electronics

MDPI

Review

Revisit of Password-Authenticated Key Exchange Protocol for Healthcare Support Wireless Communication

Mijin Kim [1], Jongho Moon [1], Dongho Won [1] and Namje Park [2,*]

[1] Department of Computer Engineering, Sungkyunkwan University, Gyeonggi-do 16419, Korea; mjkim@security.re.kr (M.K); jhmoon@security.re.kr (J.M.); dhwon@security.re.kr (D.W.)

[2] Department of Computer Education, Teachers College, Jeju National University, 61 Iljudong-ro, Jeju-si 690-781, JejuSpecial Self-Governing Province, Korea

* Correspondence: namjepark@jejunu.ac.kr

Received: 5 April 2020; Accepted: 28 April 2020; Published: 29 April 2020

Abstract: Wireless communication is essential for the infrastructure of a healthcare system. This bidirectional communication is used for data collection and to control message delivery. Wireless communication is applied in industries as well as in our daily lives, e.g., smart cities; however, highly reliable communication may be more difficult in environments with low power consumption, many interferences, or IoT wireless network issues due to resource limitations. In order to solve these problems, we investigated the existing three-party password-authenticated key exchange (3PAKE) and developed an enhanced protocol. Currently, Lu et al. presented a 3PAKE protocol to improve the security flaws found in Farash and Attari's protocol. This work revisits the protocol proposed by Lu et al. and demonstrates that, in addition to other security weaknesses, the protocol does not provide user anonymity which is an important issue for healthcare environment, and is not secure against insider attacks that may cause impersonation attacks. We propose a secure biometric-based efficient password-authenticated key exchange (SBAKE) protocol in order to remove the incidences of these threats, and present an analysis regarding the security and efficiency of the SBAKE protocol for practical deployment.

Keywords: authentication; security; key agreement; healthcare; wireless communication

1. Introduction

Healthcare systems have emerged as exchangers of information that utilize the internet to discern health issues. The implementation of this system has magnified privacy and security issues. Innovative technologies like artificial intelligence (AI) and internet of things (IoT) enable internet-connected "things" to analyze information via platforms for various services that are accessible to users. As various devices communicate using the existing network infrastructure, we must evaluate whether they are not compromised or connected with malicious adversaries, and permission must be obtained to access each device while establishing a connection. Moreover, protecting anonymity and privacy requires the employment of effective authentication and key management schemes. The password-authenticated key exchange (PAKE) protocol ensures that information transmitted among communication entities is available to the authorized party. The initial works, i.e., Bellovin-Merritt's two-party PAKE (2PAKE) [1] protocol proposals [2–5] are widely applied to establish session keys between two communicating parties in various communication environments. It is known that 2PAKE protocols strain storage capacity in large-scale peer-to-peer architectures. In order to effectively overcome this problem, researchers developed three-party password-authenticated key exchange (3PAKE) protocols [6–11], which enable two users to generate a shared cryptographically-strong key with the support of

an authentication server over an insecure open network. Potential security risks, privacy issues, and efficiency are still challenging tasks that must be achieved in order to enhance protection in IoT support 3PAKE. Many proposed PAKE protocols involve the use of a server's public key or a smart card [8,12–15], or both, to protect the user's password. Constructing a server's public key through key generation and key management constrains the capacity for greater complexity and increases the computational costs of the protocol, while using a smart card can weaken security due to the resultant exposure to side-channel attacks [16] that can disclose sensitive information stored on a smart card.

In [17], Huang proposed a simple 3PAKE protocol that does not involve a smart card. Following its publication, Yoon and Yoo figured out that Huang's protocol could not withstand an undetectable online password guessing attack [18]. Then, using the undetectable online password guessing attack, Tallapally demonstrated an unknown key-share attack on Huang's protocol and proposed a more secure and efficient scheme [19] to eliminate the security flaws. However, Farash and Attari indicated that Tallapally's scheme was vulnerable to an undetectable online password guessing attack and insecure against an offline password guessing attack, and proposed an enhanced protocol [11]. In [9], Lu et al. observed that Farash and Attari's protocol was still insecure against an offline password guessing attack, which causes an impersonation attack, and proposed a modified 3PAKE protocol for wireless communication (3WPAKE for short) without using smart cards. In [20], Chen et al. launched an offline password guessing attack on Lu et al.'s protocol, and proposed an enhanced version not preserving anonymity.

The contribution of this work is as follows. First, we point out the following weaknesses in the 3WPAKE protocol that are less feasible for healthcare support practical application: (1) 3WPAKE protocol does not achieve user anonymity or untraceability, and it is vulnerable to a privileged insider attack that causes impersonation attack. (2) As users in the 3WPAKE calculate exponentiation operations, it is inefficient for resource-constrained healthcare environments. Second, we develop a new secure biometric-based efficient password-authenticated key exchange (SBAKE) protocol and present the healthcare support wireless communication environment that is aimed to deploy the SBAKE protocol. To manage authorization and access to the server, we employ chaotic map and biometric verification, along with password verification. When chaos properties such as unpredictability are applied, there is an understanding of parameter sensitivity, e.g., of initial conditions, such that these properties satisfy the goal of efficiency, specifically, of being more computationally efficient than modular exponential computation and multiplication operations of an elliptic curve [21,22] and the essential properties of cryptography. Researchers have proposed security enhanced protocols [21,23–29] that use biological characteristics such as fingerprints or irises. A practical implementation is a fuzzy extractor for biometric key extraction. Fuzzy extractors have the advantage of protecting the biometric template by rendering its storage useless [30,31]. Their performance in terms of key entropy and key stability is consistently improved. Third, we prove that the proposed mechanism satisfies various security properties and provide formal security proofs using random oracle model and automated validation of internet security protocols and applications (AVISPA). Finally, we analyze that the SBAKE protocol performs better computational complexity and time consumption than other existing protocols. The high security and significantly low computation and communication costs of our protocol make it suitable for healthcare support PAKE protocols.

2. Materials and Methods

This section introduces the cryptographic one-way hash function [29], Chebyshev chaotic map [32,33], fuzzy extractor [24,25,34], notations used in this paper, threat model, and security properties.

Collision-Resistant one-way Hash Function

Definition 1. *A collision-resistant one-way hash function* $h : \{0,1\}^* \rightarrow \{0,1\}^n$ *takes a random length binary string* $x \in \{0,1\}^*$ *as an input and outputs a fixed n-bit binary string* $h(x) = \{0,1\}^n$. *The probability of an adversary A finding a collision is defined as*

$$Adv_A^{HASH}(t) = \Pr[A(x,x') : x \neq x', h(x) = h(x')], \quad (1)$$

where Pr*[E] refers to the probability an event E occurring, and A(x, x′) means that the pair (x, x′) is chosen by A. In this case, the probability in the advantage is computed over that of the random choices made by A with execution time t. The hash function h(·) is collision-resistant, if* $Adv_A^{HASH}(t) \leq \varepsilon$ *for sufficiently small* $\varepsilon > 0$.

Chebyshev Chaotic Map and its Properties

Let n be an integer and x be a real number within the interval $[-1, 1]$. The Chebyshev polynomial of degree n is defined as $T_n(x) = \cos(n \cdot \arccos(x))$. With this equation, the recurrence relation $T_n(x)$ is defined as $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$, where $n \geq 2$, $T_0(x) = 1$, $T_1(x) = x$, and satisfies the semigroup property: $T_r(T_s(x)) = T_{rs}(x) = T_s(T_r(x))$. In order to improve security, Zhang [35] proved that the semigroup property holds for Chebyshev polynomials defined over the interval $(-\infty, +\infty)$. In this work, the enhanced Chebyshev polynomials are used: $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)(mod\ p)$, where $n \geq 2$, $x \in (-\infty, +\infty)$, and p is a large prime. They are subject to the following two problems.

Definition 2. *Given x and y, the Chaotic Maps Discrete Logarithm Problem (CMDLP) is that finding an integer r such that* $y = T_r(x)$ *is computationally infeasible. The probability of an adversary A being able to solve the CMDLP is defined as*

$$Adv_A^{CMDLP}(t) = \Pr[A(x,y) = r : r \in Z_p^*, y = T_r(x)(mod\ p)]. \quad (2)$$

Definition 3. *Given x,* $T_r(x)$, *and* $T_s(x)$, *the Chaotic Map Diffie-Hellman Problem (CMDHP) is that calculating* $T_{rs}(x)$ *is computationally infeasible.*

Fuzzy Extractor

A fuzzy extractor extracts a string σ from its biometric input Bio_i in an error-tolerant way. This method was used in order to avoid the problem of bio-hash [30] and utilization of noisy biometrics. The fuzzy extractor method involves the following two operations:

Gen: This procedure is defined as $\text{Gen}(Bio_i) = (\sigma_i, \tau_i)$, where the biometric data Bio_i is the input of Gen, and it outputs an "extracted" secret key string $\sigma_i \in \{0,1\}^l$ of length l and an auxiliary public reproduction string τ_i.

Rep: This procedure takes a noisy biometric Bio_i' and its corresponding string τ_i as input, and if the Hamming distance between Bio_i and Bio_i' is less than the threshold th, the Rep procedure recovers the biometric key data σ_i: $\text{Rep}(Bio_i', \tau_i) = \sigma_i$ if $\text{d}(Bio_i, Bio_i') < th$.

If the input changes but remains close, the extracted σ_i remains the same. To assist in recovering σ_i from Bio_i', a fuzzy extractor outputs a public string τ_i. The extracted key σ_i from Bio_i by a fuzzy extractor can be used as a key in any cryptographic application, but, unlike traditional keys, need not be stored because it can be recovered from any Bio_i' close to Bio_i.

Notations

The notations used throughout this paper are listed in Table 1.

Table 1. Notations used in this paper and 3WPAKE.

Notation		Description
3WPAKE	**SBAKE**	
U_A, U_B	U_A, U_B	Communication parties user A and user B
S	S	Server
ID_A, ID_B	ID_A, ID_B	IDs of user A and user B
pw_A, pw_B	pw_A, pw_B	Passwords of user A and user B
p, q	p	Large prime numbers
Z_p	Z_p	Ring of integer modulo p
Z_p^*	Z_p^*	The multiplicative group of non-zero integers modulo p
g	-	A generator of G ($\subseteq Z_p^*$)
$h(\cdot)$	$h_1(\cdot)$	One-way hash function h_1:$\{0,1\}^*\rightarrow\{0,1\}^l$
-	$h_2(\cdot)$	One-way hash function h_2:$[-1, 1]\rightarrow\{0,1\}^l$
-	y_A, y_B	Random numbers selected by server S
r_A,	d	Random numbers selected by user A
r_B,	k	Random numbers selected by user B
x, y, z	-	Random exponents selected by user A, user B, and server S, respectively
-	s	Secret master key selected by server S
H_A, V_A, W_A, R_A	f_A, C_A,W_A, Y_A, α_A	Authentication parameters of user A
H_B, V_B, W_B, R_B	f_B, C_B, W_B, Y_B, α_B	Authentication parameters of user B
N_s, T_A, T_B	P_A, P_B, Q_A, Q_B	Authentication parameters of server S
K_{SA}, K_{SB}	S_A, S_B	Server S's keys used to authenticate user A and user B
K_{AS}, K_{BS}	-	User's keys used to authenticate the server S
$Auth_A$, $Auth_B$	$Auth_A$, $Auth_B$	Computed user's parameters used to authenticate user A and user B
Q_A, Q_B	V_A, V_B, I_A, I_B,	Computed server's parameters for user A and user B
sk	sk	Session key shared with user A and user B
$\oplus$, $\|$	$\oplus$, $\|$	Exclusive-or and concatenation operation
-	Bio_A, Bio_B	Biometric data of user A and user B
-	Gen, Rep	Fuzzy generator and reproduction procedure, respectively
-	σ_A, σ_B	Biometric secret key of user A and user B
-	τ_A, τ_B	Biometric public reproduction parameter of user A and user B
-	th	Error tolerance threshold used by fuzzy extractor
-	$T_n(.)$	Chebyshev polynomial of degree n

Abbreviations: SBAKE, secure biometric-based efficient password-authenticated key exchange; 3WPAKE, three-party password-authenticated key exchange for wireless communication.

Threat Model

We introduce the following security assumptions [16] regarding the capabilities of the probabilistic polynomial-time adversary A to achieve the security properties of the 3PAKE protocols for wireless communications.

- A can eavesdrop, insert, intercept, alter, and delete messages exchanged among the protocol, user U_A, user U_B, and the server S.
- A may be a legitimate protocol participant (an insider), an external party (an outsider), or some combination of the two.

Security Requirements

A secure 3PAKE protocol with mutual anonymity in wireless communication should satisfy the following requirements [36,37]:

- User anonymity: Even if an adversary eavesdrops on the messages transmitted in the communication parties, the user's identity should be protected.
- Mutual authentication: Two partnering users and the server can authenticate one another.
- Session key security: No one except for those who are partnered can establish the session key.
- Known-key security: When a particular session key is lost, it does not reveal the other session keys.
- Forward secrecy: Even if a user's password is compromised, it does not reveal past session keys or the new password.
- Robustness: The protocol should withstand various types of attacks, such as offline password guessing, replay, insider, and impersonation.

3. Results

In this section, we analyze why the 3WPAKE does not achieve anonymity or untraceability, and why it is vulnerable to privileged insider attack that causes impersonation attacks. Moreover, as users in and the 3WPAKE protocol calculate exponentiation operations, it is inefficient for resource-constrained healthcare mobile environments. The details are as follows.

3.1. Revisit of 3WPAKE Protocol

Before we explain the issue of failure to achieve privacy and other security properties, this section revisits the 3WPAKE protocol.

Step 1. U_A randomly selects x, $r_A \in Z_p^*$, computes $H_A = h(pw_A) \oplus r_A$, $V_A = h(pw_A \| r_A \| ID_A)$, $R_A = g^x \oplus V_A \pmod q$, and sends $M_{A1} = \{H_A, V_A, R_A, ID_A\}$ to S. Similarly, U_B sends $M_{B1} = \{H_B, V_B, R_B, ID_B\}$ to S.

Step 2. With the received messages M_{A1} and M_{B1}, S computes r_A, $r_B \in Z_p^*$ using the known passwords pw_A and pw_B, and checks whether or not the received V_A and V_B are equal to $h(pw_A \| r_A \| ID_A)$ and $h(pw_B \| r_B \| ID_B)$, respectively. Then, S computes $R_A' = R_A \oplus h(pw_A \| r_A \| ID_A)$ *and* $R_B' = R_B \oplus h(pw_B \| r_B \| ID_B)$, chooses a random number $z \in Z_p^*$ and computes $N_S = g^z \pmod q$, $K_{sa} = (R_A')^z \pmod q$, $K_{SB} = (\mathrm{R_B}')^z \pmod q$, $T_A = h(pw_A \| r_A \| K_{SA} \| ID_A)$, $T_B = h(pw_B \| r_B \| K_{SB} \| ID_B)$. After that, S sends the messages $M_{SA1} = \{T_A, ID_B, N_S\}$ to U_A and $M_{SB1} = \{T_B, ID_A, N_S\}$ to U_B.

Step 3. With the received message M_{SA1}, U_A computes $K_{AS} = (N_S)^{\mathrm{x}} \pmod q$, checks whether or not the received T_A is equal to $h(pw_A \| r_A \| K_{AS} \| ID_A)$, then computes $W_A = h(ID_A \| ID_B \| K_{AS} \| pw_A \| r_A)$, sends $M_{A2} = \{ID_A, W_A\}$ to S. Simultaneously, with the received message M_{SB1}, U_B sends $M_{B2} = \{ID_B, W_B\}$ to S.

Step 4. With the received messages M_{A2} and M_{B2}, S checks whether or not $W_A = h(ID_A \| ID_B \| K_{SA} \| pw_A \| r_A)$ and $W_B = h(ID_A \| ID_B \| K_{SB} \| pw_B \| r_B)$, computes $Q_A = K_{SB} \oplus h(ID_A \| ID_B \| K_{SA} \| pw_A \| r_A)$ and $Q_B = K_{SA} \oplus h(ID_A \| ID_B \| K_{SB} \| pw_B \| r_B)$. Then, S sends Q_A to U_A and Q_B to U_B.

Step 5. With the received message Q_A, U_A computes $K_{SB} = Q_A \oplus h(U_A \| U_B \| K_{SA} \| pw_A \| r_A)$, $sk = (K_{SB})^x \pmod q$, $Auth_A = h(sk \| ID_A \| ID_B)$, sends $Auth_A$ to U_B. With the received message Q_B, U_B sends $Auth_B$ to U_A.

Step 6. With the received $Auth_A$ and $Auth_B$, U_A and U_B evaluate the correctness of $Auth_A$ and $Auth_B$, then agree on the session key $sk = g^{xyz}$.

3.2. Security Analysis of 3WPAKE Protocol

Violating anonymity: An authentication protocol could provide user anonymity if no adversary can compromise the user's identity by launching active or passive attacks in any phase [21]. In the 3WPAKE protocol, messages M_{A1}, M_{B1}, M_{SA1}, M_{SB1}, M_{A2}, and M_{B2} include plain text information about the U_A and U_B, so an adversary can easily acquire communication entities for sending and receiving messages through eavesdropping on the communication channel. This violates the preservation of user anonymity (privacy), which is a basic property of the authentication protocol.

Violating untraceability: User untraceability means that an adversary cannot identify any previous sessions involving the same user. In the 3WPAKE protocol, messages M_{A1}, M_{B1}, M_{SA1}, M_{SB1}, M_{A2}, and M_{B2} include plain text information about the U_A and U_B so that an adversary can easily acquire communication entities for sending and receiving messages through eavesdropping on the communication channel. This violates the preservation of user untraceability, which is a basic property of an authentication protocol.

Vulnerability to privileged insider attack: Authenticated principals acting maliciously form the basis of a powerful attacking model. This model has been used by other researchers such as Bellare and Rogaway [38], who assumed that the adversary can corrupt any principal at any time. It was pointed out by Gollmann [39] that this corresponds to the most realistic situation in commercial applications, where most real-world attacks come from insiders. A privileged insider attack occurs

when an administrator can access a user's password so as to impersonate that user. In Step 2 of the 3WPAKE protocol, S knows U_A and U_B's identities and passwords; so as malicious S can easily conduct a privileged insider attack as follows.

1. Malicious S selects $x, r_A \in Z_p^*$ imitating U_A, computes $R_A = g^x \oplus h(pw_A \| r_A \| ID_A)$, $K_{SA} = (R_A)^z$, $Q_B = K_{SA} \oplus h(ID_A \| ID_B \| K_{SB} \| pw_B \| r_B)$, and sends Q_B to U_B in Step 4 of the 3WPAKE protocol.

2. Malicious S can compute the session key $sk = (K_{SB})^x$, $Auth_A = h(sk \| ID_A \| ID_B)$, then sends $Auth_A$ and ID_A to U_B.

Therefore, malicious S easily impersonates U_A, and shares a session key sk with U_B, without any authentication problem. The session key should be computed only by the intended parties like U_A and U_B, and not by S. A secure authenticated key exchange protocol should block the malicious S impersonating any legal user. This vulnerability may cause security risks in a real situation, like the IoT environment. We must modify the 3WPAKE protocol such that no malicious inside attacker can impersonate any legal user.

Vulnerability to replay attack: Suppose an adversary A records the login request message M_{A1} and resends it to S, then S computes r_A, V_A, R_A', N_S, K_{SA}, T_A, and sends a response message M_{SA1} to U_A without recognizing that the login request message was old and had been sent again. In the 3WPAKE protocol, S cannot distinguish between old login request messages and fresh login request messages. Even if A cannot compute the session key, this vulnerability may be exploited by an adversary, leading to a waste of system resources that can threaten the entire system.

Inefficient authentication phase: Lu et al. [9] assume open access to wireless services for wireless communications using various portable devices (mobile phones, laptops, USB thumb drives, and PDAs). Efficiency is crucial for resource-constrained portable devices. In the 3WPAKE protocol, U_A and U_B compute exponentiation operations in Step 1, 3, and 5 in the previous section, which exhausts resource-constrained devices. We need to improve efficiency in order to satisfy the need for secure/private access to services via wireless communication networks.

3.3. The SBAKE Protocol

This section demonstrates our SBAKE protocol, which fixes the vulnerabilities of the 3WPAKE protocol by applying biometric data and adopting a chaotic map that is much more efficient than performing point multiplication operations of the elliptic curve [21]. Figure 1 presents the healthcare support wireless communication environment that is aimed to deploy the SBAKE protocol, where users can be patients and staff can be doctors, pharmacists, or the medical billing center. Users and staff register to the server, then perform the login and authentication phase. The data of user is collected using IoTs and transferred to the server, which is able to store and process a huge amount of data. The data is accessible to the staff of the healthcare organization, then the staff can provide health services to the users. We assume that the following information has been pre-established in the registration phase. In order to compute message size, based on [16], we set both the block size of one-way hash function $h_1(.)$, $h_2(.)$, and Chebyshev chaotic map to a length of 20 bytes; identities ID_A, ID_B, and passwords pw_A, pw_B to 8 bytes, and the random numbers d and k to 16 bytes in length.

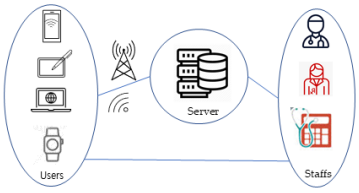


Figure 1. Healthcare support wireless communication environment in the SBAKE protocol.

Registration phase: By performing the following steps (Figure 2a), a new U_A registers to S.

Figure 2. (**a**) Registration phase; (**b**) Password and biometric update; (**c**) Login and authentication phase of SBAKE protocol.

Step 1. U_A selects ID_A, pw_A, and inputs biometric Bio_A in U_A's mobile device. U_A extracts σ_A and τ_A as Gen(Bio_A) = (σ_A, τ_A) by applying a Fuzzy extractor on Bio_A, computes $h_1(pw_A \| \sigma_A)$, and sends $\{ID_A, h_1(pw_A \| \sigma_A)\}$ to S via a secure channel.

Step 2. With the received information from U_A, S randomly selects y_A, computes $V_A = h_1(y_A \| s) \oplus h_1(s)$, $S_A = h_1(ID_A \| s) \oplus h_1(pw_A \| \sigma_A)$, $I_A = ID_A \oplus h_1(y_A \| s)$, stores $\{S_A, I_A, h_1(y_A \| s)\}$ in its database, and sends $\{S_A, V_A\}$ to U_A via a secure channel.

U_A computes $f_A = h_1(ID_A \oplus \mathrm{pw_A} \oplus \sigma_A)$, and stores $\{f_A, \tau_A, S_A, V_A, \mathrm{Gen}, \mathrm{Rep}\}$ in U_A's mobile device.

Login and authentication phase: By performing the following steps (Figure 2c), U_A and U_B login to S, authenticate each other, and securely share a session key.

Step 1. U_A inputs ID_A, pw_A, $Bio_A{}'$, and computes $\sigma_A{}^* = \mathrm{Rep}(Bio_A{}', \tau_A)$, $f_A{}^* = h_1(ID_A \oplus pw_A \oplus \sigma_A{}^*)$. If $f_A{}^* \neq f_A$, then the process aborts, otherwise (U_A, pw_A, and σ_A are approved) U_A randomly generates $d \in Zp^*$ and computes $C_A = ID_A \oplus T_d(x)$, $W_A = S_A \oplus h_1(pw_A \| \sigma_A) = h_1(ID_A \| s)$, $Y_A = W_A \oplus C_A \oplus ID_B$, $\alpha_A = h_1(ID_A \| C_A \| W_A)$. Then U_A sends $M_{A1} = \{C_A, Y_A, \alpha_A, V_A\}$ to S. Similarly, U_B inputs ID_B, pw_B,

Bio_B, and computes σ_B*= Rep(Bio_B, τ_B), f_B* = $h_1(ID_B \oplus pw_B \oplus \sigma_B$*). If f_B* ≠ f_B, then the process aborts, otherwise U_B randomly generates $k \in Z_p$* and computes $C_B = ID_B \oplus T_k(x)$, $W_B = S_B \oplus h_1(\ pw_B \| \sigma_B) = h_1(ID_B \| s)$, $Y_B = W_B \oplus C_B \oplus ID_A$, $\alpha_B = h_1(ID_B \| C_B \| W_B)$. Then U_B sends $M_{B1} = \{C_B, Y_B, \alpha_B, V_B\}$ to S.

Step 2. When receiving messages M_{A1} and M_{B1}, S computes $h_1(y_A \| s) = V_A \oplus h_1(s)$ using its secret master key s, derives $ID_A = I_A \oplus h_1(y_A \| s)$, $T_d(x) = C_A \oplus ID_A$, and $W_A = h_1(ID_A \| s)$, similarly, $h_1(y_B \| s) = V_B \oplus h_1(s)$, $T_k(x) = C_B \oplus ID_B$, and $W_B = h_1(ID_B \| s)$. Then, S checks whether or not the received α_A and α_B are equal to $h_1(ID_A \| C_A \| W_A)$ and $h_1(ID_B \| C_B \| W_B)$, respectively. Then, S derives $Y_A \oplus W_A \oplus C_A = ID_B$, $Y_B \oplus W_B \oplus C_B = ID_A$, so S acquires communication partners, then computes $P_A = h_1(h_1(pw_A \| \sigma_A) \| T_d(x) \| ID_A \| ID_B)$, $Q_A = T_k(\mathrm{x}) \oplus ID_A$, similarly, $P_B = h_1(h_1(pw_B \| \sigma_B) \| T_k(x) \| ID_B \| ID_A)$, $Q_B = T_d(\mathrm{x}) \oplus ID_B$, and sends $M_{A2} = \{P_A, Q_A\}$ to U_A and $M_{B2} = \{P_B, Q_B\}$ to U_B.

Step 3. With the received M_{A2}, U_A checks whether or not P_A is equal to $h_1(h_1(pw_A \| \sigma_A\ ) \| T_d(x) \| ID_A \| ID_B)$, and derives $T_k(x) = Q_A \oplus ID_A$, computes $sk = h_2(T_d(x), T_k(x), T_{dk}(x))$ and $Auth_A = h_1(sk \| T_k(x))$. Then, U_A sends $Auth_A$ to U_B. Similarly, with the received M_{B2}, U_B checks whether or not P_B is equal to $h_1(h_1(pw_B \| \sigma_B) \| T_k(x) \| ID_B \| ID_A)$, and derives $T_d(x) = Q_B \oplus ID_B$, computes $sk = h_2(T_d(x), T_k(x), T_{dk}(x))$ and $Auth_B = h_1(sk \| T_d(x))$. Then, U_B sends $Auth_B$ to U_A. Finally, U_A and U_B agree on the session key $sk = h_2(T_d(x), T_k(x), T_{dk}(x))$.

Based on the above descriptions, in the login and authentication phase, the message size of the $\{C_A, Y_A, \alpha_A, V_A\}$, $\{C_B, Y_B, \alpha_B, V_B\}$, $\{P_A, Q_A\}$, $\{P_B, Q_B\}$, $Auth_A$, and $Auth_B$ can be computed as (20 + 20 + 20 + 20) = 80 bytes, (20 + 20 + 20 + 20) = 80 bytes, (20 + 20) = 40 bytes, (20 + 20) = 40 bytes, 20 bytes, and 20 bytes, respectively. Adding all of these together, the communication overhead becomes (80 + 40 + 20) * 2 = 280 bytes.

Password and biometric update: If U_A intends to update his/her password and biometric data, U_A inputs old information $\{ID_A, {pw_A}^{old}, {Bio_A}^{old}\}$ in the U_A's mobile device, and computes ${\sigma_A}^{old}$ = Rep(${Bio_A}^{old}$, ${\tau_A}^{old}$), ${f_A}^{old} = h_1(ID_A \oplus {pw_A}^{old} \oplus \sigma_{\mathrm{A}}{}^{\mathrm{old}})$. If ${f_A}^{old} \neq f_A$, then terminates the connection. Otherwise, U_A inputs new password ${pw_A}^{new}$ and new biometric data ${Bio_A}^{new}$ in the U_A's mobile device, and computes Gen(${Bio_A}^{new}$) = (${\sigma_A}^{new}$, ${\tau_A}^{new}$), ${f_A}^{new} = h_1(ID_A \oplus {pw_A}^{new} \oplus {\sigma_A}^{new})$, ${S_A}^{new} = {S_A}^{old} \oplus h_1({pw_A}^{old} \| {\sigma_A}^{old}) \oplus h_1({pw_A}^{new} \| {\sigma_A}^{new})$ then replaces $\{{f_A}^{old}, {\tau_A}^{old}, {S_A}^{old}\}$ with $\{{f_A}^{new}, {\tau_A}^{new}, {S_A}^{new}\}$ into the U_A's mobile device (Figure 2b).

3.4. Security Analysis and Proof of SBAKE Protocol

This section presents the security analysis of the SBAKE protocol.

3.4.1. Simulation using AVISPA

We simulate the SBAKE protocol for formal analysis using the widely accepted simulation tool named AVISPA. The main contribution of this simulation is verifying whether the SBAKE protocol is secure of two attacks, i.e., man-in-the-middle attack and replay attack. This simulation tool is composed of four back-ends: (1) On-the-fly Model-Checker; (2) Constraint-Logic-based Attack Searcher; (3) SAT-based Model Checker; and (4) Tree Automata based on Automatic Approximations or the Analysis of Security Protocols [40].

The SBAKE protocol is implemented in High Level Protocol Specification Language (HLPSL) [41] in AVISPA. The role of U_A is shown in Table 2a. U_A first receives the start signal, then renews its state value from 0 to 1. This value is retained by the variable state. In a similar way, the roles of U_B and S of the SBAKE protocol are described in Table 2b,c, respectively. The roles of the session, goal, and environment are described in Table 2d. The simulation result of the SBAKE protocol using CL-AtSe is shown in Table 3. The result shows that the SBAKE protocol is secure of two attacks: replay and man-in-the-middle attacks.

Table 2. Role specification for (**a**) user U_A, (**b**) user U_B, (**c**) user S, (**d**) session, goal, and environment.

(a)

```
role usera(Ua, AS, Ub: agent,
SKas: symmetric_key,
H, F: hash_func,
SND, RCV: channel(dy))
played_by Ua def=
local State: nat,
IDa, IDb, PWa, BIOa, Oa, Ga, RPWa: text,
Ya, Va, Wa, Sa, La, S: text,
Aa, Ca, Pa, Pb, Qa, Qb, D, K, X, Y1, Y2, SKab: text

init State:= 0

transition

1. State = 0 /\ RCV(start) =|>
State':= 1 /\ RPWa':= H(PWa.Oa)
/\ secret({PWa,Oa}, sc1, Ua)
/\ secret(IDa, sc2, {Ua, Ub, AS})
/\ SND({IDa.RPWa'}_SKas)

2. State = 2
/\ RCV({xor(H(IDa.S),H(PWa.Oa)).xor(H(Y1'.S),H(
S)).X}_SKas) =|>
State':= 4 /\ D':= new()
/\ Ca':= xor(IDa,F(D'.X))
/\ Wa':=
xor(xor(H(IDa.S),H(PWa.Oa)),H(PWa.Oa))
/\ Ya':= xor(Wa',Ca',IDb)
/\ Aa':= H(IDa.Ca'.Wa')
/\ SND(Ca'.xor(H(Y1'.S),H(S)).Ya'.Aa')

3. State = 6
/\ RCV(H(H(PWa.Oa).F(D'.X).IDa.IDb).xor(F(K'.X)
,IDa)) =|>
State':= 7 /\ SKab':=
H(F(D'.X).xor(xor(F(K'.X),IDa),IDa).F(D'.xor(xor(F(
K'.X),IDa),IDa)))

end role
```

(b)

```
role userb(Ua, AS, Ub: agent,
SKbs: symmetric_key,
H, F: hash_func,
SND, RCV: channel (dy))
played_by Ub def=
local State: nat,
IDa, IDb, PWb, BIOb, Ob, Gb, RPWb: text,
Yb, Vb, Wb, Sb, Lb, S: text,
Ab, Cb, Pa, Pb, Qa, Qb, D, K, X, Y1, Y2,
SKba: text

init State:= 0

transition

1. State = 0 /\ RCV(start) =|>
State':= 1 /\ RPWb':= H(PWb.Ob)
/\ secret({PWb,Ob}, sc3, Ub)
/\ secret(IDb, sc4, {Ua, Ub, AS})
/\ SND({IDb.RPWb'}_SKbs)

2. State = 3
/\ RCV({xor(H(IDb.S),H(PWb.Ob)).xor(H(Y
2'.S),H(S)).X}_SKbs) =|>
State':= 5 /\ K':= new()
/\ Cb':= xor(IDb,F(K'.X))
/\ Wb':=
xor(xor(H(IDb.S),H(PWb.Ob)),H(PWb.Ob))
/\ Yb':= xor(Wb',Cb',IDa)
/\ Ab':= H(IDb.Cb'.Wb')
/\ SND(Cb'.xor(H(Y2'.S),H(S)).Yb'.Ab')

3. State = 6
/\ RCV(H(H(PWb.Ob).F(K'.X).IDb.IDa).xor
(F(D'.X),IDb)) =|>
State':= 8 /\ SKba':=
H(F(K'.X).xor(xor(F(D'.X),IDb),IDb).F(K'.xo
r(xor(F(D'.X),IDb),IDb)))

end role
```

(c)

```
role applicationserver(Ua, AS, Ub: agent,
SKas, SKbs: symmetric_key,
H, F: hash_func,
SND, RCV: channel (dy))

played_by AS def=
local State: nat,
IDa, IDb, PWa, PWb, BIOa, BIOb, Oa, Ga, Ob, Gb,
RPWa, RPWb: text,
Ya, Yb, Va, Vb, Sa, Sb, La, Lb, S, Y1, Y2: text,
Aa, Ab, Ca, Cb, Fa, Fb, Pa, Pb, Qa, Qb, D, K, X: text

init State:= 1

transition
```

(d)

```
role session(Ua, Ub, AS: agent,
SKas, SKbs: symmetric_key,
H, F: hash_func)

def=
local Z1, Z2, Z3, S1, S2, S3: channel (dy)
composition

usera(Ua, AS, Ub, SKas, H, F, Z1, S1)
/\ userb(Ua, AS, Ub, SKbs, H, F, Z2, S2)
/\ applicationserver(Ua, AS, Ub, SKas,
SKbs, H, F, Z3, S3)

end role

role environment() def=
```

Table 2. *Cont.*

1. State = 1 /\ RCV(IDa.H(PWa.Oa)) =|>	
State′:= 2 /\ Y1′:= new()	const ua, as, ub: agent,
/\ Va′:= xor(H(Y1′.S),H(S))	skas, skbs, skab, skba: symmetric_key,
/\ Sa′:= xor(H(IDa.S),H(PWa.Oa))	h, f: hash_func,
/\ La′:= xor(IDa,H(Y1′.S))	ca, va, ya, aa: text,
/\ secret(S, sc5, AS)	cb, vb, yb, ab: text,
/\ secret(H(Y1′.S), sc6, AS)	pa, qa, pb, qb: text,
/\ SND({Sa′.Va′.X}_SKas)	sc1, sc2, sc3, sc4, sc5, sc6, sc7, sc8: protocol_id
2. State = 3 /\ RCV(IDb.H(PWb.Ob)) =|>	
State′:= 4 /\ Y2′:= new()	intruder_knowledge = {ua, as, ub, h, f, ca,
/\ Vb′:= xor(H(Y2′.S),H(S))	va, ya, aa, cb, vb, yb, ab, pa, qa, pb, qb}
/\ Sb′:= xor(H(IDb.S),H(PWb.Ob))	
/\ Lb′:= xor(IDb,H(Y2′.S))	composition
/\ secret(H(Y2′.S), sc7, AS)	
/\ SND({Sb′.Vb′.X}_SKbs)	session(ua, as, ub, skas, skbs, h, f)
3. State = 5	end role
/\ RCV(xor(IDa,F(D′.X)).xor(H(Y1′.S),H(S)).xor(H(IDa.S),xor(IDa,F(D′.X)),	
IDb).H(IDa.xor(IDa,F(D′.X)).H(IDa.S)))	goal
/\ RCV(xor(IDb,F(K′.X)).xor(H(Y2′.S),H(S)).xor(H(IDb.S),xor(IDb,F(K′.X)),IDa).	secrecy_of sc1
H(IDb.xor(IDb,F(K′.X)).H(IDb.S))) =|>	secrecy_of sc2
State′:= 6 /\ Pa′:=	secrecy_of sc3
H(xor(H(IDa.S),xor(H(IDa.S),H(PWa.Oa))).F(D′.X).	secrecy_of sc4
IDa.IDb)	secrecy_of sc5
/\ Qa′:= xor(F(K′.X),IDa)	secrecy_of sc6
/\ Pb′:=	secrecy_of sc7
H(xor(H(IDb.S),xor(H(IDb.S),H(PWb.Ob))).F(K′.X).	end goal
IDb.IDa)	
/\ Qb′:= xor(F(D′.X),IDb)	environment()
/\ SND(Pb′.Qb′)	
/\ SND(Pa′.Qa′)	
end role	

Table 3. The result of simulation CL-AtSe backends.

SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
TYPED_MODEL
PROTOCOL
/home/span/span/testsute/results/test.if
GOAL
As Specified
BACKEND
CL-AtSe
STATISTICS
Analysed: 0 states
Reachable: 0 states
Translation: 0.02 seconds
Computation: 0.00seconds

3.4.2. Formal Security Proof

This subsection describes the formal security analysis of the SBAKE using the random oracle model and demonstrates that the protocol is secure. First, we recall Definition 1 of collision resistant one-way hash function in the preliminaries section.

Theorem 1. *Under the assumption that the collision-resistant one-way hash function $h(\cdot)$ closely behaves like an oracle, then the SBAKE is provably secure against an adversary A for the protection of a user U_A's personal information including the identity ID_A, password pw_A and biometric key σ_A of the user U_A, secret master key s that is selected by S, a shared secret key S_A between the U_A and S, and the session key sk between users U_A and U_B.*

Proof. The formal proof of the SBAKE protocol is similar to those shown in [41–43]. Using the following oracle model to construct an A that will have the ability to derive U_A's identity ID_A, password pw_A, biometric key σ_A, the secret master key s that is selected by S, the shared session key sk between U_A and U_B, and the shared secret key S_A between U_A and S.

Reveal: This random oracle will unconditionally output the input x from given hash result $y = h(x)$.

Now, an adversary A runs the experimental algorithm shown in Algorithm 1, $EXP^{SBAKE}_{HASH,A}$ for the SBAKE protocol. We define the success probability for $EXP^{SBAKE}_{HASH,A}$ as $Success^{SBAKE}_{HASH,A} = |\Pr[EXP^{SBAKE}_{HASH,A} = 1] - 1|$. The advantage function for this experiment becomes $Adv^{SBAKE}_{HASH,A}(t, q_R) = max_A\{Success^{SBAKE}_{HASH,A}\}$, where the maximum is taken over all of A with execution time t and the number of queries q_R made to the Reveal oracle. Considering the experiment presented in Algorithm 1, we acquire that if there exists a Reveal oracle that can invert $h_1(.)$ and $h_2(.)$, then A could directly derive U_A's identity ID_A, password pw_A, biometric key σ_A, the secret master key s selected by S, shared session key sk, and the shared secret key S_A between the U_A and S. In this case, A will discover the complete connections between U_A and S; however, it is computationally infeasible to invert a one-way hash function $h(\cdot)$, i.e., $Adv^{SBAKE}_{HASH,A}(t) \leq \varepsilon$, $\forall \varepsilon > 0$. Then, we have $Adv^{SBAKE}_{HASH,A}(t, q_R) \leq \varepsilon$ since $Adv^{SBAKE}_{HASH,A}(t, q_R)$ depends on $Adv^{SBAKE}_{HASH,A}(t)$. Therefore, the SBAKE protocol is provably secure against adversaries for deriving $\{ID_A, pw_A, Bio_A, s, S_A, sk\}$. Deriving $\{ID_B, pw_B, Bio_B, S_B\}$ shows a similar phenomenon, so we omit the description here. Hence, the theorem is proven. □

3.4.3. Informal Security Proof

This subsection examines that the SBAKE protocol is resistant against various known attacks and achieves the basic security properties described in the preliminaries section. Figure 3a compares the security attributes among the SBAKE and other existing protocols.

SA1. Anonymity: Assume that an adversary A intercepts communication messages M_{A1}, M_{B1}, M_{SA1}, M_{SB1}, $Auth_A$, $Auth_B$ during the login and authentication phase, then ID_A and ID_B cannot be derived from $\{C_A, Q_A\}$ *and* $\{C_B, Q_B\}$ without knowing the random numbers d and k selected by communicating users U_A and U_B, respectively. Moreover, because of the one-way hash function, ID_A and ID_B cannot be derived from $\{\alpha_A, P_A\}$, and $\{\alpha_B, P_B\}$. Therefore, the SBAKE protocol provides user anonymity.

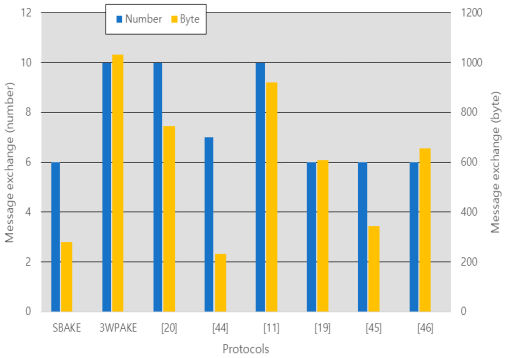
SA2. Untraceability: The messages $\{C_A\ Y_A, \alpha_A\}$ and $\{P_A, Q_A, Auth_A\}$ sent during the login and authentication phase are changed dynamically in each session. Since U_A generates a random number d and computes $T_d(x)$ in each session, messages $\{C_A\ Y_A, \alpha_A\}$ and $\{P_A, Q_i, Auth_A\}$ differ from the messages in the previous sessions. Therefore, the SBAKE protocol provides user untraceability.

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

Eavesdrop login request message $\{C_A, Y_A, \alpha_A, V_A\}$
Call the Reveal oracle. Let $(ID_A', C_A', W_A') \leftarrow$ Reveal (α_A)
Call the Reveal oracle. Let $(ID_A'', s') \leftarrow$ Reveal (W_A')
if $(C_A' = C_A)$ and $(ID_A' = ID_A'')$ **then**
  Accept ID_A' as the correct ID_A of user U_A and s' as the correct private key of S
  Compute $T_d(x) = C_A \oplus ID_A$
  Eavesdrop login response message $\{P_A, Q_A\}$
  Call the Reveal oracle.
  Let $(h_1(pw_A' \| \sigma_A'), T_d(x)', ID_A''', ID_B''') \leftarrow$ Reveal (P_A)
  Call the Reveal oracle.
  Let $(pw_A'', \sigma_A'') \leftarrow$ Reveal $(h_1(pw_A' \| \sigma_A'))$
  if $(T_d(x)' = T_d(x))$ and $(ID_A''' = ID_A)$ **then**
    Accept pw_A'' and σ_A'' of the correct password and biometric key of user U_A
    Compute $S_A = h_1(pw_A \| \sigma_A) \oplus h_1(ID_A \| s)$
    Compute $T_k(x) = Q_A \oplus ID_A$
    Evesdrop authentication message $\{Auth_A\}$
    Call the Reveal oracle.
    Let $(sk', T_k(x)') \leftarrow$ Reveal $(Auth_A)$
    if $(T_k(x)' = T_k(x))$ **then**
      Accept S_A as the correct shared secret key between U_A and S, sk' as the session key sk shared between U_A and U_B
      return 1
    else
      reutrn 0
    end if
  else
    return 0
  end if
else
  return 0
end if

Security attributes⇨ / Schemes⇩	*SA1*	*SA2*	*SA3*	*SA4*	*SA5*	*SA6*	*SA7*
SBAKE	√	√	√	√	√	√	√
3WPAKE	x	x	x	x	√	x	√
[20]	x	x	x	x	√	√	√
[44]	x	x	x	√	x	x	x
[11]	x	x	x	x	√	x	√
[19]	x	x	x	x	√	x	√
[45]	x	x	x	x	√	√	√
[46]	x	x	x	x	√	x	√

(a)



(b)

Figure 3. (**a**) SA1: provide anonymity, SA2: provide untraceability, SA3: provide mutual authentication, SA4: resist privileged insider attack, SA5: provide session key secrecy, SA6: resist offline password guessing attack, SA7: provide perfect forward secrecy, √: yes, x: no; (**b**) performance comparison between SBAKE and existing other protocols.

SA3. Mutual authentication: In the SBAKE protocol, when the message $M_{A1} = \{C_A, Y_A, \alpha_A, V_A\}$ is received from U_A, S computes $h_1(y_A \| s) = V_A \oplus h_1(s)$ using its secret master key s and derives $ID_A = I_A \oplus h_1(y_A \| s)$. S computes $W_A = h_1(ID_A \| s)$ and $\alpha_A{}' = h_1(ID_A \| C_A \| W_A)$, then checks whether or not the received α_A is equal to $\alpha_A{}'$. If this is indeed the case, U_A is authenticated. U_A authenticates S by checking the verification of P_A. Finally, the authentication between U_A and U_B is completed through the correctness of $Auth_B$ and $Auth_A$, which are only available to themselves.

SA4. Privileged insider attack: In the registration phase of the SBAKE protocol, U_A sends a registration message $\{ID_A, h_1(pw_A \| \sigma_A)\}$ to S via a secure channel. The pw_A is protected under the one-way hash function $h_1(\cdot)$ and the biometric key σ_A. Thus, guessing a password pw_A from $h_1(pw_A \| \sigma_A)$ is computationally infeasible. Therefore, the SBAKE protocol is secure against privileged insider attacks.

SA5. Session key security: Suppose that adversary A intercepts all of the messages $\{C_A, Y_A, \alpha_A, V_A, P_A, Q_A, Auth_A\}$ and $\{C_B, Y_B, \alpha_B, V_B, P_B, Q_B, Auth_B\}$ that are transmitted via public channel among U_A, U_B, and S, steals the mobile devices of U_A and U_B, then extracts all information $\{f_A, f_B, \tau_A, \tau_B, S_A, S_B, V_A, V_B, Gen, Rep\}$; however, A cannot compute the session key $sk = h_2(T_d(x), T_k(x), T_{dk}(x))$. Even if A obtains $T_d(x)$ and $T_k(x)$, A is required to solve the CMDLP for computing the $sk = h_2(T_d(x), T_k(x), T_{dk}(x))$. In order to compute $T_d(x)$ and $T_k(x)$ from $\{C_A, Q_A\}$ and $\{C_B, Q_B\}$, ID_A and ID_B are needed, respectively. In order to retrieve ID_A and ID_B from $\{C_A, Q_A\}$ and $\{C_B, Q_B\}$, A needs to know $T_d(x)$ and $T_k(x)$, respectively. Without knowing d and k, which are selected randomly by communicating with users U_A and U_B, respectively, A cannot obtain $T_d(x)$ and $T_k(x)$. Moreover, computing $T_{dk}(x)$ from $T_d(x)$ and $T_k(x)$ is required if one wants to obtain the session key, then one will be required to solve the CMDHP. Therefore, the SBAKE protocol provides session key security.

SA6. Offline password guessing attack: Assume that adversary A steals the mobile device of U_A and extracts all of the information stored in the device. This information includes $\{f_A, \tau_A, S_A, V_A, Gen, Rep\}$ that $f_A = h(ID_A \oplus pw_A \oplus \sigma_A)$, $S_A = h(ID_A \| s) \oplus h(pw_A \| \sigma_A)$. The ID_A and pw_A are protected by one-way hash function $h_1(\cdot)$ by the secret parameter including the server's secret master key s and secret biometric σ_A, so guessing the ID_A and pw_A is computationally infeasible without knowing the biometric Bio_A of U_A and the server's master secret key s. Moreover, U_A's ID_A and pw_A are not sent during the communication; so the SBAKE protocol is resistant against offline password guessing attacks.

SA7. Perfect forward secrecy: In the SBAKE protocol, even if adversary A obtains U_A and U_B 's $\{pw_A, pw_B\}$ or S's secret master key s, he/she cannot compute the previous session keys. In order to compute previous session key sk, A has to know the random numbers d and k generated by U_A and U_B. First, A cannot obtain the user's identity from the login and authentication information $\{C_A, Y_A, \alpha_A, V_A, P_A, Q_A, Auth_A\}$ and $\{C_B, Y_B, \alpha_B, V_B, P_B, Q_B, Auth_B\}$, and even if A obtains $T_d(x)$ and $T_k(x)$, A is required to solve the CMDHP in order to compute the $sk = h_2(T_d(x), T_k(x), T_{dk}(x))$. Because of CMDLP, it is infeasible to find d or k from $T_d(x)$ and $T_k(x)$, and the random numbers $\{d, k\}$ are different for each session. Therefore, the SBAKE protocol provides perfect forward secrecy.

4. Discussion

This section presents a comparison made between the SBAKE protocol and the other existing protocols [9,11,19,20,44–46] for the computation and communication complexities that express the superiority of the SBAKE efficiency. In Table 4, the performance comparison is made for the total computation cost measured by each operation's computation complexity, and the communication cost is measured by numbers and the length of a message exchange among entities. For the computation complexity comparison, the definitions of T_H, T_{ECC}, T_{EXP}, T_{CCM}, and T_{FE} are the time complexity of a one-way cryptographic hash function, an elliptic curve point multiplication, a modular exponentiation, a Chebyshev chaotic map, and a fuzzy extractor operation used in biometric verification, respectively.

As shown in Table 4, we observed that SBAKE, 3WPAKE, Chen et al. [20], Xie et al. [44], Farash and Attari [11], Tallapally [19], Wu et al. [45], Chang et al. [46] protocols require the computation

complexities of $21T_H + 4T_{CCM} + 2T_{FE}$, $20T_H + 9T_{EXP}$, $16T_H + 10T_{EXP}$, $12T_H + 10T_{CCM}$, $18T_H + 9T_{EXP}$, $15T_H + 6T_{EXP}$, $12T_H + 10T_{ECC}$, and $14T_H + 10T_{EXP}$, respectively. According to Chatterjee et al. [28] and Wazid et al. [29], the running times of different cryptographic operations are as follows, on average T_H is 0.0005 s, T_{ECC} is 0.063075 s, T_{CCM} is 0.02102 s, and T_{FE} is nearly 0.063075 s. In addition, T_{EXP} approximates $240T_H$ [33]. Hence, the result of computation complexity indicates that the SBAKE protocol is more efficient in contrast to 3WPAKE, Chen et al., Xie et al., Farash and Attari, Tallapally, Wu et al., Chang et al.'s protocols.

Table 4. Performance comparison for the login and authentication phase.

Protocol	Computation Cost			Total Cost	Message Exchange (Number/Byte)			
	U_A	U_B	S		U_A-S	U_B-S	U_A-U_B	Total
SBAKE	$7T_H + 2T_{CCM} + 1T_{FE}$	$7T_H + 2T_{CCM} + 1T_{FE}$	$7T_H$	$21T_H + 4T_{CCM} + 2T_{FE}$	2/120	2/120	2/40	6/280
3WPAKE [9]	$5T_H + 3T_{EXP}$	$5T_H + 3T_{EXP}$	$10T_H + 3T_{EXP}$	$20T_H + 9T_{EXP}$	4/488	4/488	2/56	10/1032
Chen et al. [20]	$5T_H + 3T_{EXP}$	$5T_H + 3T_{EXP}$	$6T_H + 4T_{EXP}$	$16T_H + 10T_{EXP}$	4/372	4/372	0/0	10/744
Xie et al. [44]	$3T_H + 3T_{CCM}$	$3T_H + 3T_{CCM}$	$6T_H + 4T_{CCM}$	$12T_H + 10T_{CCM}$	4/124	3/108	0/0	7/232
Farash and Attari [11]	$6T_H + 3T_{EXP}$	$6T_H + 3T_{EXP}$	$6T_H + 3T_{EXP}$	$18T_H + 9T_{EXP}$	4/440	4/440	2/40	10/920
Tallapally [19]	$5T_H + 2T_{EXP}$	$5T_H + 2T_{EXP}$	$5T_H + 2T_{EXP}$	$15T_H + 6T_{EXP}$	2/284	2/284	2/40	6/608
Wu et al. [45]	$4T_H + 3T_{ECC}$	$4T_H + 3T_{ECC}$	$4T_H + 4T_{ECC}$	$12T_H + 10T_{ECC}$	3/136	1/100	2/108	6/344
Chang et al. [46]	$5T_H + 3T_{EXP}$	$5T_H + 3T_{EXP}$	$4T_H + 4T_{EXP}$	$14T_H + 10T_{EXP}$	3/312	1/20	2/324	6/656

For the communication complexity comparison, we set the block size of one-way hash function $h_1(.)$, $h_2(.)$, Chebyshev chaotic map, and elliptic curve point are 20 bytes long, and the modular exponentiation to be 128 bytes long. As shown in Figure 3b, the SBAKE protocol exchanged the least number of messages and had a lower communication load than 3WPAKE, Chen et al. [20], Farash and Attari [11], Tallapally [19], Wu et al. [45], Chang et al. [46]'s protocol, but a slightly increased communication load compared to Xie et al.'s protocol [44] that might be acceptable for security purposes. In short, the SBAKE protocol outperforms when considering computation and communication complexity, as illustrated in Table 4 and Figure 3b.

5. Conclusions

The role of innovative technologies like AI and blockchain in the healthcare system is expected in the future, the system has to be uncompromised and complete. In this work, we analyzed the security of the 3WPAKE protocol and determined the protocol to be vulnerable to privileged insider attacks, to not preserve privacy, and to have inefficient authentication verification. Furthermore, we proposed a SBAKE that fixes the security flaws of 3WPAKE and substantially improves efficiency. Our security and performance comparison implies that the SBAKE protocol achieves both better security and higher efficiency for healthcare support deployment. This is a meaningful conclusion, proposing a SBAKE protocol that is more lightweight and secure than the former protocol.

Author Contributions: Conceptualization, M.K.; validation, D.W.; formal analysis, M.K. and J.M.; writing-original draft, review and editing, M.K.; supervision, N.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute for Information & communications Technology Promotion (IITP), grant funded by the Korea government (MSIT) [2019-0-00203, The Development of Predictive Visual Security Technology for Preemptive Threat Response]. This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2019S1A5C2A04083374).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bellovin, S.M.; Merritt, M. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, CA, USA, 4–6 May 1992; pp. 72–84. [CrossRef]
2. Khan, M.K.; Zhang, J. Improving the security of 'a flexible biometrics remote user authentication scheme'. *Comput. Standards Interfaces* **2007**, *29*, 82–85. [CrossRef]

3. Tseng, Y.M.; Yu, C.H.; Wu, T.Y. Towards scalable key management for secure multicast communication. *Inf. Technol. Control* **2012**, *41*, 173–182. [CrossRef]
4. Sun, H.; Wen, Q.; Zhang, H.; Jin, Z. A strongly secure pairing-free certificateless authenticated key agreement protocol for low-power devices. *Inf. Technol. Control* **2013**, *42*, 113–123. [CrossRef]
5. Jiang, Q.; Ma, J.; Li, G.; Ma, Z. An improved password-based remote user authentication protocol without smart cards. *Inf. Technol. Control* **2013**, *42*, 150–158. [CrossRef]
6. Roy, S.; Chatterjee, S.; Das, A.K.; Chattopadhyay, S.; Kumari, S.; Jo, M. Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing Internet of Things. *IEEE Internet Things J.* **2018**, *5*, 2884–2895. [CrossRef]
7. Al-Zubaidie, M.; Zhang, Z.; Zhang, J. LAMHU: A new lightweight robust scheme for mutual users authentication in healthcare applications. *Secur. Commun Netw.* **2019**. [CrossRef]
8. Lee, T.F.; Liu, J.L.; Sung, M.J.; Yang, S.B.; Chen, C.M. Communication-efficient three-party protocols for authentication and key agreement. *Comput. Math. Appl.* **2009**, *58*, 641–648. [CrossRef]
9. Lu, Y.; Li, L.; Peng, H.; Yang, Y. A three-party password-based authenticated key exchange protocol for wireless communications. *Inf. Technol. Control* **2015**, *44*, 404–409. [CrossRef]
10. Jeon, W.; Kim, J.; Nam, J.; Lee, Y.; Won, D. An enhanced secure authentication scheme with anonymity for wireless environments. *IEICE Trans. Commun.* **2012**, *95*, 2505–2508. [CrossRef]
11. Farash, M.S.; Attari, M.A. An enhanced and secure three-party password-based authenticated key exchange protocol without using server's public-keys and symmetric cryptosystems. *Inf. Technol. Control* **2014**, *43*, 143–150. [CrossRef]
12. Liu, T.; Pu, Q.; Zhao, Y.; Wu, S. ECC-based password-authenticated key exchange in the three-party setting. *Arab. J. Sci. Eng.* **2013**, *38*, 2069–2077. [CrossRef]
13. Chien, H.Y.; Wu, T.C. Provably secure password-based three-party key exchange with optimal message steps. *Comput. J.* **2009**, *52*, 646–655. [CrossRef]
14. Guo, C.; Chang, C.C. Chaotic maps-based password-authenticated key agreement using smart cards. *Commun. Nonlinear Sci. Numer. Simul.* **2013**, *18*, 1433–1440. [CrossRef]
15. Yau, W.; Phan, R. Cryptanalysis of a chaotic map-based password-authenticated key agreement protocol using smart cards. *Nonlinear Dyn.* **2015**, *79*, 809–821. [CrossRef]
16. Stallings, W. *Cryptography and Network Security: Principles and Practices*, 4th ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2006.
17. Huang, H.F. A simple three-party password-based key exchange protocol. *Int. J. Commun. Syst.* **2009**, *22*, 857–862. [CrossRef]
18. Yoon, E.J.; Yoo, K.Y. Cryptanalysis of a simple three-party password-based key exchange protocol. *Int. J. Commun. Syst.* **2011**, *24*, 532–542. [CrossRef]
19. Tallapally, S. Security enhancement on simple three party PAKE protocol. *Inf. Technol. Control* **2012**, *41*, 15–22. [CrossRef]
20. Chen, C.M.; Wang, K.H.; Yeh, K.H.; Xiang, B.; Wu, T.Y. Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications. *J. Ambient Intell. Hum. Comput.* **2019**, *10*, 3133–3142. [CrossRef]
21. Zhang, L.P.; Zhu, S.H.; Tang, S. Privacy protection for telecare medicine information system using a chaotic map-based three-factor authenticated key agreement scheme. *IEEE J. Biomed. Health Inform.* **2015**, 2168–2194. [CrossRef]
22. Lee, T.F. Efficient and secure temporal credential-based authenticated key agreement using extended chaotic maps for wireless sensor networks. *Sensors* **2015**, *15*, 14960–14980. [CrossRef]
23. Renuka, K.; Kumar, S.; Kumari, S.; Chen, C.M. Cryptanalysis and improvement of a privacy-preserving three-factor authentication protocol for wireless sensor networks. *Sensors* **2019**, *19*, 4625. [CrossRef]
24. He, D.; Kumar, N.; Lee, J.H.; Sherratt, R. Enhanced three-factor security protocol for consumer USB mass storage devices. *IEEE Trans. Consum. Electron.* **2014**, *60*, 30–37. [CrossRef]
25. Ravanbakhsh, N.; Nazari, M. An efficient improvement remote user mutual authentication and session key agreement scheme for E-health care system. *Multimed. Tools Appl.* **2016**, 1–34. [CrossRef]
26. Lee, C.C.; Hsu, C.W. A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps. *Nonlinear Dyn.* **2013**, *71*, 201–211. [CrossRef]

27. Mishra, D.; Das, A.K.; Mukhopadhyay, S. A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. *Expert Syst. Appl.* **2014**, *41*, 8129–8143. [CrossRef]
28. Chatterjee, S.; Roy, S.; Das, A.K.; Chattopadhyay, S.; Kumar, N. Secure biometric-based authentication scheme using Chebyshev chaotic map for multi-server environment. *IEEE Trans. Dependable Secur. Comput.* **2016**. [CrossRef]
29. Wazid, M.; Das, A.K.; Kumari, S.; Li, X.; Wu, F. Design of an efficient and provably secure anonymity preserving three-factor user authentication and key agreement scheme for TMIS. *Secur. Commun. Netw.* **2016**, *9*, 1983–2001. [CrossRef]
30. Wei, F.; Ma, J.; Jiang, Q.; Shen, J.; Ma, C. Cryptanalysis and improvement of an enhanced two-factor user authentication scheme in wireless sensor networks. *Inf. Technol. Control* **2016**, *45*, 62–70. [CrossRef]
31. Dodis, Y.; Kanukurthi, B.; Katz, J.; Rezin, L.; Smith, A. Robust fuzzy extractors and Authenticated key agreement from close secrets. *IEEE Trans. Dependable Secur. Comput.* **2012**, *58*, 6207–6222. [CrossRef]
32. Kuo, W.; Lin, C.; Chuang, C.; Kao, M. Simultaneous and anonymous mobile network authentication scheme based on chaotic maps. *Inf. Technol. Control* **2016**, *45*, 208–213. [CrossRef]
33. Wang, W.; Cheng, Q.A. Multi-party secret handshake scheme based on chaotic maps. *Inf. Technol. Control* **2017**, *46*, 138–149. [CrossRef]
34. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *Adv. Cryptol. (Eurocrypt)* **2004**, *LNCS 3027*, 523–540.
35. Zhang, L. Cryptanalysis of the public key encryption based on multiple chaotic systems. *Chaos Solitons Fractals* **2008**, *37*, 669–674. [CrossRef]
36. Xu, D.; He, D.; Choo, K.K.R.; Chen, J. Provably secure three-party password authenticated key exchange protocol based on ring learning with error. In *Cryptographers' Track at the RSA Conference*; Springer: Cham, Switzerland, 2017; p. 360.
37. Chen, C.; Fang, W.; Liu, S.; Wu, T.; Pan, J.; Wang, K. Improvement on a chaotic map-based mutual anonymous authentication protocol. *J. Inf. Sci. Eng.* **2016**, *34*, 371–390.
38. Bellare, M.; Rogaway, P. Entity authentication and key distribution. In *Advances in Cryptology—CRYPTO'93*; Lecture Notes in Computer Science; Stinson, D.R., Ed.; Springer: Berlin/Heidelberg, Germany, 1993; Volume 773, pp. 232–249.
39. Gollman, D. *Insider Fraud. Security Protocols-6th International Workshop*; Springer Science & Business Media: Cambridge, UK, 1998; Volume 1550, pp. 220–226.
40. Moon, J.; Lee, Y.; Kim, J.; Won, D. Improving an anonymous and provably secure authentication protocol for a mobile user. *Secur. Commun. Netw.* **2017**. [CrossRef]
41. Das, A.K. A secure and effective user authentication and privacy preserving protocol with smart cards for wireless communications. *Netw. Sci.* **2013**, *2*, 12–27. [CrossRef]
42. Das, A.K.; Paul, L.N.; Tripathy, R. Cryptanalysis and improvement of an access control in user hierarchy based on elliptic curve cryptosystem. *Inf. Sci.* **2012**, *209*, 80–92. [CrossRef]
43. Jung, J.; Moon, J.; Lee, D.; Won, D. Efficient and security enhanced anonymous authentication with key agreement scheme in wireless sensor networks. *Sensors* **2017**, *17*, 644. [CrossRef]
44. Xie, Q.; Hu, B.; Wu, T. Improvement of a chaotic maps-based three-party password-authenticated key exchange protocol without using servers public key and smart card. *Nonlinear Dyn.* **2015**. [CrossRef]
45. Wu, S.; Pu, Q.; Wang, S.; He, D. Cryptanalysis of a communication-efficient three-party password authenticated key exchange protocol. *Inf. Sci.* **2012**, *215*, 83–96. [CrossRef]
46. Chang, T.Y.; Hwang, M.S.; Yang, W.P. A Communication-Efficient Three-Party Password Authenticated Key Exchange Protocol. *Inf. Sci.* **2011**, *181*, 217–226. [CrossRef]