The Resilience of DoS Attacks in User-Authentication to Preserving The Availability

E Haodudin Nurkifli

Abstract—User authentication is imperative to ensure legitimate users—The majority of the existing user authentications based on identity and password. Nowadays, user authentications that are based on biometric and password have been applied widely. Unfortunately, most existing user authentications are vulnerable to various attacks, including online/offline password guessing attacks, stolen device/smart card attacks, user masquerades attacks, server spoofing attacks, and DoS attacks. Additionally, most of them did not consider solving the loss of the synchronization problem. On the other hand, availability is essential in authentication protocol; authentication protocol must still operate even if loss of synchronization occurs. This article is proposed to resolve user authentication with the main security feature achievement, namely, availability, resolve the loss of synchronization, and withstand DoS attack. Informal analysis is used to ensure that our authentication protocol achieves security features, and we also evaluate our proposal formally using BAN Logic, Real-or-Random Model, and Scyther tool—comparison of computational complexity to ensure that our protocol obtains low computational complexity.

Index Terms—User Authentication, Biometric, PUF, informal and formal analyses, BAN Logic, ROR model, Scyther tool, computational complexity.

1 INTRODUCTION

USER authentication is essential to ensure a legitimate user. Three crucial phases must be done if the user wants to access information systems. The user must register initially and then perform Login, mutual authentication between the user’s device and server. Wireless-based communication between a user’s device and the server enables the attacker to intercept, alter, and delete the message. Additionally, the attacker can perform a severe attack. The attacker steals the device and obtains the data from the device’s memory, then the attacker conducts a severe attack, such as online/offline password guessing attack, masquerade attack, spoofing attack, or tracking attack. Several schemes have been proposed to resolve the issues in user authentication. [1] proposed user authentication based on identity and password. [2] and [3] proposed a user authentication based on a smart card, identity, and password. However, most of the existing schemes fail to withstand various attacks, such as online/offline guessing attacks, masquerade attack, spoofing attack, tracking attack [4], [5].

The other researchers, [4], [5], [14], [6]–[13] proposed user authentication using passwords, biometric, and smart cards. Unfortunately, [4], [5], [14], [6]–[13] have a similar problem with [2] and [3]. Further, based on our investigation that most of them did not consider resolving the loss of the synchronization problem. It is very dangerous if the scheme is applied to the IoT-based health care system. For instance, the doctor needs the data of a critical patient from the server immediately but a loss of synchronization occurs and the protocol cannot handle that problem. Therefore, the doctor cannot obtain the data due to this problem. It can be said the existing schemes cannot achieve availability. We define availability at the user authentication protocol, namely, the user authentication protocol is still operating even if the loss of synchronization occurs. Moreover, due to the previous user authentication protocols did not consider to achieve unclonable device, hence, the previous user authentication protocols cannot guarantee to withstand cloning attacks.

[10]–[12] introduce Physical Unclonable Function (PUF) to be applied in a limited resource and withstand cloning attack. A PUF is a unique property of a circuit that maps a Challenge C to the Response R in the chip manufacturing process; internal structure provides a one-way function; hence PUF cannot be duplicated and almost impossible to clone. Recently, Weixin Bian et al. [15] proposed the user authentication protocol using biometric and PUF-based. However, based on our investigation that Weixin Bian et al.’s scheme [15] fails to achieve availability, resolve the loss of synchronization problems and cannot withstand the DoS attack.

In this paper, we propose a new user authentication protocol using biometric and PUF. The proposed protocol can achieve various security features, mutual authentication, anonymity, untraceability, resolve the loss of synchronization problem, and availability. Further, the proposed protocol withstands kinds of attacks such as DoS Attack, tracking attack, and cloning attack (see Table 3).

The rest of this paper is organized as follows. Section II presents the detailed proposal and is followed by informal...
analysis in Section III, Section IV presents performance analysis, and comparison followed by formal analysis using BAN Logic, RoR Model, Syther tool in Section V, VI, VII, respectively and finally conclusion in Section VI.

1.1 Relater Works and Motivation
Several researchers have proposed user authentication. One example is a traditional user authentication based on identity and password, proposed by Lamport [1], which must store the password’s table of the user to verify the legitimate user. Unfortunately, [1] suffered an offline password guessing attack. Hwang et al. [2] proposed user authentication in smart cards but [2] did not store the password in the table. Nonetheless, [3] pointed out that [2] suffer against forged identity attacks. Lee et al. [14] tried to resolve the problem of [2] in the smart card using biometric. Unfortunately, [16] showed that [14] fails to withstand masquerade attacks. Additionally, [14] and [16] did not provide mutual authentication; only consider user authentication; hence [14] and [16] suffer server spoofing attack. [17] then proposed a new authentication protocol to improve security features in [16]. Unfortunately, [17] fails to withstand password guessing attack and masquerade attack.

Another researcher, Khan et al. [4] proposed a new user authentication to improve security features based on biometric and a one-way hash function. Unfortunately, Khan et al.’s proposed authentication protocol fails to withstand the password guessing attack and the masquerade attack after the attacker extracts data from the device’s memory. Chen et al. [5] proposed the scheme using biometric and password. However, their scheme is vulnerable to password guessing attacks. Troung et al. [6] proposed user authentication to improve security features including anonymity and withstand spoofing attack and reply attack. However, Troung et al.’s scheme [6] has a loophole; namely, the user can reveal the server’s secret key. Khan et al. [7] proposed the scheme to enhance security features. Unfortunately, Khan et al.’s scheme [7] is as vulnerable as the schemes proposed by [5] and [6] to malicious attacks.

Several researchers offered the smart card to resolve the security problems in user authentication. One of them is Poh et al. [18] who proposed user authentication in a smart home. However, Poh et al. [18] fail to the verified users in the login phase. Later, Das [8] proposed a scheme in user authentication which eventually fails to withstand external attacks. An [9] offered the scheme to improve Das’s scheme [8]. However, Khan et al. [10] and Ibjaoun [11] point out that An [9] fails to achieve security features similar to Das [8]. Li et al. [12] proposed a scheme to improve Das’s scheme [8] by administering mutual authentication and establishing a session key between user and server. However, Chaturvedi et al. [13] point out that Li et al. [12] cannot achieve user anonymity. Therefore, Chaturvedi et al. [13] proposed a new scheme to resolve user authentication by providing mutual authentication and establishing a session key. However, [13] fails to withstand the cloning attack of the device.

Recently, many researchers proposed user authentication based on biometric which is different from traditional password-based user authentication. It gives many advantages based on unique individual biometric input [18]. Additionally, it prevents the user from remembering and losing passwords. The other researchers, [19]–[22] use biometric for user authentication. The biometric should not store in the user device and server directly to mitigate risk [14], [16], [17], [4]. Meanwhile [5], [6], [7], [8] proposed biometric-based user authentication using a one-way hash function to protect biometric before stored. The fact that biometric have a noisy output and will change over time makes the device not recognize the user in the future because of noisy output and biometric changes. To resolve this problem, a fuzzy extractor is used to generate a fixed key, help data from biometric input, where (K, hd)FE.Gen(Biometric) and then it can perform reconstruction to obtain a similar fixed key, where (K)=FE.Rec(Biometric, hd) [23], [24]. However, if only Biometric is used, it cannot guarantee to withstand cloning attack. [10]–[12] introduce Physical Uncloneable Function (PUF) to be applied in a limited resource and to withstand a cloning attack. A PUF is a unique property of a circuit that maps a Challenge C to the Response R in the chip manufacturing process; internal structure provides a one-way function; hence PUF cannot be duplicated and is almost impossible to clone.

Recently, Weixin Bian et al. [15] proposed a user authentication protocol using biometric and PUF-based. However, Weixin Bian et al.’s scheme cannot resolve the loss of synchronization problems and withstands the DoS attack. It is very dangerous if the scheme is applied in the IoT-based health care system. For example, the doctor needs critical data patients from the server immediately but the protocol cannot operate because of loss of synchronization; hence the doctor cannot obtain the data.

This article offers the solution to the user authentication problems, which has not been achieved by previous user authentication proposed by [16], [5], [6], [7], [18], [8], [9], [10], [12], [13], and [15]. Contributions to this article as follows:

1. Designing user authentication using PUF and Biometric, our proposal can achieve security features of mutual authentication, anonymity, untraceability, perfect forward, session key security, resolve the loss of synchronization problem, and availability.
2. Our scheme also has the ability to withstand various attack, such as online/offline password guessing attack, stolen mobile device/smart card, reply attack, user masquerade attack, server spoofing attack, man-
in-the-middle attack, known session key attack, cloning attack, and DoS attack.

3. A formal analysis using BAN Logic was carried out to ensure that our user authentication protocol achieves secure mutual authentication.

4. The RoR model and Scyther tool are used as also analysis formally to ensure that our proposal withstands various attacks.

5. Comparison, in terms of security features and computational complexity, to ensure that our proposal is not only secure but also has low computational complexity.

1.2 Preliminaries

This subsection presents briefly the preliminary background of the Fuzzy Extractor and PUF

a. Fuzzy Extractor.

The fuzzy extractor is a function that has two processes, namely \( FE.Gen() \) and \( FE.Rec() \). \( FE.Gen() \) is the process to generate the fixed key \( K_u \) and helper data \( hd \) from the input Biometric \( B_u \), where \( (K_u,hd) = FE.Gen(B_u) \). \( FE.Rec() \) is the process to reconstruct \( K_u \) from helper data \( hd \) and noisy input \( B_u' \) where \( B_u \) is biometric of the user and \( B_u' \) is an approximation of \( B_u \) from a harmaning distance, where \( dis(B_u,B_u') \leq t \), and \( t \) is a threshold. Therefore, \( (K_u,hd) = FE.Gen(B_u) \Rightarrow K_u = FE.Rec(hd,B_u') \). The successful \( FE \) is based on the similarity of original data and noisy data. Therefore, in this paper, we use the fuzzy extractor to obtain a fixed key, namely \( K_u \) from Biometric \( B_u \), where \( (K_u,hd) = FE.Gen(B_u) \).

b. Physically Unclonable Function (PUF)

A PUF is a unique property of a circuit that maps a Challenge \( C \) to the Response \( R \) in the chip manufacturing process [25]. Formally, input \( C \) into PUF to produce \( R \) where \( R=PUF(C) \). Significantly, PUFs are hard to clone [26]. We divide PUF into two types, namely non-ideal PUF and ideal PUF. The first is a non-ideal PUF, where PUF may produce a different response with similar input Challenge \( C \) into PUF due to temperature. We can use the fuzzy extractor to ensure the stable PUF output [23], [24]; however, a fuzzy extractor increases computational overhead. The second is an ideal PUF, where PUF will produce a similar response with similar challenges inputted into PUF. Even if the non-stable temperature occurs, in the last few years, researchers have developed ideal PUF, ensuring 0% Bit-Error Rate [27]-[30]. Therefore, in this paper, we use the ideal PUF.

### Table 1: Notations Of Cryptography Functions

| Notation | Definition |
|----------|------------|
| \( ID_u \) | Identity of the User |
| \( PUF_d \) | Physically unclonable function of user’s device |
| \( PID_{us} \) | One-Time pseudonym to achieve anonymity |
| \( CRP(C_u,R_u) \) | Challenge-Response Pairs |
| \( CRP(C_{syn},R_{syn}) \) | Synchronization (Challenge-Response Pairs) \( (C_{syn},R_{syn}) = \{(c_1,r_1),(c_2,r_2),..., (c_n,r_n)\} \) |
| \( (PID_{syn},K_{syn}) \) | Synchronization [pseudonym, secret key pairs] \( (PID_{syn},K_{syn}) = \{(pid_1,k_1),(pid_2,k_2),..., (pid_n,k_n)\} \) |
| \( K_d \) | Private key of server / Secret key (Long term key) |
| \( e_u \) | A user unique random number generated by server |
| \( || \) | Concatenation operation |
| \( \oplus \) | Exclusive-OR operation |
| \( SK \) | Session key between user and server |
| \( B_{IO_u} \) | Biometric of User |
| \( FE \) | Fuzzy extractor for Biometric |
| \( K_u,hd \) | Fixed key \( K_u \) and help data \( hd \) are generated from \( B_u \) where \( (K_u,hd) = FE.Gen(B_{IO_u}) \) |

2 Proposed Scheme

This section presents our proposed scheme starting from System structure, Assumptions, Notation, user registration phase, login, mutual authentication and session key establishment between device and server. For convenience, we use the notation in Table 1 throughout our scheme.

2.1 System Structure

In Fig. 1, the system structure consists of two components device and server; the device may be a mobile phone, laptop, or iPad, and the server stores user data. User login
to the device and the device verifies user based on biometric, if the biometric input is not equal to the biometric stored in the device, the login will be terminated. Otherwise, the protocol performs mutual authentication and establishes a session key between device and server; after the establishment of the session key, devices and servers can communicate securely.

Our scheme has two phases, namely the user’s device registration (Figure 2), login, mutual authentication, and session key establishment between the user’s device and server (Figure 3).

2.2 Assumptions

We have several assumptions as follows:

a. The devices are equipped with biometric input and PUF.

b. The devices have constrained resources.

c. The server has no constrained resources.

2.3 Notations of Cryptography Function

This subsection presents the notations of the cryptography function throughout this paper. Additionally, based on the assumption aforesaid, the devices are equipped with biometric and PUF; hence our scheme uses this facility (see Table 1).

2.4 User’s device registration

The user’s device registration has 4 steps as follows:

Step 1: the user sends identity ID_u, registration request Reg_uq to the server.

Step 2: the server generates challenge C_u, one-time pseudonym between user and server PID_u, synchronization [challenge, pseudonym, key] where C_syn = {c_1, c_2, ..., c_n}, PID_syn = {pid_1, pid_2, ..., pid_n}, K_syn = {k_1, k_2, ..., k_n}. The server sends C_u, C_syn, PID_u, PID_syn, K_syn to the user.

Step 3: upon receiving C_u, C_syn, PID_u, PID_syn, K_syn, the user generates R_u = PUF(C_u), R_syn = PUF(C_syn). The user imprints biometric BIO_u generates (K_u, hd) = FE.Gen(BIO_u), computes authentication code $V = h(hd||K_u||ID_u)$, computes $W = h(ID_u||K_u)$, and sends R_u, R_syn, W to the server. Subsequently, the device protects several tuples using $K_u$, PID_us = PID_us $\oplus$ h(K_u), PID_syn = PID_syn $\oplus$ h(K_u), K_syn = K_syn $\oplus$ h(K_u), ID_syn = $\Pi$pid_i $\oplus$ h(K_u), and the user stores V, hd, ID_syn, PID_syn, K_syn.

Step 4: upon receiving R_u, R_syn, W, the server generates long term key K_s, unique number of user e_u, computes $E_i = h(K_s||e_u)\oplus W$ and the server sends E_i to the user. Finally, the server stores (C_u, R_u), (C_syn, R_syn), (PID_syn, K_syn), ID_syn, PID_syn, e_u, K_s.

Step 5: upon the receiving E_i, the user computes $E_i^* = E_i \oplus h(K_u)$ and finally, the user stores V, hd, ID_syn, K_syn, PID_syn, PID_u, E_i.

On normal conditions, the device using PID_us is replaced by new values [PID_new] given by the server. When desynchronization occurs, the device uses (pseudo-identity and shared-key synchronization) pairs, that is {(pid_1, k_1), ..., (pid_n, k_n)} $\in$ (PID_syn, K_syn).

2.5 Login, Mutual Authentication and Session Key Establishment between Device and Server

This subsection presents the notations of the cryptography function throughout this paper. Additionally, based on the assumption aforesaid, the devices are equipped with biometric and PUF; hence our scheme uses this facility.

Login, Mutual authentication, and session key establishment between device and server has 4 steps as follows:

Step 1: user login to device, user imprints biometric BIO_u, the device extracts $K_u = \text{FE.Req}(BIO_u, hd)$, computes $\Pi$pid_i $\oplus$ h(K_u), $V^* = h(hd||K_u||ID_u)$, and verifies $V^* = ? V$, if $V^*$ is not equal to V the login fails. Otherwise, the device computes $\Pi$pid_i $\oplus$ h(K_u), $E_i = E_i \oplus h(K_u)$, $W = h(ID_u||K_u)$, generates $N_u$, computes $K = E_i \oplus W, N_u^* = N_u \oplus K$, computes verification code $V_0 = h(K||\Pi$pid_i$|N_u)$, and the user sends $M_i = \Pi$pid_i$|N_u^*|V_0$ to the server.

Step 2: upon receiving $M_i$, the server computes $N_u = N_u^* \oplus h(K_u||e_u)$, checks $\Pi$pid_i, reads challenge-response pairs (C_u, R_u), verifies $V_0$, and if verification fails,
the communication will be terminated. Otherwise, the server generates randomly \( N'_s \), computes \( N'_s = N_0 \oplus h(K_0)[e_u] \), \( C_u = C_u \oplus h(N_u)[N_u] \), the server decides to update pseudonym and protects using \( R_u \) where \( PID'_u = PID^new_u \oplus h(ID_u)[R_u] \), computes verification code \( V'_u = h(h(K_e_u)[e_u])\|ID_u[N_u][C_u][PID^new_u][N'_u][N'_s] \) and sends \( M_3: \{N'_u, C_u, PID'_u, V'_u \} \) to the user.

**Step 3:** Upon receiving \( M_3: \{N'_u, C_u, PID'_u, V'_u \} \), the user computes \( N_1 = N'_u \oplus K \), verifies \( V_1 \). Computes \( C_u = C_u \oplus h(N_u)[N_u] \), generates \( R_u = PUF(C_u) \), update challenge-response \( C^new_u = h(C_u)[R_u] \), \( R^new_u = PUF(C^new_u) \), computes \( N_1 = h(R_u)[N_1] \oplus R^new_u \), \( SK = h(R_u)[N_1][N_u] \), \( V_2 = h(ID_u)[C^new_u][R^new_u][N'_u][N'_s] \). Computes \( PID^new_u = PID^new_u \oplus h(ID_u)[R_u] \), protects new pseudonym with \( K_u \) where \( PID^new_u = PID^new_u \oplus h(K_u) \), and the user sends \( M_3: \{V_2, N_1 \} \). Finally, the user stores \( PID^new_u \).

**Step 4:** Upon receiving \( M_3: \{V_2, N_1 \} \), the server computes \( R^new_u = h(R_u)[N'_u] \oplus N_1 \), \( C^new_u = h(C_u)[R_u] \), verifies \( V_2 \). The server computes session key \( SK = h(R_u)[N_1][N_u] \). Finally, the server stores \( PID^new_u \), \( (C^new_u, R^new_u) \).

### 2.6 The Solution to Resolve the Loss of Synchronization and DoS Attack

The solution to resolve the loss of synchronization as follows:

**Step 1:** When desynchronization occurs, the device uses (pseudo-identity and shared-key synchronization) pairs, that is \( \{( pid_1, k_1), \ldots, (pid_n, k_n) \} \in (PID_{syn}, K_{syn}) \).

**Step 2:** Run the same step with the authentication phase.

### 3 Analysis Informally and Comparison

This section presents the achievement of security features as well as a comparison of capabilities to withstand kinds of attacks with existing the scheme proposed by [16], [5], [6], [7], [18], [8], [9], [10], [12], [13], and [15].

### 3.1 Security Features Analysis

This subsection presents the security feature analysis. The details are as follows:

a. Mutual Authentication

Both parties recognize the identity of the other party by the possession of a \( h(K_e_u)[e_u] \) and \( R_u \). The device checks \( N'_s \), and the server checks \( N_u \) to ensure the freshness. Therefore, the scheme achieves mutual authentication.

b. Anonymity

Our protocol uses a one-time pseudonym \( PID^new \). Even if the attacker intercepts all messages in public channels including \( PID^new \), the attacker cannot get the user’s real identity. Therefore, our protocol achieves anonymity.

c. Untraceability

Since our protocol achieves anonymity, it can be said that the attacker finds it hard to track the legitimate user. Additionally, the challenge \( C_u \) is protected by nonce pairs \( \{N_u, N'_u \} \) so the challenge is not public, the attacker cannot know the owner of the challenge based on the collected message in the public channel. Therefore, the scheme achieves untraceability.

d. Perfect forward secrecy.

In our scheme, the message updates in every session include Nonce, challenge \( C^new_u \), and response \( R^new_u \), session key. There is no relationship between them. Even if the attacker obtains the current session key, the attacker cannot obtain past session key. Therefore, the scheme achieves perfect forward.

e. Session key security

In our protocol, the session key changes every session and there is no relation between them. Even if the attacker obtains the session key, the attacker cannot obtain past and future session key. Therefore, our authentication protocol achieves session key security.

f. Resolve the desynchronization problem
If loss of synchronization occurs, the device replaces \( PID_{us} \) by \( pid_1 \) from \( PID_{syn} = \{ pid_1, pid_2, ..., pid_n \} \) deletes \( pid_1 \) in \( PID_{syn} \), the server replaces \( C_u \) by \( c_1 \) from \( C_{syn} = \{ c_1, c_2, ..., c_n \} \), replaces Nonce by \( k_{syn} = \{ k_1, k_2, ..., k_n \} \), and runs the same step with the authentication phase. Therefore, the scheme can provide resolve for the loss of synchronization.

g. Availability
Since our protocol can resolve the loss of synchronization problem, the protocol is still operating even if the synchronization is lost. Therefore, our protocol achieves availability.

3.2 Attack Analysis
This subsection presents the attack analysis to ensure that our protocol withstands various attacks. The details are as follows:

a. Withstanding online/offline password guessing attack
Our user authentication protocol does not need a password, the protocol only use \( K_u \) to protect the data stored in the device. Therefore, the attacker never has the opportunity to guess the password and our protocol does not have any risks of guessing the password as well.

b. Withstanding stolen device smart card
If the attacker has stolen the mobile device and extracted data from the memory \( V, hd, ID_{us}, PID_{syn}, PID_{us}^*, E_i^* \), since the data are protected by \( K_u \), where \( K_u \) extract from biometric, the attacker is hard to obtain the original real data. Therefore, our protocol withstands stolen mobile device.

c. Withstanding replay attack
In our scheme, message updates in every session e.g. \( M_1(PID_{us}, N_u^*, V_0) \). If the attacker intercepts message in public channel e.g. \( M_1(PID_{us}, N_u^*, V_0) \) and resends \( M_1(PID_{us}, N_u^*, V_0) \). Due to the updated server to new \( PID_{new}^* \) in step 2; the server will be easy to detect based on \( PID_{sh}^* \), new nonce. Therefore, our scheme can withstand replay attack.

d. Withstands user masquerade attack
If a legitimate user \( U_i \) attempts to masquerade other user \( U_a \). The user \( U_i \) has \( V, hd, ID_{us}^*, PID_{syn}^*, PID_{us}^*, E_i \), Even if \( U_i \) obtains \( V, hd, ID_{us}^*, PID_{syn}^*, PID_{us}^*, E_i \), the \( U_i \) cannot impersonate to login as legitimate user \( U_a \). Therefore, our proposal withstands masquerade attack.

e. Withstanding server spoofing attack
If user \( U_a \) attempts to masquerade as server to spoof the other user \( U_i \), \( U_a \) must generate valid tuples \( \{ N_u^*, C_u, V_i \} \) where \( N_u^* = N_e \oplus h(K_u^e) | e_u ), \( C_u = C_u \oplus h(N_u | N_u_e) \), \( V_i = h(K_u^e) | PID_{us}^* | N_u^* | N_u_e | \{ PID_{syn}^* \} | \{ N_u \} \). However, \( U_a \) is impossible to obtain \( K_u^e, e_u \). Therefore, our proposal withstands server spoofing attack.

f. Withstanding MIM attack.
The attacker intercepts \( M_1(PID_{us}, N_u^*, V_0) \) and modifies to be \( M_1(PID_{us}, N_u^*, V_0) \). Subsequently, the attacker sends \( M_1(PID_{us}, N_u^*, V_0) \). The server will be easy to detect based on \( PID_{us}^*, N_u \). Therefore, our scheme can withstand MIM attack.

g. Withstanding known session key attack
Since session key security has been achieved, even if the attacker knows one session key, the attacker cannot obtain past and future session key. Therefore, our scheme can withstand session key attack.

h. Withstanding DoS attack.
Based on the achievement of resolve for loss of synchronization problem, our scheme can withstand DoS attack.

i. Withstanding Cloning Attack
Our scheme uses PUF, therefore, our scheme can guarantee to withstand cloning attacks.

3.3 Comparison
This section presents a comparison between our proposed scheme with the previous protocol authentication by [16], [5], [6], [7], [18], [8], [9], [10], [12], [13], and [15] in terms of security features and the capability to resist attack. The comparison is below:

| Schemes               | SF1 | SF2 | SF3 | SF4 | SF5 | SF6 | SF7 |
|-----------------------|-----|-----|-----|-----|-----|-----|-----|
| C.H. Lin and Y.Y. Lay[16] | No  | No  | No  | Yes | Yes | No  | No  |
| C.L. Chen et al. [5]    | Yes | No  | No  | N/A | No  | No  | No  |
| T.T. Truong et al. [6]  | Yes | Yes | Yes | Yes | N/A | No  | No  |
| M.K. Khan et al. [7]    | Yes | Yes | Yes | Yes | Yes | No  | No  |
| G.-S. Poh et al. [18]   | No  | No  | No  | Yes | No  | No  | No  |
| A.K. Das [8]            | No  | No  | No  | N/A | No  | No  | No  |
| Y.H. An [9]             | No  | No  | No  | Yes | No  | No  | No  |
| M.K. Khan and S. Kumari [10] | Yes | Yes | Yes | N/A | No  | No  | No  |
| X. Li et al. [12]       | Yes | Yes | Yes | Yes | Yes | No  | No  |
| A. Chaturvedi et al. [13] | Yes | Yes | Yes | Yes | Yes | No  | No  |
| Weixin Bian et al. [15] | Yes | Yes | Yes | Yes | Yes | No  | No  |
| Our propose             | Yes | Yes | Yes | Yes | Yes | Yes | Yes |

The details of notations are SF: Security Features, SF1: Mutual authentication, SF2: Anonymity, SF3: Untraceability, SF4: Perfect forward secrecy, SF5: Session key security, SF6: resolve desynchronization problem, SF7: availability.
achieve SF1-SF3, SF5. However, [6] fail to achieve SF6-SF7 and do not provide SF4. M.K. Khan et al. [7] achieve SF1-SF5. However, [7] cannot achieve SF6-SF7. G.-S. Poh et al. [18] only achieve SF4. A.K. Das [8] cannot achieve all security features. Y.H. An [9] only achieve SF4. M.K. Khan and S. Kumari [10] achieve SF1-SF3. However, [10] do not achieve SF5-SF7 and do not provide SF4. X. Li et al. [12] achieve SF1, SF4, SF5. However, [12] fail to achieve SF6-SF7. A. Chaturvedi et al. [13] and Weixin Bian et al. [15] achieve SF1-SF5. However, [13] and [15] cannot achieve SF6-SF7. Our protocol achieves all security features SF1-SF7, only our protocol fulfills security features.

Based on Table 3, C.H. Lin and Y.Y. Lay [16] can withstand WA1, WA3-WA5, WA7-WA8. However, [16] cannot withstand WA6, WA9, WA10, and also their scheme do not provide WA2. C.L. Chen et al. [5] withstand WA3, WA7 and WA8. However, [5] fail to withstand WA1, WA2, WA4-WA6, WA9, WA10. T.T. Truong et al. [6] withstand WA4, WA7, WA8. However, [6] cannot withstand WA1-WA3, WA5, WA6, WA9, WA10. M.K. Khan et al. [7] withstand WA2-WA8. However, [7] cannot withstand WA9, WA10. G.-S. Poh et al. [18] only withstand WA8. A.K. Das [8] can withstand attack WA4, WA7, WA8. However, [8] fail to withstand attack WA1-WA3, WA5, WA6, WA9, WA10. Y.H. An [14] only withstand attack WA3. M.K. Khan and S. Kumari [10] can withstand attack WA1, WA2, WA5, WA6. However, [10] cannot withstand attack WA9, WA10 and don’t provide WA3, WA4, WA7, WA8. X. Li et al. [12] can withstand attack WA2, WA3, WA5-WA7. However, [12] fail to withstand attack WA4, WA9, WA10 and do not provide WA1, A. Chaturvedi et al. [13] withstand attack WA2, WA4, WA5, WA8. However, [13] cannot withstand attack WA9, WA10 and do not provide WA1, WA3, WA6, WA7, and Weixin Bian et al. [15] withstand attack WA1-WA8, WA10 but fail to withstand WA9. Our protocol can withstand all kinds of attacks (WA1-WA10).

### 3.3 Performance Analysis and Comparison

The subsection presents a comparison between our proposed scheme with the previous protocol authentication by [16], [5], [6], [7], [18], [8], [9], [10], [12], [13], and [15] in terms of execution complexity and execution time. The comparison is below:

The computational time is the time of simulation result from several cryptography libraries in JAVA programming which are the previous research conducted by [15]. PUF’s execution time [31], BCH fuzzy extractor operations (FE.Gen and FE.Rec) [23], bilinear pairing based on Pairing Based Cryptography (PBC) library [32], and Symmetric-key cryptosystem AES-CBC based on Java Cryptography Architecture (JCA) library [33]. The execution time for different cryptographic algorithm with ours refer to the simulation using JCA [33] and PBC [32] in the scheme [34], [35], [36], [37].

![Diagram of computational time comparison](image)

**Fig. 4. Diagram of computational time comparison**

The execution time of cryptography operation will be explained in detail as follows. The execution time of $T_{PUF}$ in the device is 0.12 ms, $T_{FE.Gen}$ in the device is 1.67 ms, and the server has 1.17 ms, $T_{FE.Rec}$ in the device is 3.28 ms and

### Table 3

Comparison of Attack Resistance

| Schemes                              | WA1 | WA2 | WA3 | WA4 | WA5 | WA6 | WA7 | WA8 | WA9 | WA10 |
|-------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| C.H. Lin and Y.Y. Lay [16]         | Yes | N/A | Yes | Yes | Yes | No  | Yes | Yes | No  | No   |
| C.L. Chen et al. [5]                | No  | No  | Yes | No  | No  | No  | Yes | Yes | No  | No   |
| T.T. Truong et al. [6]              | No  | No  | No  | Yes | No  | No  | Yes | Yes | No  | No   |
| M.K. Khan et al. [7]                | N/A | Yes | Yes | Yes | Yes | Yes | Yes | Yes | No  | No   |
| G.-S. Poh et al. [18]               | No  | No  | No  | No  | No  | No  | Yes | Yes | No  | No   |
| A.K. Das [8]                       | No  | No  | No  | Yes | No  | No  | Yes | Yes | No  | No   |
| Y.H. An [9]                        | No  | No  | Yes | No  | No  | No  | N/A | No  | No  | No   |
| M.K. Khan and S. Kumari [10]       | Yes | Yes | N/A | N/A | Yes | Yes | Yes | N/A | N/A | No   |
| X. Li et al. [12]                  | N/A | Yes | Yes | No  | Yes | Yes | Yes | Yes | No  | No   |
| A. Chaturvedi et al. [13]          | N/A | Yes | N/A | Yes | Yes | Yes | Yes | N/A | N/A | No   |
| Weixin Bian et al. [15]            | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | No  | Yes  |
| Our propose                         | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes  |

The details of notations are WA: Withstands kinds of attacks, WA1: withstands online password guessing attack, WA2: withstands offline password guessing attack, WA3: withstands stolen mobile device/smart card attack, WA4: withstands replay attack, WA5: withstands user masquerade attack, WA6: withstands server spoofing attack, WA7: withstands man-in-the-middle attack, WA8: withstands known session key, WA9: withstands DoS attack, WA10: withstands cloning attack.
the server is 2.85 ms, $T_{me}$ in the device is 7.86 ms and in the server is 2.34 ms, $T_h$ in the device is 0.026 ms and in the server is 0.011 ms.

Based on table 4, Lin and Lay [16] get the computational complexity of registration phase $T_h + T_{me}$, login and mutual authentication for user $2T_h + 2T_{me}$ and server $T_h + 2T_{me}$ with a total computational time of 28.35 ms. Chen et al. [5] get the computational complexity of registration phase $3T_h$, login and mutual authentication for user $5T_h$ and server $3T_h$ with a total computational time of 0.24 ms. Troung et al. [6] get the computational complexity of registration phase $4T_h$, login and mutual authentication for user $6T_h$ and server $5T_h$ with a total computational time of 0.312 ms. Khan et al. [7] get the computational complexity of registration phase $5T_h$, login and mutual authentication for user $7T_h$ and server $6T_h$ with a total computational time of 0.348 ms. Poh et al. [18] get the computational complexity of the registration phase $3T_h$, login and mutual authentication for user $4T_h$ and server $3T_h$ with a total computational time of 0.215 ms.

Das [8] gets the computational complexity of registration phase $3T_h$, login and mutual authentication for user $5T_h$ and server $5T_h$ with a total computational time of 0.263 ms. An [9] gets the computational complexity of registration phase $3T_h$, login and mutual authentication for user $5T_h$ and server $5T_h$ with a total computational time of 0.24 ms. Khan and Kumari [10] get the computational complexity of registration phase $3T_h$, login and mutual authentication for user $5T_h$ and server $5T_h$ with a total computational time of 0.289 ms. Li et al. [12] get the computational complexity of registration phase $T_{FE, gen} + 3T_h$, login and mutual authentication for user $T_{FE, rec} + 5T_h$ and server $2T_{me}$ with a total computational time of 48.48 ms. Li et al. [12] get the computational complexity of registration phase $T_{FE, gen} + 3T_h$, login and mutual authentication for user $T_{FE, rec} + 6T_h + 2T_{me}$ and server $5T_h + 2T_{me}$ with a total computational time of 25.64 ms.

Chaturvedi et al. [13] get the computational complexity of registration phase $T_{FE, gen} + 3T_h + 2T_{me}$, login and mutual authentication for user $T_{FE, rec} + 5T_h + 3T_{me}$ and server $4T_h + 3T_{me}$ with a total computational time of 48.48 ms. Weixin Bian et al. [15] get the computational complexity of registration phase $T_{FE, gen} + 7T_h + T_{PUF}$, login and mutual authentication for user $T_{FE, rec} + T_{FE, gen}$ and server $T_{FE, rec} + 7T_h$ with a total computational time of 10.26 ms. Our proposal has the computational complexity of registration phase $T_{FE, gen} + 2T_h + 2T_{PUF}$, login and mutual authentication for user $T_{FE, rec} + 7T_h + 2T_{PUF}$ and server $3T_h$ with a total computational time of 5.68 ms. Table 5 and the diagram in Fig. 2 show that our protocol has lower computational time than the user authentication protocol proposed by [16], [12], [13], and [15]. Additionally, our protocol not only has low computational time but also fulfills

| Schemes | Registration | Login and Authentication | Total (MS) |
|---------|--------------|--------------------------|------------|
| C.H. Lin and Y.Y. Lay [16] | $T_h$ + $T_{me}$ | $2T_h$ + $2T_{me}$ | 19.156 + $5T_h$ + $3T_h$ | 4.735 + 25.64 |
| L. Chen et al. [5] | $3T_h$ | $5T_h$ | $2T_h$ | 0.078 + 0.13 + 0.033 | 0.24 |
| T.T. Truong et al. [6] | $4T_h$ | $6T_h$ | 0.104 + 0.156 + 0.055 | 0.312 |
| M.K. Khan et al. [7] | $5T_h$ | $7T_h$ | 0.13 + 0.182 + 0.066 | 0.348 |
| G.-S. Poh et al. [18] | $3T_h$ | $4T_h$ | 0.078 + 0.104 + 0.033 | 0.215 |
| A.K. Das [8] | $3T_h$ | $5T_h$ | 0.078 + 0.13 + 0.055 | 0.263 |
| Y.H. An [9] | $3T_h$ | $5T_h$ | 0.078 + 0.13 | 0.252 |
| M.K. Khan and S. Kumar [10] | $4T_h$ | $5T_h$ | 0.104 + 0.13 + 0.055 | 0.289 |
| X. Li et al. [12] | $T_{FE, gen}$ | $T_{FE, rec}$ | $5T_h$ | 1.748 + 19.156 + 4.735 | 26.54 |
| A. Chaturvedi et al. [13] | $T_{FE, gen}$ | $T_{FE, rec}$ | $4T_h$ | 17.468 + 26.99 + 4.02 | 48.48 |
| Weixin Bian et al. [15] | $T_{FE, gen}$ | $T_{FE, rec}$ | $T_{FE, rec}$ | 1.972 + 5.356 + 2.927 | 10.26 |
| Our propose | $T_{FE, gen}$ | $T_{FE, rec}$ | $T_{FE, rec}$ | 1.962 + 3.681 + 5.68 | 5.68 |

The details of equation are $T_{FE, gen}$: Time of Fuzzy Extractor Generation, $T_{FE, rec}$: Time of Fuzzy Extractor Reconstructions, $T_h$: Time of one-way hash function, $T_{PUF}$: Time of PUF process, $T_{me}$: Time of modular exponential.
achievement of all security features SF1-SF7, and withstand kinds of attacks (WA1-WA9) in table 2, and 3.

4 Analysis Formally Using Ban Logic

This subsection presents an analysis formally using BAN logic as evidence that our scheme achieves secure mutual authentication.

4.1 Brief Explanation of BAN Logic

The BAN Logic has three objects [38], namely, participants, encryption keys, and logical formulas. In this paper, participants are device and server, encryption keys are $K = h(K_e||e_u), N_s, N_u, R_u$, logical formulas are based on Table 5 and 6.

| Symbol | Explanation |
|--------|-------------|
| $P \equiv X$ | $P$ believes $X$ |
| $P > X$ | $P$ sees $X$ |
| $P \sim X$ | $P$ said $X$ |
| $P \Rightarrow X$ | $P$ has jurisdiction over $X$ |
| $(\# X)$ | $X$ is fresh |
| $P \leftrightarrow Q$ | $P$ and $Q$ may use the shared $K$ to communicate |
| $P \ni X$ | $P$ processes or is capable of processing, formula $X$ |
| $(X)_K$ | Formula $X$ is encrypted under the key $K$ |

| Rule | Postulate |
|------|-----------|
| R1 is the rule of Message-meaning | $\frac{\{P \equiv P \ni Q \ni X\} \ P\equiv Q \ni X \ P\equiv Q \ni X}{\ P\equiv Q \ni X}$ |
| R2 is the rule of Nonce-verification | $\frac{\{P \equiv Q \ni X\} \ P\equiv Q \ni X}{\ P\equiv Q \ni X}$ |
| R3 is the rule Jurisdiction | $\frac{\{P \equiv Q \ni X\} \ P\equiv Q \ni X}{\ P\equiv Q \ni X \ P\equiv Q \ni X}$ |
| R4 is the rule Seeing | $\frac{\{P \ni X\} \ P\ni X \ P\ni X}{\ P\ni X \ P\ni X}$ |
| R6 is the rule of Freshness | $\frac{\{P \equiv Q\} \ P\equiv Q\ X}{\ P\equiv Q\ X}$ |
| R7 is the rule of Belief | $\frac{\{P \equiv Q\} \ P\equiv Q\ X}{\ P\equiv X}$ |

4.2 Proof of the scheme’s formal analysis using BAN Logic

This subsection presents the formal analysis of Mutual Authentication between user’s device (U) and the server (S), starting from the idealized protocol, the assumption and goal, and protocol analysis.

1) Create the idealize protocol

We create the ideal protocol and omit the first process,

a. Message

$S \rightarrow U: \{N'_s, C'_u, PID'_u, V_1\}$

$U \rightarrow S: \{N_s, V_2, N'_i\}$

b. Logical Formula

$S \rightarrow U: \{N'_s, C'_u, PID'_u, h(K_e||e_u), N_s, N_u, R_u\}$

$U \rightarrow S: \{N_s, R_u^{new}, U \leftrightarrow S\}$

2) The assumption or goals

This subsection presents the assumptions and goals including the belief of shared keys, the belief of freshness, and trust.

a. Belief of shared keys

$U| \equiv U_{K=h(K_e||e_u)} S$

The user believes that user and server using the same key $K = h(K_e||e_u)$

$S| \equiv S_{K=h(K_e||e_u)}=K U$

The server believes that server and using the same key $h(K_e||e_u) = K$

$U| \equiv U_{R_u} S$

The user believes that user and server using a shared key $R_u$

$S| \equiv S_{\leftrightarrow U} R_u$

The server believes that server and user using a shared key $R_u$

b. Belief of freshness

$U| \equiv \#(N_u)$

The user believes that $N_u$ is fresh.

$S| \equiv \#(N_u)$

The server believes that $N_u$ is fresh.

$S| \equiv \#(N_u)$

The server believes that $N_u$ is fresh.

$U| \equiv \#(N_u)$

The user believes that $N_u$ is fresh.

The user believes that the server believes between server and user using a shared secret $R_u$.

$U| \equiv S| \rightarrow U_{\leftrightarrow S} R_u$

The user believes that the server has jurisdiction over session key $SK$ between user and server.

$S| \equiv U| \rightarrow U_{\leftrightarrow S} S$

The server believes that the user has jurisdiction over session key $SK$ between server and user.
3) The protocol analysis

This subsection presents the protocol analysis in mutual authentication between the user’s device (U) and the server (S), starting from step 2 then followed by step 3.

\[ \text{Step2: } S \rightarrow U: \{N'_u, C'_m, PID'_{us}\}_{K=h(K)|e_u}, N_s, N_u, R_u \]

a. message meaning (R1)

\[ U \vdash S \rightarrow \{N'_u, C'_m, PID'_{us}\}_{K=h(K)|e_u}, N_s, N_u, R_u \]

\[ U \vdash \#(N_u, N_s) \]

Because the user believes that the server and the user use a secret key \( K = h(K)|e_u \), the user sees that \( \{N'_u, C'_m, PID'_{us}\}_{K=h(K)|e_u}, N_s, N_u, R_u \) is encrypted by \( K = h(K)|e_u \), \( N_s, N_u, R_u \) the user believes that the server once said \( \{N'_u, C'_m, PID'_{us}\}_{K=h(K)|e_u} \) and the user believes that the \( \{N_u, N_s\} \) are fresh, the user believes that the \( \{N_u, N_s\} \) are fresh.

\[ \text{Step3: } U \rightarrow S: \{N_u, R_u^{new}, U \rightarrow S\} \]

a. message meaning (R1)

\[ S \vdash \{N_u, R_u^{new}, U \rightarrow S\} \]

\[ S \vdash \#(N_u) \]

Because the server believes that the server and the user use secret key \( R_u, N_u \), the server sees that \( \{N_u, R_u^{new}, U \rightarrow S\} \) are encrypted by \( R_u, N_u \), the server believes that the user once said \( \{N_u, R_u^{new}, U \rightarrow S\} \)

b. Nonce verification (R2)

\[ S \vdash \#(N_u) \]

Because the server believes that he user once said \( \{N_u, R_u^{new}, U \rightarrow S\} \) and the server believes that \( N_s \) is fresh, the server believes that the \( \{N_u, N_s, R_u^{new}, U \rightarrow S\} \) are fresh.

c. Belief (R7) and Jurisdiction (R3)

\[ S \vdash \#(N_u) \rightarrow \{N_s, (N_u, R_u^{new}, U \rightarrow S)^{SK}\}, S \vdash \#(N_s) \]

Because the server believes that the user once said \( \{N_s, (N_u, R_u^{new}, U \rightarrow S)^{SK}\} \) and the server believes that \( N_s \) is fresh, the server believes that the user believes \( \{N_s, (N_u, R_u^{new}, U \rightarrow S)^{SK}\} \).

\[ S \vdash \#(N_u) \rightarrow \{N_u, (N_r, R_u^{new}, U \rightarrow S)^{SK}\}, S \vdash \#(N_u) \]

Because the server believes that the user believes \( \{N_u, (N_r, R_u^{new}, U \rightarrow S)^{SK}\} \), the server believes that the user believes \( \{N_u, (N_r, R_u^{new}, U \rightarrow S)^{SK}\} \).

\[ S \vdash \#(N_u) \rightarrow \{N_u, (N_r, R_u^{new}, U \rightarrow S)^{SK}\}, S \vdash \#(N_u) \]

Because the server believes that the user believes \( \{N_u, (N_r, R_u^{new}, U \rightarrow S)^{SK}\} \), the server believes that the user believes \( \{N_u, (N_r, R_u^{new}, U \rightarrow S)^{SK}\} \).

\[ S \vdash \#(N_u) \rightarrow \{N_u, (N_r, R_u^{new}, U \rightarrow S)^{SK}\}, S \vdash \#(N_u) \]

Because the server believes that the user believes \( \{N_u, (N_r, R_u^{new}, U \rightarrow S)^{SK}\} \), the server believes that the user believes \( \{N_u, (N_r, R_u^{new}, U \rightarrow S)^{SK}\} \).

\[ S \vdash \#(N_u) \rightarrow \{N_u, (N_r, R_u^{new}, U \rightarrow S)^{SK}\}, S \vdash \#(N_u) \]

Because the server believes that the user believes \( \{N_u, (N_r, R_u^{new}, U \rightarrow S)^{SK}\} \), the server believes that the user believes \( \{N_u, (N_r, R_u^{new}, U \rightarrow S)^{SK}\} \).

5 ANALYSISI FROMALLY USING ROR MODEL

In this section, we also proof security analysis formally using the Real or Random (ROR) model, and in our ROR model has two participants are the user’s device (D_i), and Server (S), the model as follows.

1) Participants: \( \pi_{D_i}^d \) and \( \pi_{D_i}^s \) are the oracle of \( d, s \) related with user’s device \( D_i, server S \).

2) Partnering: the partnering achieved if and only if “fulfills two conditions” 1. Communication between \( \pi_{D_i}^d \) and \( \pi_{D_i}^s \) based on communication session-id sid. 2. All
messages exchanged between $\pi_{D_i}^d$ and $\pi_{S}^d$ are unique.

3) **Freshness:** The freshness achieved if the adversary cannot divulge the session key SK between $\pi_{D_i}^d$ and $\pi_{S}^d$

4) **Adversary:** In the ROR model, the adversary $A$ has capabilities with the Dolev-Yao (DY) adversary model [39], where the adversary can fully control the transmitted message in public channels. The adversary $A$ can perform eavesdropping, altering, deleting, and also have access to the queries as follow:

a. $\text{Execute}(\pi^d, \pi^e)$: the equation denotes the Adversary can intercept all communication in the public channel between the user’s device and the server. The model is called an eavesdropping attack.

b. $\text{Send}(\pi^d, m)$: Where the adversary has the capability to send a message to any participant e.g. $\pi^d$, and the adversary can receive and reply as well. This query is called the active attack.

c. $\text{CorruptDevice}(\pi^e)$: The adversary captures the device and he/she can extract the credential stored from the memory of the device. It called a stolen or lost device attack.

d. $\text{Test}(\pi^d, \pi^e)$: The semantic security is determined by the establishment of the session key SK between the user’s device $\pi^d$ and the server $\pi^e$ by following the ROR model’s indistinguishability in the ROR model [40]. Initially, the adversary tores an unbiased coin $c$ and then the result is an outcome of the toss. If $c = 1$ denotes the new share key is fresh, hence, the adversary $A$ doesn’t need to Reveal($\pi^d, \pi^e$), namely, session key SK. Otherwise, If SK is not fresh, $\pi^d, \pi^e$ returns a null, it denotes the outcome is a random key.

5) **Semantic Security Of Session Key**

In the ROR model, the adversary must have the ability to distinguish the actual session key and random key. The adversary repeated running $\text{Test}()$ queries $\pi^d$ or $\pi^e$ and stores the test’s result in a bit $c$. If $c = c’$ denotes that the adversary is a winner. Where $c’$ represents the bit guessed by the adversary randomly. The ability of the adversary to break semantic security from the Authentication Key Agreement protocol P in $t$ specific time is defined $\text{Adv}_{EAK}^t(t) = |2\cdot P[\text{SUCCESS}] - 1|$, where $\text{SUCCESS}$ represents an event that adversary A wins the game.

6) **Random Oracle**

In our ROR model, the collision $h(\cdot)$ and the secure PUF($\cdot$) can be accessed by the adversary and every participant.

7) **Security Proof**

The definition of the secure PUF and the Collision resistant-one-way-hash function is showed, and it will be used to prove our scheme’s semantic security. The description of them as follows:

**Definition 1 (collision resistant one-way hash function):**

Formally definition of Collision resistant-one-way hash function, $h: \{0,1\}^* \rightarrow \{0,1\}^n$ denotes the mathematical model of hash function output, where arbitrary length input of bit into hash function yields a fixed-length output’s bit. The definition of the adversary’s advantage for finding the Collision in time t is denoted by $\text{Adv}_{HA}^{t}(t)$, where $\text{Adv}_{HA}^{t}(t) = \Pr[(x_1, x_2) \in A: x_1 \neq x_2, h(x_1) = h(x_2)]$. $\Pr(X)$ represents a probabilistic of $X$ randomly; $x_1, x_2$ are strings selected by A randomly. The maximum execution time of the adversary ($\phi, t$) success for attacking collision hash $h(\cdot)$ is denoted by $\text{Adv}_{HA}^{t}(t) \leq \phi$.

**Definition 2 (PUF function is secure):**

Secure PUF can be achieved, e.g., $PUF_1$ produces output responses $R_1, R_2 \in \{0,1\}^k$ with at least $d_1$ variation for two input challenges $C_1, C_2 \in \{0,1\}^k$, and if an input challenge $C_1$ produces distinct output responses $R_1, R_2 \in \{0,1\}^k$ with at least $d_2$ variation for any two different PUFs ($PUF_1, PUF_2$). Therefore, $\Pr[HD(PUF_1(C_1), PUF_2(C_2))] > d_1 \big|_{1} = 1 - \epsilon$, $\Pr[HD(PUF_1(C_1), PUF_2(C_2))] > d_2 \big|_{1} = 1 - \epsilon, \epsilon$ is a small value negligible, the adversary selects the challenges randomly denoted by $C_1, C_2$, HD is Hamming Distance and $d_1, d_2$ are PUF’s error tolerance thresholds.

Rely on Definition 1 and Definition 2 aforesaid and acknowledging Zipf’s law [41]. Theorem 1 yields semantic security of proposed protocol as follow:

**Theorem 1:** Let the Adversary run an attack in the proposed protocol P at the polynomial-time under the ROR model. The user-chosen passwords based on Zipf’s law [41], $l_1, l_2$ are the bits in biometric’s secret key $K_u, ID_u$ is secret user identity. The estimation of Adversary’s advantage to break the semantic security of the protocol and reveal Session Key between user’s device and server as follows.

$\text{Adv}_{EAK}^t(t) \leq \frac{q_h^2}{|\text{Hash}|} + \frac{q_s^2}{|\text{PUF}|} + 2 \cdot \max(C’, q_s^2, 2^{\frac{1}{2}l_1}, \frac{1}{2^{l_2}})$

Where $q_h, q_s, q_s$ are the number of Hash, PUF, and send queries, respectively. $|\text{Hash}|$ is the range space’s $h(\cdot)$, $|\text{PUF}|$ is the range space’s PUF($\cdot$), and $C’$ and $s’$ are Zipf’s parameters [41].

**Proof:** in this article, we adopt the similar proof present by [42], [43], and [44]. $G_j$ denotes five sequence games, where $j \in [0,4]$. The Adversary A successes guess bit $c$ in the game $G_j$ is represented by $\text{SUCCESS}$. The detail of the game as follows.

**Game $G_0$:** The game is considered an actual attack on protocol P by Adversary A under the ROR model. The beginning of game $G_0$ as follows:

$\text{Adv}_{EAK}^t(t) = |2 \cdot \Pr[\text{SUCCESS}_0] - 1 |$ \hspace{1cm} (1)

**Game $G_1$:** The game 1 is an eavesdropping attack, Adversary A activate $\text{Execute}(\pi^d, \pi^e)$ hence A can intercept all the transmitted message such as $M_1: \{PID_{u0}, N_{u0}, V_0\}$, $M_2: \{N_s, C_u, PID_{u0}, V_1\}$, and $M_3: \{V_1, N_3\}$. After that adversary A executes the Test to verify whether the result is the actual secret response or random number. In our protocol SK is computed from $SK = h(R_u|||N_3||V_1)$. In this case, the computation of session key SK needs to reveal the secret credentials $R_u, ID_u, K_u$ and in fact that the adversary unknow these credentials, where only a legitimate device can compute $R_u$, and only legitimate device can compute SK. Therefore, Adversary A’s probability to win in $G_1$ by eavesdropping attack is not increased. Consequently, we have following the result:

$\Pr[\text{SUCCESS}_1] = \Pr[\text{SUCCESS}_0]$ \hspace{1cm} (2)

**Game $G_2$:** game 2 simulates the Send and Hash queries
for modeling an active attack where Adversary A attempts to defraud legitimate participants to accept the messages are modified by adversary A. Because all messages transmitted $M_{1a}, M_{2a}, M_{3a}$ are constructed using the hash and no collision occurs when adversary A activates Send query by helping $h(\cdot)$ query (see Definition 1). based on the birthday paradox, the result as follows:

$$\Pr[\text{SUCCESS}] - \Pr[\text{SUCCESS}] \leq \frac{q_h^2}{2^{(|\text{hash}|)}}$$

(3)

**Game $G_3$:** This game has a similar argument with $G_2$, but the difference between $G_3$ and $G_2$ is that $G_3$ simulates the Send and PUQ queries. Based on secure PUF in Definition 2, the relationship as follows:

$$\Pr[\text{SUCCESS}] - \Pr[\text{SUCCESS}] \leq \frac{q_h^2}{2^{(|\text{PUF}|)}}$$

(4)

**Game $G_4$:** this is the final game, the adversary activates CorruptDevice, where adversary A can extract $V, h, PIDsyn, PIDus, K_{syn}$ from the device. However, the adversary cannot obtain all credentials, including identity and a fixed key $K_w$, generated from biometric input. The probability of guessing the fixed key $K_w$ of $l_1$ bits and identity of $ID_v$ of $l_2$ bits are $\frac{1}{2^l_1}$ and $\frac{1}{2^l_2}$ respectively [41]. Additionally, the adversary can guess the password based on Zipf's law [41]. If only guess the password, the adversary has an advantage over 0.5 when $q_s = 10^7$ or $10^8$ [41]. Hereafter, if the attacker was guessing the password based on the user's identity, the adversary has an advantage over 0.5 when $q_s \leq 10^6$ [41]. In actual conditions, the system does not allow many to try incorrect passwords. Therefore, the game $G_3$ and $G_4$ have to indicates nothingsness of guessing attack. Thus, the result as follows:

$$\Pr[\text{SUCCESS}] - \Pr[\text{SUCCESS}] \leq \max\{C', q_s, q_s, q_s\}$$

(5)

In the final session, after activating the Test query, the adversary A must guess bit $c'$ to win the game $G_4$. Therefore, the result as below:

$$\Pr[\text{SUCCESS}] = \frac{1}{2}$$

(6)

Based on equation (1), (2), and (6), we have

$$\frac{1}{2} \text{Adv}_{P,A}(e) = \frac{1}{2} \Pr[\text{SUCCESS}] - \frac{1}{2} = \Pr[\text{SUCCESS}] - \Pr[\text{SUCCESS}]$$

(7)

We get the following result by applying the triangle inequality and using equations (3), (4), and (5).

$$\Pr[\text{SUCCESS}] - \Pr[\text{SUCCESS}] \leq \Pr[\text{SUCCESS}] - \Pr[\text{SUCCESS}]$$

$$\leq \Pr[\text{SUCCESS}] - \Pr[\text{SUCCESS}]$$

$$\leq \Pr[\text{SUCCESS}] - \Pr[\text{SUCCESS}]$$

$$\leq \Pr[\text{SUCCESS}] - \Pr[\text{SUCCESS}]$$

$$\leq \frac{1}{2} \max\{C', q_s, q_s, q_s\}$$

(8)

Finally, we obtain the necessary result by solving equations (7) and (8):

$$\text{Adv}_{P,A}(e) \leq \frac{q_h^2}{|\text{Hash}|} + \frac{q_h^2}{|\text{PUF}|} + 2 \max\{C', q_s, q_s, q_s\}$$

6 **ANALYSIS FORMALLY USING SCTHYER TOOL**

This article also uses the Scythe tool as a formal analysis to validate our proposed protocol. The Scythe tool is developed based on Python programming to verify the protocol security [45]. This tool also follows Delov-Yao (DY) adversary model [39]. There are two steps for evaluating the protocol using this tool. Step 1, the modeling of the protocol based on Security Description Language (SPDL), and then Step 2 run the tool to show security

---

Fig. 5. The modeling of the proposed protocol in SPDL
claims of the protocol.

Figure 5 is the SPDL modeling of our protocol, and Figure 6 is a validating result of the Scyther tool, showing that the Scyther tool cannot find the attack in our scheme. Hence, it can be said that our protocol withstands various attacks based on the Scyther tool claim.

7 Conclusion

This article proposed a new user authentication protocol based on biometric and PUF to improve the security features and resolve the user authentication problem. Based on the informal analysis, our proposed scheme fulfills security features such as Mutual Authentication, Anonymity, Untraceability, Perfect forward secrecy, secrecy session key security, resolve Desynchronization, availability (SF1-SF7), and withstand various kinds of attacks (WA1-WA10). Additionally, formal analysis using BAN Logic ensures our scheme to achieve mutual authentication, the result of the ROR model and Scyther tool show that our proposed scheme withstands various kinds of attacks and it proofs and strengthens our informal analysis. On the other hand, based on the computational complexity comparison, our protocol obtains lower computational cost compared to the scheme proposed by [16], [12], [13], and [15]. Therefore, our scheme is much suitable to be applied in user authentication.

Acknowledgment

The author would like to thank the National Science Council of Taiwan for supporting this work under Contract No. MOST 109-2221-E-006-168-; No. MOST 108-2221-E-006-107.

References

[1] L. Lamport, “Password Authentication with Insecure Communication,” Commun. ACM, vol. 24, no. 11, 1981.
[2] M.-S. Hwang and L.-H. Li, “A New Remote User Authentication Scheme Using Smart Cards,” IEEE Trans. Consum. Electron., vol. 46, pp. 28–30, 2000.
[3] C. Chan and L.M. Cheng, “CRYPTANALYSIS OF A REMOTE USER AUTHENTICATION SCHEME USING SMART CARDS,” IEEE Trans. Consum. Electron., vol. 46, no. 4, pp. 992–993, 2000.
[4] M. K. Khan, J. Zhang, and X. Wang, “Chaotic hash-based fingerprint biometric remote user authentication scheme on mobile devices,” Chaos, Solitons and Fractals, vol. 35, pp. 519–524, 2008.
[5] C. Chen, C. Lee, and C. Hsu, “Mobile device integration of a fingerprint biometric remote authentication scheme,” Int. J. Commun. Syst., no. April 2011, 2012.
[6] T. Truong, M. Tran, and A. Duong, “Robust mobile device integration of a fingerprint biometric remote authentication scheme,” IEEE Int. Conf. Adv. Inf. Netw. Appl. Robust, 2012.
[7] M. Khurram, K. Saru, and M. K. Gupta, “More efficient key-hash based fingerprint remote authentication scheme using mobile device,” Computing, pp. 793–816, 2014.
[8] A. K. Das, “Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards,” IET Inf. Secur., no. June 2010, pp. 145–151, 2011.
[9] Y. An, “Security Analysis and Enhancements of an Effective Biometric-Based Remote User Authentication Scheme Using,” J. Biomed. Biotechnol., vol. 2012, 2012.
[10] M. K. Khan and S. Kumari, “An Improved Biometrics-Based Remote User Authentication Scheme with User Anonymity,” Biomed Res. Int., vol. 2013, p. 9, 2013.
[11] S. Ibaoun, A. Abou, E. Kalam, and V. Poirriez, “Analysis and enhancements of an efficient biometric-based remote user authentication scheme using smart cards,” ACS/IEEE Int. Conf. Comput. Syst. Appl., 2016.
[12] X. Li, J. Niu, Z. Wang, and C. Chen, “Applying biometrics to design three-factor remote user authentication scheme with key agreement,” Secur. Commun, NETWORKS, no. April 2013, pp. 1488–1497, 2014.
[13] A. Chaturvedi, D. Mishra, S. Jangirala, and S. Mukhopadhyay, “A privacy preserving biometric-based three-factor remote user authenticated key agreement,” J. Inf. Secur. Appl., no. April 2018, 2016.
[14] J. Lee, S. Ryu, and K. Yoo, “Fingerprint-based remote user authentication scheme using smart cards,” ElectronicsLetters, vol. 38, no. 12, pp. 554–555, 2002.
[15] W. Bhan, P. Gope, Y. Cheng, and Q. Li, “Bio-AKA : A flexible biometrics remote user authentication scheme,” Security and Communication Networks, vol. 2014, pp. 26, 2014.
[16] M. K. Khan and J. Zhang, “Improving the security of a flexible biometric remote user authentication scheme,” Comput. Intell. Inter. Comm., vol. 29, pp. 23–35, 2007.
[17] G. Sen Poh, P. Gope, and J. Ning, “PrivHome : Privacy-Preserving Authentication Communication in Smart Home Environment,” IEEE Trans. DEPENDABLE Secur. Comput., vol. 99, no. 99, 2019.
[18] S. Barra, M. Nappi, and K. Raymond, “Cloud-Based Biometrics (Biometrics as a Service ) for Smart Cities, Nations, and Beyond,” IEEE Cloud Comput., no. October, pp. 90–100, 2018.
[19] K. Zhou, S. Member, J. Ren, and S. Member, “PassBio: Privacy-Preserving User-Centric Biometric Authentication,” IEEE Trans. Inf. FORENSICS Secur., vol. 13, no. 12, pp. 3050–3063, 2018.
[20] Y. Wang, J. Wan, J. Guo, Y. Cheung, and P. C. Yuen, “Inference-Based Similarity Search in Randomized Montgomery Domains for Privacy-Preserving Biometric Identification,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 7, 2018.
[21] A. K. Jain, K. Nandakumar, and A. Ross, “50 years of biometric research: Accomplishments, challenges, and opportunities,” Pattern Recognit. Lett., vol. 79, pp. 80–105, 2016.
[22] P. Tuyll and L. Batina, “RFID-Tags for Anti-counterfeiting,” Top. Cryptol. CT-RSA (LNCS 3860), Heidelberg, Ger. Springer, pp. 115–131, 2006.
Y. Dodis, L. Reyzin, and A. Smith, “Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data,” Adv. Cryptology — EUROCRYPT’2004 (Lecture Notes Comput. Sci. Heidelberg, Ger. Springer, pp. 523–540, 2004.

G. E. Suh and S. Devadas, “Physical Unclonable Functions for Device Authentication and Secret Key Generation,” 2007 44th ACM/IEEE Des. Autom. Conf. San Diego, pp. 9–14, 2007.

C. Bohm and M. Hofer, Physical Unclonable Functions in Theory and Practice. New York: NY, USA: Springer, 2012.

D. Jeon, J. H. Baek, D. K. Kim, and B. Choi, “Toward Zero Bit-Error Rate Physical Unclonable Function: Mismatch-Based vs. Physical-Based Approaches in Standard CMOS Technology,” 2015 Euromicro Conf. Digit. Syst. Des., pp. 407–414, 2015.

K. Chuang et al., “A Physically Unclonable Function Using Soft Oxide Breakdown Featuring 0 % Native,” IEEE J. Solid-State Circuits, vol. 54, no. 10, pp. 2765–2776, 2019.

P. P. Crystals, X. Lu, S. Member, L. Hong, and S. Member, “CMOS Optical PUFs Using Noise-Immune Incorporating Passive Variations,” IEEE J. Solid-State Circuits, vol. 53, no. 9, pp. 2709–2721, 2018.

W. Wang, Y. Yona, S. N. Diggavi, and P. Gupta, “Design and Analysis of Stability-Guaranteed PUFs,” IEEE Trans. Inf. FORENSICS Secur., vol. 13, no. 4, pp. 978–992, 2018.

C. Herder, M. Yu, F. Koushanfar, and S. Devadas, “Physical Unclonable Functions and Applications: A Tutorial,” Proc. IEEE, vol. 102, no. 8, pp. 1126–1141, 2014.

L. Ben, “The Pairing-Based Cryptography Library,” [Online]. Available: https://crypto.stanford.edu/pbc/ [Accessed: 11-May-2020].

Oracle, “Java Cryptography Architecture (JCA),” [Online]. Available: https://docs.oracle.com/javase/8/docs/technotes/guides/security/crypto/CryptoSpec.html [Accessed: 11-May-2020].

A. Mohammadali, M. S. Haghighi, and M. H. Tadayon, “A Novel Identity-Based Key Establishment Method for Advanced Metering Infrastructure in Smart Grid,” IEEE Trans. Smart Grid, vol. 9, no. 4, pp. 2834–2842, 2018.

D. Wu, S. Member, C. Zhou, and S. Member, “Fault-Tolerant and Scalable Key Management for Smart Grid,” IEEE Trans. Smart Grid, vol. 2, no. 2, pp. 375–381, 2011.

J. Tsai and N. Lo, “Secure Anonymous Key Distribution Scheme for Smart Grid,” IEEE Trans. Smart Grid, vol. 7, no. 2, pp. 906–914, 2016.

V. Odelu, A. K. Das, M. Wazid, M. Conti, and S. Member, “Provably Secure Authenticated Key Agreement Scheme for Smart Grid,” IEEE Trans. Smart Grid, vol. 9, no. 3, pp. 1900–1910, 2018.

M. Burrows and R. Needham, “A Logic of Authentication,” ACM Trans. Comput. Syst., vol. 8, no. 1, 1989.

D. Wang, S. Member, H. Cheng, P. Wang, and S. Member, “Zipf’s Law in Passwords,” IEEE Trans. Inf. FORENSICS Secur., vol. 12, no. 11, pp. 2776–2791, 2017.

P. Gope, A. K. Das, N. Kumar, and Y. Cheng, “Lightweight and Physically Secure Anonymous Mutual Authentication Protocol for Real-Time Data Access in Industrial Wireless,” IEEE Trans. Ind. INFORMATICS, vol. 15, no. 9, pp. 4957–4968, 2019.

S. Roy et al., “Provably Secure Fine-Grained Data Access Control Over Multiple Cloud Servers in Mobile Cloud Computing Based Healthcare Applications,” IEEE Trans. Ind. INFORMATICS, vol. 15, no. 1, pp. 457–468, 2020.

S. Roy, S. Chatterjee, A. K. Das, and S. Chattopadhyay, “On the Design of Provably Secure Lightweight Remote User Authentication Scheme for Mobile Cloud Computing Services,” IEEE Access, no. October, 2017.

C. Cremers, “The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols,” Proc. Int. Conf. Comput. Aided Verification, pp. 1–4.