
(2) Call the Reveal oracle. Let $\langle ID_1', T_r(H_i)' \rangle \leftarrow$ Reveal(Z_i)
(3) Eavesdrop the authentication response message $\{Y_s, Z_s, T_3\}$
(4) Use the Reveal oracle. Let $\langle \lambda', T_r(H_i)'' \rangle \leftarrow$ Reveal(Z_s)
(5) **if** $(T_r(H_i)' = T_r(H_i)'')$ **then**
(6) Compute $y_i' \parallel h(y_i' \parallel s) = Y_s \oplus T_r(H_i)'$
(7) Call the Reveal oracle. Let $\langle ID_i'' \rangle \leftarrow$ Reveal(λ')
(8) **if** $(ID_i' == ID_i'')$ **then**
(9) Compute $h(y_i \parallel s) = CID_i \oplus ID_i'$
(10) Call the Reveal oracle. Let $\langle s' \rangle \leftarrow$ Reveal$(M_1 = h(y_i \parallel s))$
(11) Call the Reveal oracle. Let $\langle s'' \rangle \leftarrow$ Reveal$(M_2 = h(y_i' \parallel s))$
(12) **if** $(s' == s'')$ **then**
(13) Accept ID_i', $T_r(H_i)'$, λ', s' as the correct ID_i, $T_r(H_i)$, λ, s, respectively.
(14) **return** 0 (Success)
(15) **else**
(16) **return** 0 (Failure)
(17) **else**
(18) **return** 0 (Failure)
(19) **else**
(20) **return** 0 (Failure)
(21) **end if**

Algorithm 1: Algorithm $EXP^{BBSMK}_{HASH,A}$.

roles. The fundamental types available in the HLPSL are [46] as follows:

(i) agent: it means a primary name. The intruder always has the special identifier *i*.

(ii) symmetric_key: it is the key using the symmetric-key cryptosystem.

(iii) text: the text values are applied for messages. They are often used as nonces.

(iv) nat: the nat is used for meaning the natural numbers in nonmessage contexts.

(v) const: it is the type for representing constants.

(vi) hash_func: the basic type hash_func expresses collision-resistance secure one-way hash functions.

The role of the initiator, user U_i, is shown in Algorithm 2. U_i first receives the signal for starting and modifies its state variable from 0 to 1. This state variable is retained by the variable *state*. Similar to user, the roles of server S are implemented and shown in Algorithm 3. The specifications in HLPSL for the roles of environment, session, and goal are described in Algorithm 4. The result for the formal security verification of the improved protocol using OMFC is provided in Algorithm 5. It is clear that the improved protocol is invulnerable to passive and active attacks including the two attacks.

6.3. Informal Security Analysis

6.3.1. Mutual Authentication. Not only does the proposed scheme guarantee security as the other biometric-based schemes, but also U_i and S authenticate each other. S authenticates U_i by checking whether Z_i is valid or not, because only a legitimate user can compute a valid $h(ID_i \parallel T_r(H_i) \parallel y_i \parallel T_1)$ using a chaotic map. U_i then authenticates S by checking Z_s, which only S can compute using the long-term key s and timestamp T_3.

6.3.2. User Anonymity. To compromise the anonymity of user U_i, adversary $\mathscr{E}$ must be able to compute $h(y_i \parallel s)$. The value s is the master secret key of server S, and the random value y_i changes every session. Thus, the login request message changes every session. Even if adversary $\mathscr{E}$ eavesdrops on the login request message of a user U_i, $\mathscr{E}$ does not know ID_i. The proposed protocol provides user anonymity.

6.3.3. User Impersonation Attack. Suppose that an adversary $\mathscr{E}$ steals the mobile device MD_i of user U_i and extracts the parameters $\{V_i, W_i, y_i, \beta_i, X_i\}$ from MD_i. To make the login request message $\langle CID_i, y_i, Z_i, T_1 \rangle$, where $CID_i = ID_i \oplus h(y_i \parallel s)$ and $Z_i = h(ID_i \parallel T_r(H_i) \parallel y_i \parallel T_1)$, the server's master key s is needed. Without the master secret key s from server S, $\mathscr{E}$ cannot compute Z_i. The proposed protocol can therefore resist a user impersonation attack.

6.3.4. Privileged Insider Attack. In the proposed protocol, user U_i sends the login request message $\langle ID_i, DPW_i = RPW_i \oplus t \rangle$. Even if the privileged insider adversary $\mathscr{E}$ obtains these values $\langle ID_i, DPW_i = RPW_i \oplus t \rangle$, $\mathscr{E}$ does not know RPW_i and cannot impersonate user U_i. The proposed protocol can therefore resist a privileged insider attack.

6.3.5. Lost Mobile Device Attack. Suppose that user U_i's mobile device MD_i has been stolen or lost and any adversary $\mathscr{E}$ obtains it. $\mathscr{E}$ then tries to login to server S using MD_i; however, $\mathscr{E}$ does not know the correct password PW_i. To

```
role user (Ui, AS: agent,
SKuas: symmetric_key,
H, F: function,
SND, RCV: channel (dy))

played_by Ui def=

local State: nat,
IDi, PWi, BIOi, RPWi, DPWi, T, Ai: text,
Hi, Vi, VVi, R, S, Xi, Yi, Wi: text,
CIDi, Zi, T1, T3, SK, Y2, Ys, Zs: text
const as_ui_y2,
sc1, sc2, sc3, sc4: protocol_id

init State := 0

transition

(1) State = 0 ∧ RCV(start) =|>
State' := 1 ∧ T' := new()
∧ RPWi' := H(PWi.Ai)
∧ DPWi' := xor(RPWi',T')
∧ secret({PWi.Ai}, sc1, Ui)
∧ secret(IDi, sc2, {Ui,AS})
∧ SND({IDi.DPWi'}_SKuas)

(2) State = 2 ∧ RCV({ xor(H(IDi.xor(H(PWi.Ai),T')),F(R.H(S.IDi))).xor((Yi'.H(Yi'.S)),
xor(H(PWi.Ai),T')) }_SKuas) =|>
State' := 4 ∧ secret(R, S, sc3, AS)
∧ secret(F(R.H(S.IDi)), sc4, Ui, AS)
∧ VVi' := xor(H(IDi.H(PWi.Ai)), F(R.H(S.IDi)))
∧ Wi' := H(H(IDi.H(PWi.Ai)).F(R.H(S.IDi)))
∧ Xi' := xor((Yi'.H(Yi'.S)),H(PWi.Ai))
∧ CIDi' := xor(IDi, H(Yi'.S))
∧ T1' := new()
∧ Zi' := H(IDi.F(R.H(S.IDi)).Yi'.T1')
∧ SND(CIDi'.Yi'.Zi'.T1')

(3) State = 6 ∧ RCV(xor((Y2'.H(Y2'.S)),F(R.H(S.IDi))).H(SK.F(R.H(S.IDi)).T1'.T3').T3') =|>
State' := 8 ∧ SK' := H(IDi.F(R.H(S.IDi)).H(Y2'.S).T1'.T3')
∧ Xi' := xor((Y2'.H(Y2'.S)),H(PWi.Ai))
∧ request(Ui, AS, as_ui_y2, Y2')

end role
```

Algorithm 2: Role specification for user U_i.

login to S, the biometrics BIO_i is also needed. The proposed protocol can therefore resist a lost mobile device attack.

6.3.6. Replay Attack. One of the best solutions to prevent replay attack is to use a timestamp technique. The proposed protocol also uses timestamps. Even if any adversary $\mathscr{E}$ eavesdrops on any user's login request message and sends it to the server S, the server S checks the freshness of the timestamp and rejects the request. Furthermore, an adversary E cannot compute Z_i without ID_i and y_i. The proposed protocol can therefore resist a replay attack.

6.3.7. Off-Line Password Guessing Attack. To obtain a password of user U_i, the biometrics BIO_i is needed. Biometrics is uniquene and it cannot be guessed or stolen. The proposed protocol can therefore resist an off-line password guessing attack.

6.3.8. Stolen Verifier Attack. In the proposed protocol, a server S does not store any information related to the user's identity or password. The proposed protocol can therefore resist a stolen verifier attack.

6.3.9. Session Key Forward Security. One important objective of any user authentication protocols is to constitute a session key between user U_i and server S. The forward secrecy can protect previous and future session keys from adversary $\mathscr{E}$ if the master secret key of S is exposed. Suppose that the master

```
role applicationserver (Ui, AS: agent,
SKuas: symmetric_key,
H, F: function,
SND, RCV: channel(dy))

played_by AS def=

local State: nat,
IDi, PWi, BIOi, RPWi, DPWi, T, Ai: text,
Hi, Vi, VVi, R, S, Xi, Yi, Wi: text,
CIDi, Zi, T1, T3, SK, Y2, Ys, Zs: text
const as_ui_y2,
sc1, sc2, sc3, sc4: protocol_id

init State:= 1

transition

(1) State = 1 ∧ RCV(IDi.xor(H(PWi.Ai),T')) =|>
State' := 3 ∧ Hi' := H(S.IDi)
∧ Vi' := xor(H(IDi.xor(H(PWi.Ai),T')),F(R.H(S.IDi)))
∧ Yi' := new()
∧ Xi' := xor((Yi'.H(Yi'.S)),xor(H(PWi.Ai),T'))
∧ secret(F(R.H(S.IDi)), sc4, {Ui, AS})
∧ SND({Vi'.Xi'}_SKuas)

(2) State = 5 ∧ RCV(xor(IDi,H(Yi'.S).Yi'.H(IDi.F(R.H(S.IDi).Yi'.T1')).T1')) =|>
State' := 7 ∧ Hi' := H(S.IDi)
∧ Y2' := new()
∧ T3' := new()
∧ SK' := H(IDi.F(R.H(S.IDi)).H(Y2'.S).T1'.T3')
∧ Ys' := xor((Y2'.H(Y2'.S)),F(R.H(S.IDi)))
∧ Zs' := H(SK'.F(R.H(S.IDi)).T1'.T3')
∧ SND(Ys'.Zs'.T3')
∧ witness(AS, Ui, as_ui_y2, Y2')

end role
```

Algorithm 3: Role specification for application server AS.

secret key s of S is known to $\mathscr{E}$. However, $\mathscr{E}$ does not know $T_r(H_i)$. Thus, the session key $\lambda = h(ID_i \parallel T_r(H_i) \parallel h(y_i' \parallel s) \parallel T_1 \parallel T_3)$ of the improved protocol is still undiscovered to $\mathscr{E}$. Therefore, forward secrecy is retained in the proposed protocol.

7. Comparison of Functionality and Performance

This section presents comparisons of the functionality between the improved protocol and related protocols [23–28], and the computational spending between the improved protocol and the other protocols [25–30] is also compared here.

7.1. Functionality Analysis. Table 1 compares the security features provided by the proposed protocol with previous protocols. The results indicate that the proposed protocol is distinctly invulnerable and achieves all of the avoidance requirements.

7.2. Performance Analysis. We demonstrated the computational cost of the improved protocol against previous protocols in terms of the computational cost. According to the simulations obtained in [34], we found that $T_c \approx 32.40$ ms and $T_h \approx 0.20$ ms, respectively, with a system using Pentium IV 3.2 G (CPU) with a 3.0 GB (RAM). According to [47], the computational cost of the fuzzy extractor technique T_f is nearly identical to ECC multiplication. Kilinc and Yanik [48] has gauged the execution time of some cryptographic algorithms by using the Pairing-Based Cryptography Library (version 0.5.12) [49] in the OS: 32-bit Ubuntu 12.04.1, 2.2 G (CPU), and 2.0 G (RAM). They demonstrated that the cost to perform an elliptic curve point multiplication T_e is nearly 2.226 ms. In addition, they proved that the cost of a bitwise XOR operation is negligible. In Table 2, we presented the

Table 1: Functionality comparison of the improved protocol with others.

Property	[23]	[24]	[25]	[26]	[27]	[28]	The proposed
Mutual authentication	×	×	×	×	√	×	√
User anonymity	×	×	×	×	√	√	√
Impersonation attack	×	×	×	×	×	×	√
Insider attack	×	×	√	√	√	×	√
DoS attack	√	√	×	√	√	√	√
Replay attack	×	√	×	√	√	√	√
Off-line password guessing attack	×	√	×	×	√	×	√
Stolen verifier attack	×	√	√	√	√	√	√
Session key attack	×	×	×	×	×	×	√
Provable security	×	×	×	×	√	×	√

```
role session (Ui, AS: agent,
SKuas: symmetric_key,
H, F: function)

def=
local H1, H2, R1, R2: channel (dy)
composition

user (Ui, AS, SKuas, H, F, H1, R1)
∧ applicationserver (Ui, AS, SKuas, H, F, H2, R2)

end role

role environment() def=

const ui, as: agent,
skuas: symmetric_key,
h, f: function,
cidi, yi, zi, t1, ys, zs, t3: text,
as_ui_y2,
sc1, sc2, sc3, sc4: protocol_id

intruder_knowledge = ui, as, h, f, cidi, yi, zi, t1, ys, zs, t3

composition

session(ui, as, skuas, h, f)
∧ session(i, as, skuas, h, f)
∧ session(ui, i, skuas, h, f)

end role

goal

secrecy_of sc1, sc2, sc3, sc4
authentication_on as_ui_y2

end goal

environment()
```

Algorithm 4: Role specification for session, goal, and environment.

```
% OFMC
% Version of 2006/02/13
SUMMARY
SAFE

DETAILS
BOUNDED_NUMBER_OF_SESSIONS

PROTOCOL
/home/span/span/testsuite/results/testrv3.if

GOAL
as_specified

BACKEND
OFMC

COMMENTS
STATISTICS
parseTime: 0.00 s
searchTime: 0.03 s
visiteNodes: 4 nodes
depth: 2 piles
```

Algorithm 5: The result of simulation using OFMC backends.

computational cost of the improved protocol for each phase and execution time (millisecond) with the related schemes. Compared to Islam et al.'s protocol, the improved protocol performs seven further hash functions and two fuzzy-extract operations. However, we reduce four extended chaotic operations. The improved protocol therefore is more effective than Islam et al.'s protocol.

8. Conclusion

Recently, Islam et al. demonstrated the security vulnerabilities in Lin et al.'s protocol and presented an improved authentication protocol using extended chaotic map. Islam

Table 2: Performance comparison of the improved protocol with others.

	[25]	[29]	[26]	[27]	[28]	[30]	The proposed
Registration	$3T_h$	$4T_h$	$4T_h + T_c$	$6T_h + T_c$	$4T_h$	$3T_h + T_c$	$7T_h + T_c + T_f$
Login	$5T_h + T_c$	$3T_h + 2T_c$	$2T_h + 2T_c$	$3T_h + 2T_c$	$5T_h + 2T_c$	$2T_h + 3T_c$	$3T_h + T_f$
Verification	$6T_h + 5T_c$	$6T_h + 4T_c$	$6T_h + T_c$	$6T_h + T_c$	$7T_h + 4T_c$	$6T_h + 2T_c$	$8T_h + 1T_c$
Total	$14T_h + 6T_c$	$13T_h + 6T_c$	$14T_h + 4T_c$	$15T_h + 4T_c$	$16T_h + 6T_c$	$11T_h + 6T_c$	$18T_h + 2T_c + 2T_f$
Time (ms)	≈197.2	≈197.0	≈132.4	≈132.6	≈197.6	≈196.6	≈72.9

et al. also asserted that their authentication protocol is more secure than Lin et al.'s protocol and that it guarantees user anonymity. However, Islam et al.'s protocol is still insecure against some types of attacks, such as on-line identity guessing and user impersonation. To overcome these security weaknesses, in the current paper, we suggest an improved user authentication protocol using a fuzzy extractor that preserves the advantages of Islam et al.'s protocol and contributes to inclusive security properties. The formal and informal analyses of this work clarify why the improved protocol is more efficient and secure.

Notations

U_i: Mobile user
MD_i: Mobile device of user
ID_i: Identity of user
PW_i: Password of user
BIO_i: Biometrics of user
S: Remote server
x: Real number chosen set $[-1, 1]$
$T_k(x)$: Chebyshev polynomial of degree k
s: Master secret key of server S
r: Positive random integer generated server S
$h(\cdot)$: Cryptographic hash function
9 α_i, β_i: U_i's nearly random binary and auxiliary binary strings
λ: Session key
T: Timestamp
$\|$: Concatenation operator
$\oplus$: Bitwise XOR operator.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2010-0020210).

References

[1] N. Park, H. W. Kim, S. Kim, and D. Won, "Open location-based service using secure middleware infrastructure in web services," in *Proceedings of the International Conference on Computational Science and Its Applications - ICCSA 2005*, pp. 1146–1155, sgp, May 2005.

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

[3] M. Kumar, "On the weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards," *IACR Cryptology ePrint Archive*, pp. 163–174, 2004.

[4] H. Lin, "Efficient mobile dynamic ID authentication and key agreement scheme without trusted servers," *International Journal of Communication Systems*, vol. 30, no. 1, Article ID e2818, 2017.

[5] M. Khan and J. Zhang, "Improving the security of "a flexible biometrics remote user authentication scheme"," *Computer Standards and Interfaces*, vol. 29, no. 1, pp. 82–85, 2007.

[6] 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.

[7] D. He, N. Kumar, M. K. Khan, and J.-H. Lee, "Anonymous two-factor authentication for consumer roaming service in global mobility networks," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 4, pp. 811–817, 2013.

[8] 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.

[9] R. Amin, S. Islam, G. Biswas, M. Khan, and N. Kumar, "A robust and anonymous patient monitoring system using wireless medical sensor networks," *Future Generation Computer Systems*, 2015.

[10] R. Amin, R. Sherratt, D. Giri, S. Islam, and M. Khan, "A software agent enabled biometric security algorithm for secure file access in consumer storage devices," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 1, pp. 53–61, 2017.

[11] P. Mohit, R. Amin, and G. Biswas, "Design of authentication protocol for wireless sensor network-based smart vehicular system," *Vehicular Communications*, vol. 9, pp. 64–71, 2017.

[12] A. Chaturvedi, D. Mishra, S. Jangirala, and S. Mukhopadhyay, "A privacy preserving biometric-based three-factor remote user authenticated key agreement scheme," *Journal of Information Security and Applications*, vol. 32, pp. 15–26, 2017.

[13] D. Mishra, S. Kumari, M. Khan, and S. Mukhopadhyay, "An anonymous biometric-based remote user-authenticated key agreement scheme for multimedia systems," *International Journal of Communication Systems*, vol. 30, no. 1, Article ID e2946, 2017.

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

[15] R. Amin and G. Biswas, "An Improved RSA Based User Authentication and Session Key Agreement Protocol Usable in TMIS," *Journal of Medical Systems*, vol. 39, no. 8, article no. 79, 2015.

[16] 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.

[17] R. Amin, S. Islam, G. Biswas, M. Khan, and N. Kumar, "An Efficient and Practical Smart Card Based Anonymity Preserving User Authentication Scheme for TMIS using Elliptic Curve Cryptography," *Journal of Medical Systems*, vol. 39, no. 11, article no. 180, 2015.

[18] C. Chen, D. He, S. Chan, J. Bu, Y. Gao, and R. Fan, "Lightweight and provably secure user authentication with anonymity for the global mobility network," *International Journal of Communication Systems*, vol. 24, no. 3, pp. 347–362, 2011.

[19] H. Debiao, C. Jianhua, and Z. Rui, "A more secure authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 36, no. 3, pp. 1989–1995, 2012.

[20] S. Wu, Y. Zhu, and Q. Pu, "Robust smart-cards-based user authentication scheme with user anonymity," *Security and Communication Networks*, vol. 5, no. 2, pp. 236–248, 2012.

[21] P. Gong, P. Li, and W. Shi, "A secure chaotic maps-based key agreement protocol without using smart cards," *Nonlinear Dynamics. An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems*, vol. 70, no. 4, pp. 2401–2406, 2012.

[22] 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, article no. 70, pp. 1–11, 2016.

[23] D. Xiao, X. Liao, and S. Deng, "A novel key agreement protocol based on chaotic maps," *Information Sciences. An International Journal*, vol. 177, no. 4, pp. 1136–1142, 2007.

[24] S. Han, H. Tseng, R. Jan, and W. Yang, "A chaotic maps-based key agreement protocol that preserves user anonymity," in *Proceedings of the IEEE International Conference on Communications (ICCâ09*, pp. 1–6, Dresden, Germany, 2009.

[25] C. Lee, C. Chen, C. Wu, and S. Huang, "An extended chaotic maps-based key agreement protocol with user anonymity," *Nonlinear Dynamics. An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems*, vol. 69, no. 1-2, pp. 79–87, 2012.

[26] H. Lin, "Chaotic map based mobile dynamic ID authenticated key agreement scheme," *Wireless Personal Communications*, vol. 78, no. 2, pp. 1487–1494, 2014.

[27] S. Islam, M. Obaidat, and R. Amin, "An anonymous and provably secure authentication scheme for mobile user," *International Journal of Communication Systems*, vol. 29, no. 9, pp. 1529–1544, 2016.

[28] C. Lee and C. Hsu, "A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps," *Nonlinear Dynamics. An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems*, vol. 71, no. 1-2, pp. 200–211, 2013.

[29] D. He, Y. Chen, and J. Chen, "Cryptanalysis and improvement of an extended chaotic maps-based key agreement protocol," *Nonlinear Dynamics. An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems*, vol. 69, no. 3, pp. 1149–1157, 2012.

[30] D. Guo, Q. Wen, W. Li, H. Zhang, and Z. Jin, "Analysis and Improvement of 'Chaotic Map Based Mobile Dynamic ID Authenticated Key Agreement Scheme," *Wireless Personal Communications*, vol. 83, no. 1, pp. 35–48, 2015.

[31] S. Han, "Security of a key agreement protocol based on chaotic maps," *Chaos, Solitons & Fractals*, vol. 38, no. 3, pp. 764–768, 2008.

[32] Y. Niu and X. Wang, "An anonymous key agreement protocol based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 4, pp. 1986–1992, 2011.

[33] E. Yoon, "Efficiency and security problems of anonymous key agreement protocol based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 7, pp. 2735–2740, 2012.

[34] K. Xue and P. Hong, "Security improvement of an anonymous key agreement protocol based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 7, pp. 2969–2977, 2012.

[35] Z. Tan, "A chaotic maps-based authenticated key agreement protocol with strong anonymity," *Nonlinear Dynamics. An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems*, vol. 72, no. 1-2, pp. 311–320, 2013.

[36] C. Li, C. Lee, and C. Weng, "An extended chaotic maps based user authentication and privacy preserving scheme against DoS attacks in pervasive and ubiquitous computing environments," *Nonlinear Dynamics. An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems*, vol. 74, no. 4, pp. 1133–1143, 2013.

[37] D. Dolev and A. Yao, "On the security of public key protocols," *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[38] 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.

[39] Y. Dodis, B. Kanukurthi, J. Katz, and A. Smith, "Robust fuzzy extractors and authenticated key agreement from close secrets," *IEEE Transactions on Information Theory*, vol. 58, no. 9, pp. 6207–6222, 2012.

[40] A. Das, "A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor," *International Journal of Communication Systems*, vol. 30, no. 1, Article ID e2933, 2017.

[41] C. Wang, X. Zhang, and Z. Zheng, "Cryptanalysis and improvement of a biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor," in *PLoS One*, vol. 11, pp. 25-25, 2016.

[42] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: how to generate strong keys from biometrics and other noisy data," in *Advances in cryptology—EUROCRYPT 2004*, vol. 3027 of *Lecture Notes in Comput. Sci.*, pp. 523–540, Springer, Berlin, 2004.

[43] P. Bergamo, P. D'Arco, A. De Santis, and L. Kocarev, "Security of public-key cryptosystems based on Chebyshev polynomials," *IEEE Transactions on Circuits and Systems. I. Regular Papers*, vol. 52, no. 7, pp. 1382–1393, 2005.

[44] A. Das, "A secure and effective user authentication and privacy preserving protocol with smart cards for wireless communication," in *Networking Science*, vol. 2, pp. 12–27, 2, 2013.

[45] Y. Lu, L. Li, X. Yang, and Y. Yang, "Robust biometrics based authentication and key agreement scheme for multi-server environments using smart cards," *PLoS ONE*, vol. 10, no. 5, Article ID 0126323, 2015.

[46] von Oheimb D. The high-level protocol specification language hlpsl developed in the eu project avispa. In Proceedings of the Applied Semantics 2005 Workshop, Frauenchiemsee, Germany, 12–15 September 2005; pp. 1–17.

[47] M. Wazid, A. K. Das, S. Kumari, X. Li, and F. Wu, "Design of an efficient and provably secure anonymity preserving three-factor user authentication and key agreement scheme for TMIS," *Security and Communication Networks*, vol. 9, no. 13, pp. 1983–2001, 2016.

[48] H. Kilinc and T. Yanik, "A survey of SIP authentication and key agreement schemes," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 2, pp. 1005–1023, 2014.

[49] Lynn B. Pairing-based cryptography library, available at http://crypto.stanford.edu/pbc/.