Practical and Provably Secure Three-Factor Authentication Protocol Based on Extended Chaotic-Maps for Mobile Lightweight Devices

Shuming Qiu, Ding Wang, Guoai Xu, and Saru Kumari

Abstract—Due to the limitations of symmetric-key techniques, authentication and key agreement (AKA) protocols based on public-key techniques have attracted much attention, providing secure access and communication mechanism for various application environments. Among these public-key techniques used for AKA protocols, chaotic-map is more effective than scalar multiplication and modular exponentiation, and it offers a list of desirable cryptographic properties such as un-predictability, un-repeatability, un-certainty, and higher efficiency than scalar multiplication and modular exponentiation. Furthermore, it is usually believed that three-factor AKA protocols can achieve a higher security level than single- and two-factor protocols. However, none of existing three-factor AKA protocols can meet all security requirements. One of the most prevalent problems is how to balance security and usability, and particularly how to achieve truly three-factor security while providing password change friendliness. To deal with this problem, in this article we put forward a provably secure three-factor AKA protocol based on extended chaotic-maps for mobile lightweight devices, by adopting the techniques of "Fuzzy-Verifiers" and "Honeywords". We prove the security of the proposed protocol in the random oracle model, assuming the intractability of extended chaotic-maps Computational Diffie-Hellman problem. We also simulate the protocol by using the AVISPA tool. The security analysis and simulation results show that our protocol can meet all 13 evaluation criteria regarding security. We also assess the performance of our protocol by comparing with seven other related protocols. The evaluation results demonstrate that our protocol offers better balance between security and usability over state-of-the-art ones.

Index Terms—Extended chaotic-maps, three-factor, authentication and key agreement, guessing attack, perfect forward secrecy

1 Introduction

With the booming of various sensitive applications over the Internet, user privacy is becoming more and more important, and has attracted widespread concern. For example, in a telecare medicine scenario, physicians are inclined to be assisted with health care systems to check the patient's medical records and diagnose the diseases. In this system, the communication among the physicians, patients and the servers is carried out through an insecure open channel, it is convenient for the intruder to carry out malicious attacks to forge the patients' records, which is fatal for the patient. Furthermore, with the rapid development of mobile application technologies, affordable and portable mobile lightweight devices are becoming very popular. Mobile lightweight devices (e.g.,: laptops, personal digital assistants, smartphone, smartwatch, and wearable devices) are able to access cloud servers for online payment, online voice and video chatting, mobile banking interaction, e-commerce, and so on anytime and anywhere. It has been estimated that by 2020, the number of mobile users worldwide will increase to 7.33 billion, 7.8 percent higher than the 6.8 billion users in 2019 [1]. This means that on average everyone in the world has one mobile device. However, due to the openness of wireless network communication, authentication and anonymity issues in mobile environments must be concerned. Otherwise, the security and privacy of the user data will be difficult to guarantee.

The network communication between mobile users and cloud servers may suffer from various attacks, such as impersonation attack and password guessing attack. Moreover, mobile devices are usually resource constrained and vulnerable to special network attacks. Therefore, it is indispensable to establish an authentication and key agreement (AKA) protocol to protect the conversations between the users with lightweight mobile devices and remote servers in various application environments. However, many protocols sacrifice security in order to achieve usability. For example, the schemes in [2], [3] and [4] can't resist key-compromised user impersonation attack, off-line password guessing attack and can't provide three-factor security. The reason for these defects lies in that, on the one hand,

- *Shuming Qiu is with the School of Mathematics and Statistics, Jiangxi Normal University, Nanchang 330022, China, and also with the National Engineering Laboratory of Mobile Network Security, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: qiushuming2008@163.com.*
- *Ding Wang is with the College of Cyber Science, Nankai University, Tianjin 300350, China, and also with the Tianjin Key Laboratory of Network and Data Security Technology, Nankai University, Tianjin 300350, China. E-mail: wangding@nankai.edu.cn; wangding@pku.edu.cn.*
- *Guoai Xu is with the National Engineering Laboratory of Mobile Network Security and School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: xga@bupt.edu.cn.*
- *Saru Kumari is with the Department of Mathematics, Chaudhary Charan Singh University, Meerut 250004, Uttar Pradesh, India. E-mail: saryusiirohi@gmail.com.*

Manuscript received 22 July 2019; revised 19 Aug. 2020; accepted 2 Sept. 2020. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Ding Wang.)
Digital Object Identifier no. 10.1109/TDSC.2020.3022797

their protocols do not follow the design principle of authentication protocol proposed by Ma *et al.* [5] to achieve robust security, on the other hand, the designers do not employ the state-of-the-art technical means to improve usability. Accordingly, designing an AKA protocol that can balance security and usability remains a challenging problem.

Since the first password authentication and key agreement protocol [6] was presented in 1981, hundreds of AKA protocols have been proposed for Client-Server [7], [8], [9], Multi-Server environment [10], Radio Frequency Identification (RFID) systems [11], etc. For example, Durlanik and Sogukpinar presented an AKA protocol for session initiation protocol (SIP) using elliptic curve cryptography (ECC) [8]. In 2014, Wang-Wang [12] provided the attack model, the design principle and solutions for two-factor AKA protocols for wireless sensor networks. In 2018, Wang *et al.* [9] analyzed the challenges in designing identity-based privacy-preserving AKA protocols for mobile devices.

In order to improve the computational efficiency of the protocol in the mobile lightweight equipment and reduce the communication cost of the protocol, initially scholars tried to design a practical AKA protocol using symmetric cryptography and hash functions (e.g., [11], [13], [14]). However, it is found that these protocols can not provide forward secrecy, and the public-key cryptography technology is indispensable to achieve forward secrecy according to Ma et al [5]. Since then, the public key cryptography technology is becoming to be widely used in AKA protocols. At present, many public key cryptography techniques can be utilized to design AKA protocols, for instance, ECC, RSA, bilinear pairings and so on. Notably, another technique, i.e., chaotic-maps [15] (e.g., Logistic maps and Chebyshev-maps), has been intensively studied by scholars. They refer to the random irregular movement in the deterministic system, whose behaviour is uncertain, unrepeatable and unpredictable. Especially, chaotic maps have been widely used in cryptography, such as using chaotic maps to design encryption systems and key agreement protocols.

In 2005, Xiao *et al.* for the first time presented a deniable AKA protocol for E-Commerce by using chaotic-maps [16]. Later, Alvarez showed that Xiao *et al.*'s protocol suffers the security and efficiency limitations, for example, they pointed out that this scheme cannot resist counterfeiting attacks [17]. Later, Xiao *et al.* [18] improved Xiao *et al.*'s protocol [16] and presented an unconventional AKA protocol based on chaotic-maps. Since then, a large number of chaotic-maps based AKA protocols [19], [20], [21], [22], [23], [24] have been proposed for diverse application scenarios.

The above AKA protocols are mainly based on a single authentication factor (i.e., password) or two authentication factors (i.e., password + smart card). Being faced with demanding security requirements and considering the popularity of mobile devices that supports biometric operations, traditional AKA protocols are integrated with the third authentication factor (i.e., biometric) to achieve higher security [20], [25], [26], [27], [28]. Biometrics belong to "something the user is" and are the invariable physiological characteristics (such as fingerprint, face and iris) that people have. It is usually called three-factor AKA protocol when it is used together with password and smart card. Before the password is verified, the user is required to provide her biological information. Compared with single-factor AKA protocols and two-factor AKA protocols, three-factor AKA protocols are considered to have certain advantages in security. Recently, AKA protocols based on three-factor have become a hot research topic. However, in many three-factor AKA protocols [2], [3], [4], [22], [29], three-factor security and other security goals actually cannot be fulfilled. Therefore, to make better usage of the advantage of biometrics, it is very important to construct an AKA protocol for mobile environments that can provide truly three-factor security while maintaining desirable usability properties.

1.1 Related Work

In 2013, Guo and Chang [26] put forward a two-factor AKA protocol by employing the chaos theory. But in 2015, Lin [30] found that Guo and Chang's protocol can not provide anonymity and session-key security. Further, Lin presented an improved protocol to address these limitations [30]. After two years, Lee *et al.* [20] remarked that Lin's protocol can not achieve the security requirements as claimed: Lin's protocol is vulnerable to DoS attacks, internal enemy attacks and cannot ensure secure key agreement. To eliminate the defects of Lin's protocol, Lee *et al.* put forward an enhanced two-factor AKA protocol while employing extended chaotic-maps that is the extended version of chaotic-map in interval $(-\infty, +\infty)$. Lee *et al.* [20] argued that their solution is secured against all known attacks. But, we find that Lee *et al.*'s protocol cannot resist the temporary information leakage attack, key compromise impersonation attack and does not provide important usability properties such as local password update (i.e., the criterion C2 in [31]) and timely false password detection (i.e., the criterion C9 in [31]).

In 2014, Islam [32] designed a three-factor AKA protocol based on extended chaotic-mapping and remarked that his protocol provides security against various known attacks including smart card loss attacks and upholds the three-factor security characteristics. Despite of such claims, in 2016, Jiang *et al.* [23] revealed that Islam's proposal can not provide false password timely inspection functions and is prone to off-line password guessing attack, smart card loss attack, and biological sample leaks. To cope the mentioned limitations, Jiang *et al.* [23] presented a new three-factor AKA protocol by using the fuzzy verification method and proved that the scheme does not only resist all kinds of attacks but also holds essential security features. Nevertheless, we remark that Jiang *et al.*'s proposal is prone to key compromise user impersonation attack and clock-synchronization attack.

In 2015, on the basis of Lee's work [33], Liu *et al.* [21] analyzed the AKA protocol using chaotic-maps that are presented by Guo *et al.* [34]. The authors revealed various security limitations of Guo *et al.*'s protocol: it cannot provide mutual authentication and is vulnerable to denial of service (DoS) attack, replay attack and guessing attack. Later, Liu *et al.* [21] put forward an improved two-party AKA protocol. Liu *et al.* remarked that their presented protocol could resist all known attacks. On the contrary, in 2016, Chen *et al.* [35] pointed out that the proposal of Liu *et al.* [21] is unable to resist the off-line password-guessing-attack, failing to achieve truly three-factor security.

In 2017, Wazid *et al.* [2] proposed a three-factor user authentication protocol for renewable-energy-based smart grid environment. We observed that, this protocol only uses the lightweight cryptographic computations to improve efficiency and the verifier parameter CI_i^* can be used to perform off-line password guessing. Therefore, this protocol can neither resist off-line password guessing attack nor provide perfect forward secrecy and three-factor security. In 2018, Roy *et al.* [3] designed a chaotic map-based anonymous authentication protocol with fuzzy extractor for crowdsourcing Internet of Things whose verifier parameter f_i is also vulnerable to be performed off-line password guessing. In the same year, Islam *et al.* [4] also proposed a provably secure three-factor protocol for multimedia big data communications. Similar defects make the protocol of Islam *et al.* [4] vulnerable to off-line guessing attack and key compromised user impersonation attack and unable to provide functionality and security features such as three-factor security.

In 2018, Chatterjee *et al.* [10] used the Chebyshev chaotic map, cryptographic hash function and symmetric key encryption/decryption to construct a three-factor AKA protocol for multi-server environments. Nevertheless, it can be observed that their protocol cannot achieve truly three-factor security: the explicit password verifier A_i enables guessing attackers to off-line guess passwords and identities when the other two factors (i.e., biometric and smart card) are leaked. Note that, truly three-factor security is the most essential goal of a three-factor AKA protocol.

1.2 Motivations and Contribution

To be practical, a three-factor AKA protocol based on chaotic-maps should be able to provide comprehensive security goals and various desirable properties. Most essentially, a three-factor AKA protocol must satisfy at least five security goals: (1) Truly three-factor security, that is, if an attacker gains any two of the three authentication factors (i.e., password, smart card and biometric), the attacker can not successfully figure out the third factor; (2) Anonymity and un-traceability, including identity protection and user un-traceability; (3) Resistance against password guessing attacks, including off-line guessing attacks and on-line guessing attacks; (4) Session-key security, that is, the attacker can not steal or calculate the session key negotiated between the user and the server, which includes that the protocol should provide perfect forward security, and can resist session-specific temporary information leakage attacks; (5) Resistance to impersonation attacks, including user impersonation attacks, server impersonation attacks, and key-compromise impersonation attacks.

However, existing state-of-the-art three-factor AKA protocols can not satisfy at least one of the above five security goals. For example, the protocols in [2] and [29] can not provide truly three-factor security, the protocol in [21] can not provide user un-traceability, the protocols in [29] and [3] can not resist off-line guessing attacks and session-key security, and the protocol in [22] can not resist counterfeiting attacks. Shin and Kobara [36] pointed out that any passive/active attacker can find out the client's password and the static key by utilizing off-line guessing attacks against the IEEE 1363.2 Standard [37] and its multi-server system [38].

Another defect of existing state-of-the-art three-factor AKA protocols is that most of them assume that passwords follow the uniform distribution when formulating the advantages of the attacker. However, Wang *et al.* [31], [39] have proved that user-chosen passwords follow the Zipf's law, a vastly different distribution from the widely used (but unrealistic) uniform distribution. It illustrates that an AKA protocol with passwords of the Zipf's distribution is inherently difficult to resist both online and offline password guessing attacks, and a proper security formulation for the advantages of the attacker shall be used. To solve the common security problems in existing three-factor AKA protocols, we put forward a secure three-factor AKA protocol by combining biometric and chaotic-maps for mobile lightweight devices to secure data communications.

In all, this paper makes the following key contributions:

1) In order to resist password guessing attacks, existing three-factor AKA protocols generally adopt two ways. The first way is that in the login phase, the protocol does not provide the usability property of timely password typo detection. The other way is that password verification parameters are stored in smart cards. Following the first way, the protocol can avoid the password guessing attacks caused by the leakage of the password verifier, but it brings additional computational burden and the risk of denial of service attacks to the server. Most protocols follows the second way, but they cannot effectively resist password guessing attacks once the password verifier is leaked. Fortunately, in this paper, we use the technique of "Fuzzy-Verifiers" [39] to construct a fuzzy password verifier, and then we combine the notion of "Honeywords" [39] to design a robust three-factor AKA protocol based on extended chaotic-maps and biometrics, which can effectively resist password guessing attack.
2) We prove that the proposed protocol is semantically secure under the random oracle model based on chaotic-maps computational Diffie-Hellman problem, demonstrate how the protocol satisfies the 13 security evaluation criteria and simulate the protocol by making use of the AVISPA automated tool.
3) We perform a comparative analysis of the security, computation and communication cost of the proposed protocol with seven state-of-the-art related protocols, e.g., Wazid *et al.* [2] (IEEE Trans. II'17), Roy *et al.* [3] (IEEE IoTJ'18) and Islam *et al.* [4] (IEEE IoTJ'18). Comparison results demonstrate that the proposed protocol for mobile lightweight devices can achieve a better balance between security and usability.

1.3 Roadmap of This Paper

The roadmap of this paper is as follows: Section 2 introduces necessary preliminaries, including the adversary model and the 13 evaluation criteria for three-factor AKA protocols. In Section 3, we present our new protocol. The formal security proof and heuristic security arguments are presented in Sections 4.1 and 4.2, respectively. Section 5 summarizes the comparison of security, communication and computation cost. Section 6 concludes the paper.

TABLE 1
Notations and abbreviations

Notation	Description	Notation	Description	Notation	Description
S	Server	v	Random number of S	u	Random numbers of U_i
U_i	User	s	Private key of S	$H(\cdot)$, $H_0(\cdot)$	One-way hash-function
ID_i, PW_i	Identity, password of U_i	$\mathcal{A}$	Malicious adversary	SK	Session key between U_i and S
p	Large prime	$\|\|$	String concatenation operation	$GEN(\cdot)$	Biometric key extraction algorithm
$T_s(x)$	Public key of S	$\oplus$	Bitwise XOR operation	$REP(\cdot)$	Biometric key reproduction algorithm

2 Preliminaries

In this section, we present some relevant preliminaries necessary for the understanding of the remaining of this paper, as well as the evaluation criteria. A summary of notations used in this paper is presented in Table 1.

2.1 Extended Chebyshev Chaotic-Maps ([40], [41])

Definition 1 (Chebyshev chaotic-maps). *Assume $n \in Z^+$ and $x \in [-1,1]$ are real numbers, $T_n(x) : [-1,1] \to [-1,1]$ denotes the Chebyshev polynomial and is formulated as $T_n(x) = cos(n \cdot cos^{-1}(x))$, while the recurrence relation is presented as $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$, where $n \geqslant 2$, $T_0(x) = 1$ and $T_1(x) = x$.*

Proposition 1. *The Chebyshev polynomial is closed under the semigroup property, that is: $T_r(T_s(x)) = cos(r \cdot cos^{-1}(cos(s \cdot cos^{-1}(x)))) = cos(rs \cdot cos^{-1}(x)) = T_{sr}(x)$, where $s, r \in Z^+$ and $x \in [-1,1]$.*

Bergamo *et al.* [42] pointed out that Chebyshev polynomial map based public key cryptography is not secure. Three years later in 2008, Zhang [41] showed that Chebyshev polynomial map holds the semigroup property on the interval $(-\infty, +\infty)$, and Chebyshev polynomial map can be extended as $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \bmod p$, where p refers to large prime and $x \in (-\infty, +\infty)$. The aforementioned polynomials are called the extended Chebyshev polynomials that are closed under semigroup property.

Definition 2 (CMDLP). *For a given real number $x \in [-\infty, +\infty]$ and an extended Chebyshev polynomial y, the* ***Chaotic-maps discrete logarithm problem*** *states that it is hard for a PPT adversary $\mathcal{A}$ to compute r (the secret integer) so that $y = T_r(x) \bmod p$. That is, for any PPT adversary $\mathcal{A}$ within a bounded time t, the probability $Adv_{\mathcal{A}}^{CMDLP}(t) = Pr[\mathcal{A}(x,y) = r : r \in \mathbb{Z}_p^*, y = T_r(x) \bmod p]$ is negligible, i.e., $Adv_{\mathcal{A}}^{CMDL}(t) \leqslant \epsilon$, where ϵ refers to some negligible function.*

Definition 3 (CMCDHP). *For a given real number $x \in [-\infty, +\infty]$ and two extended Chebyshev polynomials $T_r(x) \bmod p$ and $T_s(x) \bmod p$, the* Chaotic-maps computational Diffie-Hellman problem *states that it is hard for a PPT $\mathcal{A}$ to compute $T_{sr}(x) \bmod p$. That is, for any PPT $\mathcal{A}$ within a bounded time t, the probability $Adv_{\mathcal{A}}^{CMCDHP}(t) = Pr[\mathcal{A}(x, T_s(x) \bmod p, T_r(x) \bmod p) = T_{sr}(x) \bmod p : s, r \in \mathbb{Z}_p^*]$ is negligible, i.e., $Adv_{\mathcal{A}}^{CMDHP}(t) \leqslant \epsilon$, where ϵ refers to some negligible function.*

2.2 Fuzzy Extractor of Biometrics

Given a biometric input BIO, the fuzzy extractor extracts a nearly random string σ in an error-tolerant manner. If BIO' can be considered as the change, but closely related to BIO, then the extracted σ remains the same due to auxiliary string τ. A fuzzy extractor comprises the following two phases [23], [43]: (1). $GEN(BIO) = (\sigma, \tau)$. GEN refers to a probabilistic generator that outputs an auxiliary string τ and an extracted string σ, when given a biometric input BIO. (2). $REP(BIO', \tau) = \sigma$, If BIO' is reasonably close to BIO, then REP refers to the deterministic reproduction procedure that allows one to recover σ from the corresponding auxiliary string τ and BIO'.

2.3 Adversary Model

Based on existing adversary models, particularly those presented in [31], [39], [44], [45], [46], [47], [48], [49], [50], [51], we summarize the capacities of the adversary for three-factor AKA protocols to be as follows:

1) The adversary $\mathcal{A}$ can reveal all parameters stored in the smart-card of the user, using power analysis or other side-channel techniques.
2) $\mathcal{A}$ can block, intercept, modify, delete, and resend any message sent over the open channel.
3) $\mathcal{A}$ can offline enumerate all pairs of (PW_i, ID_i) from $(\mathcal{D}_{PW}, \mathcal{D}_{ID})$ within polynomial time, where $\mathcal{D}_{ID}$ and $\mathcal{D}_{PW}$ refer to the space of identities and passwords, respectively. In fact, the space of identity and password is very limited in real life, according to [48] and [49], $|\mathcal{D}_{ID}| \leq |\mathcal{D}_{PW}| \leq 10^6$.
4) If anyone can register as a legitimate user, then $\mathcal{A}$ can also be registered.
5) $\mathcal{A}$ can steal or copy one's credentials through any biometric based terminal.
6) Improper erasure may result in $\mathcal{A}$ obtaining previous session keys (e.g., using digital forensic techniques).
7) In three-factor AKA protocols, an adversary $\mathcal{A}$ can obtain any two of the three factors, but not all three at the same time [51].
8) When evaluating the perfect forward secrecy or key-compromise impersonation attack, $\mathcal{A}$ is assumed to have obtained the long-term private key of the server.

2.4 Security Evaluation Criteria

The key to evaluate goodness of an AKA protocol is to see whether the design of the protocol meets the basic principles of AKA protocol, that is, security and practicability. For practicality, it mainly depends on whether the computational and communication consumptions of protocol execution are optimal within the allowable range of application environment. For security, it is necessary to determine whether the protocol meets the security requirements of AKA protocols in specific environments. The security

TABLE 2
Security Evaluation Criteria for Three-Factor AKA Protocols (Adapted From [39], [53])

Notation	Term
EC_1	User Anonymity and Un-Traceability
EC_2	Password-Verifier-Table is unwanted
EC_3	Password and Biometric Friendliness
EC_4	Password Exposure is avoidable
EC_5	Timely Typo Detection
EC_6	No smart-card Loss Attack
EC_7	Resistance to Known Attacks
EC_8	Key-Agreement provision
EC_9	Mutual-Authentication verification
EC_{10}	No Clock Synchronization
EC_{11}	Sound-Repairability
EC_{12}	Three-Factor Security
EC_{13}	Perfect Forward Secrecy

objectives of AKA protocols are different in literature [27], [39], [44], [52], [53], [54]. For example, HMQV [52] pointed out that key-compromised user impersonation attacks should be considered.

Wang *et al.* [39] proposed 12 security indicators for two-factor AKA protocols, and Wang *et al.* [53] summarized the security indicators for AKA protocols in wireless sensor networks (WSN). Table 2 (adapted from [39] and [53]) refers to the security indicators proposed by these protocols [27], [39], [44], [52], [53], [54] to cover the security requirements of single-factor, two-factor ([14], [28]) and three-factor AKA protocols. Nevertheless, when analyzing specific AKA protocols, we should treat them differently according to the number of factors, the characteristics of cryptography and the application environment. For example, when analyzing the security of a single-factor AKA protocol or a double-factor AKA protocol, we do not consider three-factor security, which is the inherent security characteristic of three-factor AKA protocols.

It should be noted that in three-factor AKA protocols, password and biometric friendliness (EC_3) means that users are allowed to choose passwords and biometric features freely and update these factors efficiently and locally. No smart-card loss attack (EC_6) means that the protocol should be able to resist the smart card loss attack, that is, anyone who obtains a legitimate user's smart card should not be able to change or restore the password through online, offline or hybrid guessing techniques within polynomial time, or imitate the user to login and execute the protocol, even if the user's biometrics is compromised. It should be noted that resistance to known attacks (EC_7) should include session-specific temporary information attacks and (key-compromised) user impersonation attacks [52]. Moreover, EC_7 also includes off/on-line password guessing attack, privileged insider attack, de-synchronization attack, replay attack, server impersonation attack and man-in-the-middle attack, etc. Three-factor security (EC_{12}) means that for any three-factor AKA protocol, even if any two of the three factors are leaked, but not all, it should resist all known attacks. Undetectable online guessing attack refers to cases where failed online guesses cannot be distinguished, detected and logged by the server (who cannot obtain the identity of the victim user). Detectable online guessing attack refers to cases where the server can find out who the communicating party is. In order to resist detectable online guessing attacks, it is recommended to set a limit on the times of user login request message validation errors (preset threshold value) on the server side. When analyzing online guessing attacks, we assume that the server is legal and reliable, acting as a password verification oracle. It should be noted that the online guessing attack in this paper are the detectable online guessing attack.

3 Proposed Protocol

To meet security needs of AKA protocol for mobile users in Client-Server environment , in our protocol we adopt at least the following six approaches:

1) In order to prevent attacks from privileged insiders, in the registration phase, we only pass ID_i to the server, construct password login verifiers by using the fuzzy verification technology [39], and produce random numbers of biometrics by using the fuzzy extractor [43].
2) To resist password guessing attack, on the one hand, we use "Fuzzy-Verifiers" to provide the resilience to offline password guessing attack [39] and to ensure timely password typo detection, efficient password update and three-factor security, where $W_i = H((H(ID_i) \oplus VPW_i) \bmod n_0)$ is the key parameter. On the other hand, we set "Honeywords" [39] to resist online password guessing attack.
3) According to [25], to ensure perfect forward secrecy and efficiency, we employ Chaotic-maps (from the public key cryptosystem domain).
4) In order to provide security against key compromise user impersonation attacks, we propose storing a secret parameter r_i securely at the server side (e.g., stored in an auxiliary server as with [9]).
5) We present a dynamic identity approach by employing a public key algorithm to provide user un-traceability, i.e., $C_3 = (ID_i||B_0) \oplus H_0(C_2)$.
6) The session key will be computed as $SK = H(ID_i||CID_i||B_0||B_0^{new}||T_u(T_s(x))||T_v(T_u(x)))$, instead of $T_u(T_v(x))$ to resist session-specific temporary information attacks. In addition, our scheme provides comprehensive features including revocation and re-registration.

In all, the proposed three-factor AKA protocol meets all 13 evaluation indicators EC_1-EC_{13} outlined in Table 2. The protocol comprises six phases, namely: setup, registration (as depicted in Fig. 1), login and authentication (Fig. 2), password or biometrics updating (Fig. 3), revocation and re-registration.

3.1 System Setup Phase

The server S randomly selects a number $s \in Z_p^*$ as well as two one-way hash-functions $H(\cdot)$ (SHA-160) and $H_0(\cdot)$ (SHA-320). Then, S calculates the public key $T_s(x)$, publicizes these parameters $\{T_s(x), x, H(\cdot), H_0(\cdot)\}$, and keeps a long private key s as a secret.

3.2 Registration Phase

In order to resist privileged insider attack, we only send identity ID_i to the server S.

User (U_i)	Secure Channel	**Server** (S)
Registration Phase: Chooses ID_i	$\xrightarrow{\{ID_i\}}$	Generates two random numbers $e_i, r_i \in Z_p^*$ Computes $CID_i = H(ID_i \Vert e_i)$ $B_0 = H(ID_i \Vert s \Vert r_i \Vert CID_i)$ Store $\{ID_i, r_i, Honey\_List = 0\}$ in database
	$\xleftarrow{SC_i, B_0}$	New smart card: $SC_i := \{T_s(x), x, CID_i, H(\cdot)\}$
Chooses PW_i, inputs BIO_i Computes $(\sigma_i, \tau_i) = GEN(BIO_i)$ $VPW_i = H(PW_i \Vert ID_i \Vert CID_i \Vert \sigma_i)$ Chooses an integer $2^4 \leq n_0 \leq 2^8$ $W_i = H((H(ID_i) \oplus VPW_i) \bmod n_0)$ $B_1 = B_0 \oplus VPW_i \oplus \sigma_i$ Update smart card: $SC_i := \{CID_i, B_1, W_i, \tau_i, T_s(x), x, n_0, H(\cdot), H_0(\cdot), GEN(), REP(,)\}$		

Fig. 1. User registration.

Step 1. The user U_i selects an ID_i and sends it to the server S.

Step 2. Upon getting $\{ID_i\}$, S picks $e_i, r_i \in_R Z_p^*$ and computes $CID_i = H(ID_i||e_i)$, $B_0 = H(ID_i||s||r_i||CID_i)$. S stores $\{ID_i, r_i, Honey\_List = Null\}$ in its database, inputs $\{T_s(x), x, CID_i\}$ to a new smart card SC_i and finally sends $\{SC_i, B_0\}$ to U_i.

Step 3. Upon receiving the smart card SC_i from the server S, the user U_i inputs her new password and fingerprints BIO_i into SC_i. Then, smart card SC_i randomly generates a number $2^4 \leq n_0 \leq 2^8$ and calculates some important parameters as follows: $GEN(BIO_i) = (\sigma_i, \tau_i)$, $VPW_i = H(PW_i|| ID_i||CID_i||\sigma_i)$, $W_i = H((H(ID_i) \oplus VPW_i) \bmod n_0)$, $B_1 = B_0 \oplus VPW_i \oplus \sigma_i$. Finally, the smart card SC_i contains the following parameters: $\{CID_i, B_1, W_i, \tau_i, T_s(x), x, n_0\}$ and $\{H(\cdot), H_0(\cdot), GEN(\cdot), REP(\cdot)\}$.

Remark 1. Traditionally, the *Honey_List* contains all the honeywords (seemingly like the real password but not correct) that have been tried by the attacker [55], [56]. In this work, the *Honey_List* contains all the honey parameters B_0' (seemingly like the real parameters $B_0 = H(ID_i||s||r_i||CID_i)$ but not correct) that have been tried by the attacker: whenever the honey parameters B_0' is being tried, it indicates that the attacker has extracted the password verifier W_i stored in the user's smart card. See more detailed analysis in Section 4.2.6.

User (U_i)	Public Channel	**Sever** (S)
Login and Authentication Phase: Inserts SC_i and inputs ID_i, PW_i, BIO_i^* Cpmputes $\sigma_i = REP(BIO_i^*, \tau_i)$ $VPW_i = H(PW_i \Vert ID_i \Vert CID_i \Vert \sigma_i)$ $W_i = H((H(ID_i) \oplus VPW_i) \bmod n_0)$ Checks $W_i^* \stackrel{?}{=} W_i$ - - - - - - - - Computes $B_0 = B_1 \oplus VPW_i \oplus \sigma_i$ Generates a random number u Computes $C_1 = T_u(x), C_2 = T_u(T_s(x))$ $C_3 = (ID_i \Vert B_0) \oplus H_0(C_2)$ $C_4 = H(ID_i \Vert CID_i \Vert B_0 \Vert C_2 \Vert C_3)$	$\xrightarrow{\{CID_i, C_1, C_3, C_4\}}$	Computes $C_2 = T_s(C_1)$ $(ID_i' \Vert B_0') = C_3 \oplus H_0(C_2)$ Checks the validity of ID_i' $B_0 = H(ID_i \Vert s \Vert r_i \Vert CID_i)$ Checks $C_4 \stackrel{?}{=} H(ID_i \Vert CID_i \Vert B_0 \Vert C_2 \Vert C_3)$ Checks $B_0' \stackrel{?}{=} B_0$ Generates a random number e_i^{new} $CID_i^{new} = H(ID_i \Vert e_i^{new})$ $B_0^{new} = H(ID_i \Vert s \Vert r_i \Vert CID_i^{new})$ Updates $\{ID_i, Honey\_List\}$ - - - - - - - - Generates a random number v Computes $C_5 = T_v(x), C_6 = T_v(C_1)$ $C_7 = (CID_i^{new} \Vert B_0^{new}) \oplus H(C_6 \Vert B_0)$ $SK_s = H(ID_i \Vert CID_i \Vert B_0 \Vert B_0^{new} \Vert C_2 \Vert C_6)$ $C_8 = H(ID_i \Vert CID_i \Vert B_0 \Vert B_0^{new} \Vert C_2 \Vert SK_s)$
Computes $C_6 = T_u(C_5)$ $(CID_i^{new} \Vert B_0^{new}) = C_7 \oplus H(C_6 \Vert B_0)$ $SK_u = H(ID_i \Vert CID_i \Vert B_0 \Vert B_0^{new} \Vert C_2 \Vert C_6)$ $C_8^* = H(ID_i \Vert CID_i \Vert B_0 \Vert B_0^{new} \Vert C_2 \Vert SK_u)$ Checks $C_8^* \stackrel{?}{=} C_8$ - - - - - - - - Chooses an integer $2^4 \leq n_0^{new} \leq 2^8$ $VPW_i^{new} = H(PW_i \Vert ID_i \Vert CID_i^{new} \Vert \sigma_i)$ $W_i^{new} = H((H(ID_i) \oplus VPW_i^{new}) \bmod n_0^{new})$ $B_1^{new} = B_0^{new} \oplus VPW_i^{new} \oplus \sigma_i$ Replaces $\{CID_i, W_i, B_1, n_0\}$ with $\{CID_i^{new}, W_i^{new}, B_1^{new}, n_0^{new}\}$	$\xleftarrow{\{C_5, C_7, C_8\}}$	$\boxed{\text{The common session key: } SK = SK_u = SK_s}$

Fig. 2. Login and authentication.

User (U_i)	Secure Channel	**Smart Card** (SC_i)
Updating Phase: Inputs ID_i, PW_i, BIO_i	$\xrightarrow{\{ID_i, PW_i, BIO_i, Updating\_Request\}}$	Cpmputes $\sigma_i = REP(BIO_i^*, \tau_i)$ $VPW_i = H(PW_i \Vert ID_i \Vert CID_i \Vert \sigma_i)$ $W_i = H((H(ID_i) \oplus VPW_i) \bmod n_0)$ Checks $W_i^* \stackrel{?}{=} W_i$ If holds, accepts "*Updating_requst*" Otherwise, rejects.
Only password is updated: Inputs a new PW_i^{new}	$\xrightarrow{\{PW_i^{new}\}}$	Computes $VPW_i^{new} = H(PW_i^{new} \Vert ID_i \Vert CID_i \Vert \sigma_i)$ Chooses a new integer $2^4 \leq n_0^{new} \leq 2^8$ Computes $W_i^{new} = H((H(ID_i) \oplus VPW_i^{new}) \bmod n_0^{new})$ $B_1^{new} = B_1 \oplus VPW_i \oplus VPW_i^{new}$ Replaces $\{B_1, W_i, n_0\}$ with $\{B_1^{new}, W_i^{new}, n_0^{new}\}$
Only biometrics is updated: Inputs a new BIO_i^{new}	$\xrightarrow{BIO_i^{new}}$	Computes $(\sigma_i^{new}, \tau_i^{new}) = GEN(BIO_i^{new})$ $VPW_i^{new} = H(PW_i \Vert ID_i \Vert CID_i \Vert \sigma_i^{new})$ Chooses a new integer $2^4 \leq n_0^{new} \leq 2^8$ Computes $W_i^{new} = H((H(ID_i) \oplus VPW_i^{new}) \bmod n_0^{new})$ $B_1^{new} = B_1 \oplus VPW_i \oplus VPW_i^{new}$ Replaces $\{B_1, W_i, \tau_i, n_0\}$ with $\{B_1^{new}, W_i^{new}, \tau_i^{new}, n_0^{new}\}$
Both password and biometrics are updated: Inputs PW_i^{new}, BIO_i^{new}	$\xrightarrow{PW_i^{new}, BIO_i^{new}}$	Computes $(\sigma_i^{new}, \tau_i^{new}) = GEN(BIO_i^{new})$ $VPW_i^{new} = H(PW_i^{new} \Vert ID_i \Vert CID_i \Vert \sigma_i^{new})$ Chooses a new integer $2^4 \leq n_0^{new} \leq 2^8$ Computes $W_i^{new} = H((H(ID_i) \oplus VPW_i^{new}) \bmod n_0^{new})$ $B_1^{new} = B_1 \oplus VPW_i \oplus VPW_i^{new}$ Replaces $\{B_1, W_i, \tau_i, n_0\}$ with $\{B_1^{new}, W_i^{new}, \tau_i^{new}, n_0^{new}\}$

Fig. 3. Password or biometrics updating.

3.3 Login and Mutual Authentication Phase

After the user U_i is registered with the server S successfully, she transmits the login request to S when she wishes to obtain some service – see below:

Step 1. U_i inputs the smart card SC_i into a card reader, and provides ID_i, PW_i and BIO_i^* to SC_i. Then, SC_i computes $\sigma_i = REP(BIO_i^*, \tau_i)$, $VPW_i = H(PW_i||ID_i||CID_i||\sigma_i)$ and checks whether $W_i \stackrel{?}{=} H((H(ID_i) \oplus VPW_i) \bmod n_0)$. If not, SC_i rejects the login request. Otherwise, SC_i computes $B_0 = B_1 \oplus VPW_i \oplus \sigma_i$. Subsequently, SC_i picks $u \in_R Z_p^*$ and computes $C_1 = T_u(x)$, $C_2 = T_u(T_s(x))$, $C_3 = (ID_i||B_0) \oplus H_0(C_2)$, $C_4 = H(ID_i||CID_i||B_0||C_2||C_3)$. Finally, SC_i sends $\{CID_i, C_1, C_3, C_4\}$ to S.

Step 2. After obtaining $\{CID_i, C_1, C_3, C_4\}$, S calculates $C_2 = T_s(C_1)$, $(ID_i'||B_0') = C_3 \oplus H_0(C_2)$. Then, S searches $\{ID_i, r_i, Honey\_List\}$ in its database. If ID_i' cannot be searched, the session is terminated. Otherwise, S proceeds to the next step. S computes $B_0 = H(ID_i||s||r_i||CID_i)$ and checks whether $C_4 \stackrel{?}{=} H(ID_i||CID_i||B_0||C_2||C_3)$. If they are unequal, S terminates this session. Otherwise, S checks whether the derived B_0' equals the computed B_0. If they are equal, S proceeds to the next step. If they are unequal, S knows that U_i's smart card has been corrupted and the adversary did not get the real password. Accordingly, S inserts the honeyword B_0' into *Honey_List* and wraps this login request. Moreover, if $|Honey\_List| >= N_0$ (Such as the threshold $N_0 = 5$), where N_0 is a threshold value, S suspends the use of SC_i until U_i re-registers and requests to restore SC_i. Otherwise, S picks $e_i^{new} \in_R Z_p^*$ and calculates $CID_i^{new} = H(ID_i||e_i^{new})$, $B_0^{new} = H(ID_i||s||r_i||CID_i^{new})$. Subsequently, S updates $\{ID_i, Honey\_List = Null\}$ in its backend database. Moreover, S picks $v \in_R Z_p^*$ and computes

$C_5 = T_v(x)$, $C_6 = T_v(C_1)$, $C_7 = (CID_i^{new}||B_0^{new}) \oplus H(C_6||B_0)$, $SK_s = H(ID_i||CID_i||B_0||B_0^{new}||C_2||C_6)$ and $C_8 = H(ID_i||CID_i||B_0||C_2||C_5||SK_s)$. Lastly, S sends $\{C_5, C_7, C_8\}$ to SC_i.

Step 3. On receiving the message $\{C_5, C_7, C_8\}$, SC_i computes $C_6 = T_u(C_5)$, $(CID_i^{new}||B_0^{new}) = C_7 \oplus H(C_6||B_0)$, $SK_u = H(ID_i||CID_i||B_0||B_0^{new}||C_2||C_6)$, and verifies $C_8 \stackrel{?}{=} H(ID_i||CID_i||B_0||C_2||C_5||SK_u)$. If not, SC_i aborts this session. Otherwise, SC_i considers a shared key $SK = SK_u = SK_s$ is being shared with S. Subsequently, SC_i randomly generates a number $2^4 \leq n_0^{new} \leq 2^8$ and calculates $VPW_i^{new} = H(PW_i||ID_i||CID_i^{new}||\sigma_i)$, $W_i^{new} = H((H(ID_i) \oplus VPW_i^{new}) \bmod n_0^{new})$, and $B_1^{new} = B_0^{new} \oplus VPW_i^{new} \oplus \sigma_i$. Finally, the smart card SC_i replaces $\{CID_i, W_i, B_1, n_0\}$ with $\{CID_i^{new}, W_i^{new}, B_1^{new}, n_0^{new}\}$.

3.4 Password or Biometrics Update Phase

U_i injects her smart-card into the card reader, and provides ID_i, PW_i and BIO_i^*. Then, SC_i calculates $\sigma_i = REP(BIO_i^*, \tau_i)$, $VPW_i = H(PW_i||ID_i||CID_i||\sigma_i)$ and checks whether $W_i \stackrel{?}{=} H((H(ID_i) \oplus VPW_i) \bmod n_0)$. If not, SC_i declines this update request. Otherwise, SC_i acknowledges this update request.

Case I. If U_i only wants to change the password, then she inputs a new password PW_i^{new}. SC_i will then randomly generate $2^4 \leq n_0' \leq 2^8$ and calculates $VPW_i^{new} = H(PW_i^{new}||ID_i||CID_i||\sigma_i)$, $W_i^{new} = H((H(ID_i) \oplus VPW_i^{new}) \bmod n_0')$ and $B_1^{new} = (B_1 \oplus VPW_i \oplus \sigma_i) \oplus VPW_i^{new} \oplus \sigma_i$. Finally, SC_i replaces $\{W_i, B_1, n_0\}$ with $\{W_i^{new}, B_1^{new}, n_0'\}$.

Case II. If U_i only wants to change the biometrics, then she inputs a new biometrics BIO_i^{new}. Then, SC_i randomly generates a number $2^4 \leq n_0' \leq 2^8$ and calculates $GEN(BIO_i^{new}) = (\sigma_i^{new}, \tau_i^{new})$, $VPW_i^{new} = H(PW_i||ID_i|| CID_i||\sigma_i^{new})$, $W_i^{new} = H((H(ID_i) \oplus VPW_i^{new}) \bmod n_0')$ and $B_1^{new} = (B_1 \oplus VPW_i \oplus \sigma_i) \oplus VPW_i^{new} \oplus \sigma_i^{new}$. Finally, SC_i replaces $\{W_i, B_1, \tau_i, n_0\}$ with $\{W_i^{new}, B_1^{new}, \tau_i^{new}, n_0'\}$.

Case III. If U_i wants to change both her biometrics and the password simultaneously, then she inputs a new password PW_i^{new} and a new biometrics BIO_i^{new}. Subsequently, SC_i randomly generates a number $2^4 \leq n_0' \leq 2^8$ and calculates $GEN(BIO_i^{new}) = (\sigma_i^{new}, \tau_i^{new})$, $VPW_i^{new} = H(PW_i^{new}||ID_i||CID_i||\sigma_i^{new})$, $W_i^{new} = H((H(ID_i) \oplus VPW_i^{new}) \bmod n_0')$ and $B_1^{new} = (B_1 \oplus VPW_i \oplus \sigma_i) \oplus VPW_i^{new} \oplus \sigma_i^{new}$. Finally, SC_i replaces $\{W_i, B_1, \tau_i, n_0\}$ with $\{W_i^{new}, B_1^{new}, \tau_i^{new}, n_0'\}$.

3.5 Revocation Phase

In this subsection, we provide a feature that allows the user to block a stolen or misplaced smart card:

Step 1. U_i verifies the authentication of smart card is similar to the login phase. If SC_i authenticates U_i as a legitimate user, SC_i sends the revocation-request $\{CID_i, C_1, C_3, C_4, Revoke\_request\}$ to the server.

Step 2. Upon getting the revocation-request, S authenticates SC_i by verifying C_4. If it is found to be invalid, S denies this revocation-request. Otherwise, S sets $|Honey\_List| >= N_0$ such that SC_i is revoked. Finally, SC_i is suspended until U_i re-registers.

3.6 Re-Registration Phase

If a legitimate user's smart-card is revoked or the number of times that the user cannot be authenticated by the server S exceeds the maximum threshold value, then U_i is required to re-register. In this case, U_i will not be able to log into the system even though she inputs the correct values $\{ID_i, PW_i, BIO_i\}$. However, U_i can re-register by performing the following steps:

Step 1. U_i submits the re-registration request $\{ID_i, Re-register\_request\}$ to S through a private channel.

Step 2. Upon receiving the request message from U_i, the server S uniquely identifies the user U_i by checking her identity information such as her SSN, national card number, or some other relevant legal identity documents issued by the government. Afterwards, if ID_i is found in the database, S confirms whether $|Honey\_List| >= N_0$. And if SC_i is found to be revoked, then S accepts this request and performs Steps 2 and 3 of the Registration phase to complete re-registration. Otherwise, S rejects this request.

Remark 2. In the login and authentication phase, if the unexpected termination of the protocol occurs before the second message reaches the user, in next login phase the user still can authenticate at the server although B_0 is not updated at the user. On the other hand, since CID_i updates after each successful login authentication and key agreement, if the user uses the old B_0, then user untraceability can still be ensured. In addition, since we use random numbers when computing the session key replay attack (which will create the Denial-of-service to the legal user) is also impossible although the login request is valid forever. In summary, the reason why the above vulnerabilities will not occur is that the server side will not store B_0.

4 Security Analysis

In this section, we provide the formal security proof, heuristic security analysis and security simulation with AVISP for the proposed protocol. Among them, the security simulation based on AVISPA software can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TDSC.2020.3022797 and also at http://wangdingg.weebly.com/publications.html. For ease of description, we abbreviate the proposed protocol to CM3FAKAP.

4.1 Formal Security Proof

We will now formally prove the security of CM3FAKP under the CMCDH assumption in the random oracle model.

4.1.1 Security Model

The security of CM3FAKAP relies on the BPR2000 [57] and Bresson [58] models , and we also use the technique of Wang *et al.* [39] in our proof.

Participants. A three-factor AKAP involve two participants, namely: U and S. Each participant has a number of distinct instances called oracles. The ith instance of U and the jth instance of S are denoted by U^i and S^j, respectively. Besides, any instance is denoted as I.

Queries. The interaction between an adversary $\mathcal{A}$ and the protocol participants occurs only via oracle queries, which model the adversary capabilities in a real attack. The types of queries that can be used by $\mathcal{A}$ are as follows:

- $Execute(U^i, S^j)$: The passive eavesdropping of a protocol is captured by this query, and its output comprises all communication transcripts of the protocol between U^i and S^j.
- $Send(U^i, Start)$ This query represents the initialization of protocol $\mathcal{P}$.
- $Send(I^i, m)$: The active attacks are captured by this query. More precisely, $\mathcal{A}$ constructs a forged message m by blocking and intercepting a message. Afterwards, $\mathcal{A}$ sends m to instance I^i and obtains the response from I^i.
- $Reveal(I^i)$: The misuse of session keys is modelled by this query. It responds with I^i's SK to $\mathcal{A}$ if the instance I^i accepts the session and generates a session key SK. Otherwise, it returns $\perp$.
- $Corrupt(I^i, a)$: $\mathcal{A}$ can reveal any two of the three factors: password and biometric, password and smart-card, biometric and smart-card, but not all three simultaneously in order to evade the trivial case: (i). If $a = 1$ and $I = U$, then it reveals PW of U and all parameters held by the smart card. If $a = 2$ and $I = U$, then it reveals the biometric information of U and all parameters held by the smart card. If $a = 3$ and $I = U$, then it reveals the password and biometric information of U. (ii). If $a = 1$ and $I = S$, then it reveals the long-term private s of S.
- $Test(I^i)$: This query models the session key's semantic security. Upon receiving the query, a coin b is flipped. If $b = 0$, then a random key having the same size as that of SK is returned to $\mathcal{A}$. If $b = 1$, then SK is passed to $\mathcal{A}$. $\perp$ is returned to $\mathcal{A}$, if no SK for instance I^i is generated. During the adversary's execution time, this query can be called any time (but only once).

Partnering. Two instances U^i and S^j become partners if: (1) U^i and S^j are accepted; (2)U^i and S^j have the same session identifier (sid), i.e., $sid^i_U = sid^j_S$. (3) The partner identifier (pid) of S^j is U and the converse also holds.

Freshness. A user instance or a server instance is referred to as fresh, if (1) I has computed an accepted session key. (2) $\mathcal{A}$ or its partner does not make any Reveal-query I. (3) From the start of the game, $\mathcal{A}$ makes Corrupt-query at most once to I or to its partner.

Definition 4 (AKA-Security). *Suppose $Succ(\mathcal{A})$ refers to the event that $\mathcal{A}$ asks a $Test(U^i)$ query for some freshly accepted instances, and reveals b' as a guess bit against the bit b that was chosen for the Test-query. The advantage of an adversary for breaching the semantic security of P is presented as $Adv_p^{AKA}(\mathcal{A}) = |2Pr[Succ(\mathcal{A})] - 1| = |2Pr[b' = b] - 1|$. The CM3FAKAP achieves the $AKA - security$ if for any PPT adversary $\mathcal{A}$, the advantage $Adv^{AKA}_{CM3FAKAP}(\mathcal{A})$ is negligible.*

4.1.2 Security Proof

The Difference Lemma that is incorporated for our sequence of games is introduced in Lemma 1.

Lemma 1. *[59] Suppose E_1, E_2, F refer to the events defined in some probability distribution, and assume that $E_1 \wedge \neg F \Longleftrightarrow E_2 \wedge \neg F$. It implies that $|Pr[E_1] - Pr[E_2]| \leqslant Pr[F]$.*

Theorem 1. *Suppose that $Adv^{AKA}_{CM3FAKAP,\mathcal{D}}(\mathcal{A})$ is the probability of success of the PPT adversary $\mathcal{A}$ within a bounded time t to break the semantic security of the presented protocol. Assume that $\mathcal{A}$ asks q_h times Hash-queries, q_e times Execute-queries and q_s times Send-queries. Then,*

$$Adv^{AKA}_{CM3FAKAP,\mathcal{D}}(\mathcal{A}) \leqslant 2C' \cdot q_s^{s'} + \frac{(q_s + q_h + q_h^2)}{2^{l-1}} + \frac{2(q_s + q_e)^2}{p} + 2q_h Adv^{CMCDH}_{\mathcal{A}}(t'),$$

where $\mathcal{D}$ refers to password space that follows a Zipf's law [31] in terms of frequency distribution, s' and C' denote the Zipf parameters, l is the bit length of the output of the Hash function (e.g., $l = 160$), $t' \leqslant t + (q_s + q_e + 1)T_c$, T_c is the computational time required for the extended chaotic-maps.

Proof. Suppose that the adversary $\mathcal{A}$ is able to break the security of the CM3FAKAP, in this case, we design an algorithm $\mathcal{B}$ that solves the Chaotic-maps computational Diffie-Hellman problem (CMCDHP). More precisely, $\mathcal{B}$ responds $T_{\theta\varphi}(x)$ against the instance $(x, T_\theta(x), T_\varphi(x))$ of CMCDHP, where θ, φ are two integers. The proof is comprised of sequence of games: $E_0, E_1, \ldots, E_5$. Let E_i refer to the event that $\mathcal{A}$ outputs the correct b in E_i, where $(i = 0, 1, 2, 3, 4, 5)$.

Game E_0. In this experiment, the simulation of real attack is executed in the random oracle model. $\mathcal{A}$ has the access to all the oracles. Thus, we have

$$Adv^{AKA}_{CM3FAKAP,\mathcal{D}}(\mathcal{A}) = |2Pr[E_0] - 1|. \tag{1}$$

Game E_1. This experiment simulates the random oracle H by managing a hash list $\Lambda_{\mathcal{H}}$ and $\Lambda_{\mathcal{A}}$. Since all oracles are simulated as the real attack, therefore, this experiment cannot be distinguished from the actual execution of the protocol. Fig. 4 depicts all queries asked by $\mathcal{A}$. Thus, we have

$$|Pr[E_1] - Pr[E_0]| = 0.$$

Game E_2. This experiment simulates all kinds of queries just like in game E_1, except that the simulation is aborted in the following two cases: (1). Collisions on the output of hash queries, (2). Collisions on the partial transcripts:$((CID_i, C_1, C_3, C_4), (C_5, C_7, C_8))$. According to the birthday-paradox, we have

$$|Pr[E_2] - Pr[E_1]| \leqslant \frac{q_h^2}{2^{l+1}} + \frac{(q_s + q_e)^2}{2p}. \tag{2}$$

Game E_3. We simulate this experiment just like the prevalent game, the only difference is the termination of protocol in case of $\mathcal{A}$ correctly guesses the authentication elements C_4 and C_8 without querying the random oracle. This experiment is indistinguishable from the previous experiment except that the instance U^i (or S^j) refuses a correct authentication element. Therefore, we have

$$|Pr[E_3] - Pr[E_2]| \leqslant \frac{q_s}{2^l}. \tag{3}$$

Game E_4. In this experiment, the SK is guessed without querying the corresponding random oracle, i.e., the session key is independent from $K = T_{uv}(x)$ and $\mathcal{H}$. Accordingly, this experiment is indistinguishable from

•$Hash$-**query:** The answer is r for a hash query $\mathcal{H}(q)(\mathcal{H}_0(q))$ or $\mathcal{H}'(q)(\mathcal{H}'_0(q))$, if the record (q,r) or $(0,q,r)$ belongs to $\Lambda_{\mathcal{H}}$. Otherwise, r as an answer is defined as: randomly select an element $r \in \{0,1\}^l$, and (q,r) is added to $\Lambda_{\mathcal{H}}$. If the adversary directly asks the query, (q,r) or $(0,q,r)$ will be added to $\Lambda_{\mathcal{A}}$.

•$Send(U^i, Start)$**-query:** Randomly select a number $u \in \mathbb{Z}_p^*$ and calculate $C_1 = T_u(x), C_2 = T_u(T_s(x)), N_0 = B_1 \oplus VPW_i \oplus \sigma_i, C_3 = (ID_i||B_0) \oplus H_0(C_2)$ and $C_4 = H(ID_i||CID_i||B_0||C_2||C_3)$. Afterwards, the query is answered with (CID_i, C_1, C_3, C_4).
•$Send(S^j, (CID_i, C_1, C_3, C_4))$**-query:** Compute $C_2 = T_s(C_1)$, $(ID'_i||B'_0) = C_3 \oplus H_0(C_2)$, $B_0 = H(ID_i||s||r_i||CID_i)$ and $C_4^* = H(ID_i||CID_i||B_0||C_2||C_3)$.

- Reject if ID'_i cannot be searched.
- Reject if C_4^* is not equal to C_4.
- Reject if B'_0 is not equal to B_0.
- Inserts B'_0 into $Honey\_List$ if $|Honey\_List| < N_0$.
- Suspend SC_i if $|Honey\_List| >= N_0 \wedge (B'_0 \neq B_0)$.

Afterwards, generate a random number e_i^{new} and compute $CID_i^{new} = H(ID_i||e_i^{new})$, $B_0^{new} = H(ID_i||s||r_i||CID_i^{new})$. Choose a random number v and compute $C_5 = T_v(x), C_6 = T_v(C_1)$, $C_7 = (CID_i^{new}||B_0^{new}) \oplus H(C_6||B_0)$, $SK_s = H(ID_i||CID_i||B_0||B_i^{new}||C_2||C_6)$. Then, the query is answered with (C_5, C_7, C_8).
•$Send(U^i, (C_5, C_7, C_8))$**-query:** Compute $C_6 = T_\theta(C_5)$, $(CID_i^{new}||B_0^{new}) = C_7 \oplus H(C_6 \oplus B_0)$, $SK_u = H(ID_i||CID_i||B_0||B_i^{new}||C_2||C_6)$ and $C_8^* = H(ID_i||CID_i||B_0||B_i^{new}||C_2||SK_u)$. Reject if C_8 and C_8^* are not equal. Otherwise, generate a new integer $2^4 \leq n_0^{new} \leq 2^8$, and compute $VPW_i^{new} = H(PW_i||ID_i||CID_i^{new}||\sigma_i)$, $W_i^{new} = H((H(ID_i) \oplus VPW_i^{new}) \bmod n_0^{new})$ and $B_1^{new} = B_0^{new} \oplus VPW_i^{new} \oplus \sigma_i$. Replace (CID_i, W_i, B_i, n_0) with $(CID_i^{new}, W_i^{new}, B_i^{new}, n_0^{new})$, respectively.

•$Execute(U^i, S^j)$**-query:** The query employs the successive simulations of the Send-queries to process: $(CID_i, C_1, C_3, C_4) \leftarrow$ Sends $(U^i, Start)$, $(C_5, C_7, C_8) \leftarrow$ Sends $(S^j, (CID_i, C_1, C_3, C_4))$, and yields the transcript $((CID_i, C_1, C_3, C_4), (C_5, C_7, C_8))$.

•$Corrupt(U, a)$**-query:** The query responds with PW_i and $\{CID_i, B_1, W_i, \tau_i, T_s(x), x, n_0, H(\cdot), GEN().REP(,)\}$ stored in the smart-card in case if $a = 1$; returns the biometrics BIO_i and $\{CID_i, B_1, W_i, \tau_i, T_s(x), x, n_0, H(\cdot), GEN().REP(,)\}$ stored in the smart-card in case if $a = 2$; returns the password PW_i and biometrics BIO_i if $a = 3$.
•$Corrupt(S, a)$**-query:** Return the long-term private s of S if $a = 1$.

•$Reveal(I)$**-query:** The query responds with (SK_s) or (SK_u) calculated by the instance I (if the latter has accepted).

•$Test(I)$**-query:** The query first obtains SK from Reveal(I). Afterwards, a coin b is flipped. If $b = 0$, return a random number of the same length as SK from $\{0,1\}^l$, otherwise, return SK.

Fig. 4. Simulation of the random oracles, the Send, Execute, Corrupt, Reveal and Test-queries.

the previous game except that $\mathcal{A}$ queries from random oracle $\mathcal{H}$ on $(ID_i||CID_i||B_0||B_0^{new}||C_2||K)$, where $K = CMCDH(C_1, C_5) = T_{uv}(x)$. Therefore, we have

$$|Pr[E_4] - Pr[E_3]| \leqslant q_h Adv_{\mathcal{A}}^{CMCDH}(t') + \frac{q_h}{2^l}, \tag{4}$$

where $t' \leqslant t + (q_s + q_e + 1)T_c$. This experiment simulates the execution taking the advantage of random self reducibility of CMCDH problem. In order to simulate, $\mathcal{A}$ queries the random tuple $\langle T_\theta(x), T_\varphi(x)\rangle$ from corresponding $\mathcal{H}$ oracle to compute $CDH(T_\theta(x), T_\varphi(x)) = T_{\theta\varphi}(x)$, where $\theta, \varphi \in_R \mathbb{Z}_p^*$. This experiment simulates B_0 without querying the $\mathcal{H}$ oracle and getting s.

Game E_5. This experiment is similar to the prevalent experiment, but the only difference is the $Test$ query, this experiment is terminated if $\mathcal{A}$ queries a hash $\mathcal{H}$ query with $(ID_i||CID_i||B_0||B_0^{new}||C_2||T_{uv}(x))$. Therefore, $\mathcal{A}$ has the ability to obtain SK by asking $\mathcal{H}$ query with maximum probability of $\frac{q_h^2}{2^{l+1}}$. Moreover, if the $Corrup(U,2)$ query is asked, then $Corrupt(U,1)$ and $Corrupt(U,3)$ cannot be asked. In the case of allowing q_s times Send-queries for online guessing, the probability of getting password is at most $C' \cdot q_s^{s'}$ according to [39].

In addition, perfect forward security is also considered in this scenario. In the light of the definition of freshness, $\mathcal{A}$ can query $Corrupt(I^k)$ query after querying $Test(I^k)$ query. Thus, the obsolete transcripts are employed in the old experiments. The probability of getting $T_{uv}(x)$ is: $\frac{(q_s+q_e)^2}{2p}$ at most. Therefore, we have

$$|Pr[E_5] - Pr[E_4]| \leqslant C' \cdot q_s^{s'} + \frac{q_h^2}{2^{l+1}} + \frac{(q_s+q_e)^2}{2p}. \tag{5}$$

If $\mathcal{A}$ fails to ask any $\mathcal{H}$ query with the valid input, then there is no advantage for this experiment to distinguish the real SK with the session key of the same size made from the random value . Therefore, we have $Pr[E_5] = \frac{1}{2}$.

Finally, according to Game E_0-E_5 and Lemma 1, we have $Adv_{CM3FAKAP,\mathcal{D}}^{AKA}(\mathcal{A}) \leqslant 2C' \cdot q_s^{s'} + \frac{(q_s+q_h+q_h^2)}{2^{l-1}} + \frac{2(q_s+q_e)^2}{p}$ $+2q_h Adv_{\mathcal{A}}^{CMCDH}(t')$. □

Remark 3. The Canetti and Krawczyk (CK) model [60] allows partial/empheral information leakage. CK model is suitable for authentication and key exchange (AKE) protocols, but not suitable for analyzing password-based authentication protocols. In AKE protocols, it means that both sides of the communication have a digital certificate (i.e., a private key with *high entropy*), while password-based authentication schemes like ours are cryptographic protocols that use memorable *low entropy* password in order to eliminate the management problem of digital certificates. If password-based authentication protocols are analyzed with the CK model (i.e., temporary key information can be disclosed, partial information leakage), they cannot resist offline password guessing attack. Therefore, the BPR2000 model [57], Bresson *et al.*'s model [58], and the model of Wang *et al.* [39] are widely used to analyze the password-based authentication schemes [2], [3], [39] (and our proposed protocol).

4.2 Security Analysis Based on Evaluation Criteria

We will now evaluate the security of the proposed protocol, based on the evaluation criteria and adversary model presented in Section 2.

4.2.1 Timely Password Typo Detection

In the login phase of both protocols of Li *et al.* and Zhu *et al.*, SC_i has no capability to verify the login activity of any user until it is detected by S. However, the login phase of our protocol checks the password's validity by verifying whether $W'_i \stackrel{?}{=} W_i$ after inputting $\{ID_i, PW_i, BIO_i\}$. If these are found to be valid, then SC_i submits the request message to the server. Otherwise, it aborts this session. By doing so, our protocol reduces the communication and computation cost in the event of incorrect inputs or the attack triggered by an illegitimate user. Accordingly, the timely password typo detection is successfully handled by the proposed protocol.

4.2.2 User-Anonymity and Un-Traceability

In the proposed protocol, there is no identity information to be transmitted through the public channel or to be stored in the user's smart-card. Therefore, the adversary may only obtain user's identity during the communication session. Suppose that $\mathcal{A}$ extracts all parameters $\{CID_i, B_1, W_i, \tau_i, T_s(x), x, n_0, H(\cdot), H_0(\cdot)\}$ stored in SC_i, and obtains $\{CID_i, C_1, C_3, C_4\}, \{C_5, C_7, C_8\}$ from U_i and S by eavesdropping. CID_i can be calculated by a one-way function and is updated during every session. $C_3 = (ID_i||B_0) \oplus H_0(C_2)$ is variable because C_2 is random. $C_4 = H(ID_i||CID_i||B_0||C_2||C_3)$ is also variable and is secure due to the use of a one-way function. Therefore, $\mathcal{A}$ is unable to derive or to trace the user's identity ID_i. Thus, the proposed protocol ensures user anonymity and un-traceability.

4.2.3 Privileged Insider Attack

During the registration step of the proposed protocol, U_i only sends ID_i to S without any information relating to the password or biometrics. Then, S sends B_0 and a new smart-card SC_i to U_i. After receiving SC_i and B_0, U_i activates SC_i by providing PW_i and BIO_i that are only known to U_i. Finally, U has SC_i with the new parameters. Observing all parameters stored in SC_i, we find that PW_i is not available in plaintext but is secure due to the use of a one-way function. Moreover, it is also hard for $\mathcal{A}$ to retrieve the user's biometrics. Therefore, the presented protocol provides resistance to privileged insider attacks.

4.2.4 Key-Compromise User Impersonation Attack

In the proposed protocol, even if the long-term private key s of S is revealed to $\mathcal{A}$, $\mathcal{A}$ will not be able to impersonate the legitimate user U_i to S. This is because $\mathcal{A}$ is unable to obtain the value of B_0 to forge the login request message C_4. B_0 can be computed by two ways: (1) The legitimate user can calculate it because she knows $\{ID_i, PW_i, B_1, CID_i\}$ and can retrieve σ_i. (2) The server is able to compute it due to knowledge of $\{e_i, r_i\}$ and s. However, it is computationally challenging for $\mathcal{A}$ to obtain these key parameters. Thus, $\mathcal{A}$ has no capability to impersonate the user to S. Accordingly, the presented protocol is resilient to such attack.

4.2.5 Server Impersonation Attack

To impersonate the server$S$, $\mathcal{A}$ has to compute valid $\{C_5, C_7, C_8\}$. Since $C_7 = (CID_i^{new}||B_0^{new}) \oplus H(C_6||B_0)$ and $C_8 = H(ID_i||CID_i||B_0||B_0^{new}||C_2||SK_s)$ are computed by $\{B_0, ID_i, CID_i^{new}, B_0^{new}\}$ and protected by a one-way function, $\mathcal{A}$ must know these key parameters or to guess correct value within polynomial time. To obtain these key parameters, $\mathcal{A}$ needs to have $\{ID_i, s, r_i\}$. However, it is computationally infeasible for $\mathcal{A}$ to guess these private parameters within polynomial time, and the proposed protocol preserves user anonymity according to Section 4.2.2. Therefore, $\mathcal{A}$ is unable to compute valid response message, and hence the proposed protocol is resilient to server impersonation attacks.

4.2.6 Password-Guessing-Attack

In the presented CM3FAKAP, we suppose that $\mathcal{A}$ eavesdrops all messages over the public channel and is capable of stealing biometrics of the user and extracting all parameters from the user's smart card. Therefore, we analyze the password-guessing-attack from two aspects: (i). Off-line guessing attack: On the one hand, we assume that the user's identity ID_i is anonymous. Then, $\mathcal{A}$ retrieves σ_i and guesses $\{ID_i', PW_i'\}$ from dictionary guessing space. Afterwards, $\mathcal{A}$ will be able to compute the related parameter $VPW_i' = H(PW_i'||ID_i'||CID_i||\sigma_i)$, and compute $W_i' = H((H(ID_i') \oplus VPW_i') \bmod n_0)$. Subsequently, $\mathcal{A}$ checks whether $W_i' \stackrel{?}{=} W_i$. Finally, $\mathcal{A}$ repeats the above steps until she correctly guesses the identity and password. Clearly, $\mathcal{A}$ can guess the related ID_i' and PW_i', which satisfy $W_i' \stackrel{?}{=} W_i$ within a polynomial time. The reduced guessing size of identity and password space is presented as $\frac{|\mathcal{D}_{ID}|*|\mathcal{D}_{PW}|}{n_0}$, where $2^4 \leq n_0 \leq 2^8$, $\mathcal{D}_{ID}$ and $\mathcal{D}_{PW}$ refer to the guessing space of identity and password, respectively. However, $\mathcal{A}$ is not able to guess the correct identity and password for obtaining $B_0 = B_1 \oplus VPW_i \oplus \sigma_i$, because $\frac{|\mathcal{D}_{ID}|*|\mathcal{D}_{PW}|}{n_0} \approx 2^{32}$ is large enough when $|\mathcal{D}_{ID}| = |\mathcal{D}_{PW}| = 10^6$ and $n_0 = 2^8$. On the other hand, the user's identity ID_i may have been compromised by $\mathcal{A}$. However, $\mathcal{A}$ is still powerless to guess the correct password PW_i, because there exists $\frac{|\mathcal{D}_{PW}|}{n_0} \approx 2^{12}$ password candidates that satisfy the equation $W_i = H((H(ID_i) \oplus VPW_i') \bmod n_0)$. Therefore, no matter $\frac{|\mathcal{D}_{ID}|*|\mathcal{D}_{PW}|}{n_0} \approx 2^{32}$ or $\frac{|\mathcal{D}_{PW}|}{n_0} \approx 2^{12}$, the attacker can't find the real treasure through off-line guessing.

In order to obtain the correct password from the remaining password candidates, $\mathcal{A}$ can only launch the following on-line guessing attack. (ii). On-line guessing attack: To trigger this attack, $\mathcal{A}$ needs to guess the correct ID_i and PW_i pair from the guessing space of the reduced size $\frac{|\mathcal{D}_{ID}|*|\mathcal{D}_{PW}|}{n_0} \approx 2^{32}$ ($\frac{|\mathcal{D}_{PW}|}{n_0} \approx 2^{12}$ is practically large enough [39] when ID_i is not a secret). To pass the authentication of server S, $\mathcal{A}$ must have extracted the key parameter B_0. Moreover, $\mathcal{A}$ must be verified by the server within a limited communication duration, which is the fixed threshold value $N_0 = 5$. However, since $\frac{|\mathcal{D}_{ID}|*|\mathcal{D}_{PW}|}{n_0 * N_0} \approx 2^{30} \gg N_0$ and $\frac{|\mathcal{D}_{PW}|}{n_0 * N_0} \approx 2^{10} \gg N_0$, it is computationally hard for $\mathcal{A}$ to obtain the correct password and identity online with only N_0 chances.

Therefore, password-guessing-attack is impractical for the proposed protocol, within polynomial time.

4.2.7 De-Synchronization Attack

During the update phase, as long as the identity, password and biometric pass the verification of the smart card, the user is allowed to change the password or biometrics. Thus, the adversary remains incapable of performing the underlying attack. In the login and authentication phase, even if the attacker blocks the first message flow, the server is not required to update key parameters at the backend database. Thus, this action does not influence the consistency of next communications. Although the attacker blocks the second messages flow, it also cannot influence the consistency of communications between user and server, because only when $C_8 \stackrel{?}{=} C_8^*$ holds, the smart card will update the data of smart card. Thus, the proposed protocol can resist de-synchronization attacks.

4.2.8 Replay Attack

Assume that $\mathcal{A}$ has obtained the login request message $\{CID_i, C_1, C_3, C_4\}$ of the previous session through some public channel. If $\mathcal{A}$ replays $\{CID_i, C_1, C_3, C_4\}$ to S, then S verifies C_4. Since $\{CID_i, B_0\}$ are updated in every successive session, C_4 is also changed to C_4^{new} each time. Therefore, the older C_4 cannot be verified by the server during the current session. Otherwise, if $\mathcal{A}$ replays $\{C_5, C_7, C_8\}$ to U_i, then the older message C_8 also cannot be verified by U_i during the current session. Therefore, the proposed protocol can resist replay attacks.

4.2.9 Session-Specific Temporary Information Attack

In the proposed protocol, if all temporary information u, v, e_i are compromised, then $\mathcal{A}$ can compute C_2, C_6. To reveal $SK = H(ID_i||CID_i||B_0||B_0^{new}||C_2||C_6)$, $\mathcal{A}$ needs to calculate $\{B_0, B_0^{new}\}$ and to know the correct ID_i. Since $\mathcal{A}$ has no way of figuring out the correct $\{B_0, B_0^{new}\}$ computed by using the long-term private key s of the server. Therefore, the proposed protocol is resilient to session-specific temporary information attacks.

4.2.10 Man-in-the-Middle Attack

In the presented protocol, we suppose that $\mathcal{A}$ intercepts and blocks the login request message $\{CID_i, C_1, C_3, C_4\}$, the challenge message $\{C_5, C_7, C_8\}$, and extracts all parameters of SC_i. To be successful in the man-in-middle attack, $\mathcal{A}$ has to forge the new message flow $\{\{CID_i^*, C_1^*, C_3^*, C_4^*\}, \{C_5^*, C_7^*, C_8^*\}\}$ or to replay the previous message flow. As discussed earlier, the proposed protocol can resist impersonation attacks and replay attacks. Therefore, $\mathcal{A}$ is not capable of been authenticated by both user and server. Hence, the proposed protocol is resilient to man-in-middle attacks.

4.2.11 Mutual Authentication

In the proposed protocol, S authenticates U_i by verifying whether $C_4^* = C_4$, while U_i authenticates S by checking whether $C_8^* = C_8$. After mutual authentication, U_i and S negotiate a common session key SK. Thus, the proposed protocol achieves secure mutual authentication.

4.2.12 Perfect Forward Secrecy

This feature states that even if PW_i and BIO_i of U_i and all secret keys of S are revealed to $\mathcal{A}$, all prior session keys must remain secure. Let us suppose that all private keys $\{s, r_i, e_i\}$ of S and $\{PW_i, BIO_i\}$ of U_i are compromised, and $\mathcal{A}$ extracts $\{CID_i, B_1, W_i, \tau_i, T_s(x), x, n_0, H(\cdot), H_0(\cdot)\}$ stored in the smart card and eavesdrops and obtains $\{\{CID_i, C_1, C_3, C_4\}, \{C_5, C_7, C_8\}\}$. Then, $\mathcal{A}$ can figure out $C_2 = T_s(C_1)$ and $(ID_i||B_0) = C_3 \oplus H_0(C_2)$. However, to compute the previous session key $SK = H(ID_i||CID_i||B_0||B_0^{new}||C_2||C_6)$, $\mathcal{A}$ needs to know $C_6 = T_u(C_5) = T_v(C_1)$. However, it is computationally infeasible for $\mathcal{A}$ to extract random number u from C_1 or v from C_5 and to calculate C_6 due to the intractability of $CMDLP$ and $CMCDHP$. Thus, even if all secret parameters are compromised, $\mathcal{A}$ is not capable of computing SK. Thereupon, the proposed protocol achieves perfect forward secrecy.

4.2.13 Efficient Update Phase

In both protocols of Zhu *et al.* [22] and Li *et al.* [24], to update the user's password or biometric, the user is required to communicate with the server. This has implications on the communication and computation cost. However, in the proposed protocol, U_i can update her password or biometrics only by interacting with SC_i. Moreover, S does not need to participate in update phase. Hence, this reduces the associated communication and computation cost.

4.2.14 Three-Factor Security

It is known that any three-factor AKA protocol $\mathcal{P}$ incorporates three features including password, smart-card and biometric. This implies that $\mathcal{P}$ is different from the single-factor and two-factor AKA protocol. We now let $\mathcal{A}$ obtain any two of the three factors. According to the previous analysis, $B_0 = B_1 \oplus VPW_i \oplus \sigma_i$ is the key parameter to launch any attack and compute the session key. Thus, in this section, we will explain why $\mathcal{A}$ cannot compute B_0 in any of these combinations of knowledge: password and smart-card, password and biometric, biometric and smart-card.

- *Case I:* Although $\mathcal{A}$ knows the password and can extract all parameters in the smart card, she still cannot compute $B_0 = B_1 \oplus VPW_i \oplus \sigma_i$ to forge the valid login message without having access to biometric information σ_i.
- *Case II:* Even if $\mathcal{A}$ obtains password and biometric information without the smart card, she cannot compute B_0 without B_1, because $B_0 = B_1 \oplus VPW_i \oplus \sigma_i$ and B_1 are stored in the smart-card.
- *Case III:* Although $\mathcal{A}$ has both biometric information and the smart card, she cannot compute VPW_i without knowing the correct PW_i. So $\mathcal{A}$ is unable to compute $B_0 = B_1 \oplus VPW_i \oplus \sigma_i$.

Therefore, the proposed protocol achieves three-factor security, which is very vital for the three-factor authentication protocols.

5 Comparative Summary: Security and Performance

To demonstrate the good balance of the proposed AKA protocol in security and usability, this section provides a comparative measurement of the security, computation and communication cost of Islam *et al.* [4], Wazid *et al.* [2], Tsai *et al.* [29], Roy *et al.* [3], Liu *et al.* [21], Zhu *et al.* [22], Jiang *et al.* [23]'s AKA protocols and the proposed protocol.

5.1 Security Comparison

Now we use the evaluation metric introduced in Section 2.4 to perform an objective comparison of our new protocol with seven state-of-the-art protocols. The comparison results of [2], [3], [4], [21], [22], [23], [29] and the presented protocol are depicted in Table 3.

In schemes in [2], [3], [4] and [22], smart cards store an explicit password validation parameter, these protocols cannot resist off-line password guessing attacks. While there is no password validation parameters in [21] and [29], this will lead to denial of service attack. The scheme in [3] uses only three chaotic operations, unfortunately, it still fails to provide perfect forward security when long-term private key leaks. Despite the use of elliptic curve cryptography in [29], this protocol stores a key in storage device, which directly results in the failure of forward

TABLE 3
Security Comparison Among Relevant Three-Factor AKA Protocols

Protocols	Evaluation Criteria													Assumption
	EC_1	EC_2	EC_3	EC_4	EC_5	EC_6	EC_7	EC_8	EC_9	EC_{10}	EC_{11}	EC_{12}	EC_{13}	
Roy *et al*. (2018)[3]	✓	✓	✓	✓	✓	×	×	✓	✓	×	✓	×	×	CMCDHP
Islam *et al*. (2018)[4]	✓	✓	✓	✓	✓	×	×	✓	✓	✓	×	×	✓	ECCDHP
Wazid *et al*. (2017)[2]	✓	✓	✓	✓	✓	×	×	✓	✓	×	×	×	✓	ECCDHP
Liu *et al*. (2016)[21]	×	✓	×	✓	×	–	✓	✓	✓	×	–	–	✓	CMCDHP
Jiang *et al*. (2016)[23]	✓	✓	✓	✓	✓	✓	×	✓	✓	×	✓	✓	✓	CMCDHP
Zhu *et al*. (2015)[22]	×	✓	✓	✓	✓	×	×	✓	✓	✓	✓	×	✓	CMCDHP
Tsai *et al*. (2013)[29]	✓	✓	✓	✓	×	✓	×	✓	✓	✓	×	×	×	ECCDHP
Our protocol	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	CMCDHP

✓ means the property is satisfied; × means the property is not satisfied; – means the property is not considered; ECCDHP means Elliptic Curve Computational Diffie-Hellman Problem; CMCDHP means Chaotic-Maps Computational Diffie-Hellman Problem.

secrecy protection. In [21], it can not provide anonymity due to the transmission of plaintext identity on public channels, and it also can not resist clock synchronization attacks. Since these AKA protocols [2], [3], [4], [21] and [22] are unable to resist off-line password guessing attack, they cannot provide three-factor secrecy directly. The scheme in [29] can resist off-line password guessing attack, but it can not resist key-compromised impersonation attack and can not provide forward secrecy. Jiang *et al*.'s protocol [23] is prone to key compromise user impersonation attack and clock synchronization attack.

In Table 3, we summarize the evaluation statistics of security indicators about each protocol. Islam *et al*.'s protocol [4] is vulnerable to off-line password guessing attacks and key-compromise user impersonation attacks, and cannot provide smart-card revocation. So, based on security evaluation criteria of Table 2, Islam *et al*.'s protocol [4] does not fulfill EC_6, EC_7, EC_{11} and EC_{12}. Wazid *et al*.'s protocol [2] can neither resist off-line password guessing attack, sessionspecific temporary information attack and clock Synchronization attack, nor provide three-factor security and smartcard revocation function. That is, [2] does not achieve $EC_6, EC_7, EC_{10}, EC_{11}$ and EC_{12}. Similarly, Tsai *et al*.'s protocol [29] does not provide $EC_5, EC_7, EC_{11}, EC_{12}$ and EC_{13}, and Roy *et al*.'s protocol [3] does not satisfy $EC_6, EC_7, EC_{10}, EC_{12}$ and EC_{13}. In a nutshell, by observing Table 3, we can deduce that the presented protocol is the only one that is immune to various known attacks and achieves the desirable security and usability goals.

5.2 Computation and Communication Cost Comparison

To analyze the computational complexity of login and authentication step, the lightweight operations including exclusive-OR and string concatenation operation are omitted. For better presentation, the following additional notations are used, i.e., T_c is the computational time for the extended chaotic-maps, T_m is the computational time for executing an elliptic curve point multiplication, T_a is the computational time for executing an elliptic curve point addition, T_s is the computational time for symmetric cryptography operation, T_h is the computational time for operating a one-way hash operation.

The cost of server-end and user-end required for the protocols in login and authentication phase is presented in Table 4. The total computational cost required in the proposed protocol during login and authentication phase is $18T_h + 6T_c$. Also, we can observe that the proposed protocol requires slightly more computational cost, as compared to the other protocols. This is because of the additional functionalities and better security (i.e. trade-off between security and usability). To measure the communication cost in the login and authentication phase, we use the length of the security parameter as follows: the length of each component of the function $GEN(\cdot)$ output is 80 bits, the length of the user identity is 160 bits, the bit length of the prime number p is 256 bits, the bit length of the random number is 128 bits, the length of an elliptic curve point is 160 bits, the output length of the hash function (SHA-160) is 160 bits, the length

TABLE 4
Summary of Computational Cost in the Login and Authentication Phase

Protocols	Computational cost			CC	CR
	User(smart-card)	Server	Total		
Roy *et al*. (2018)[3]	$9T_h + 2T_c$	$6T_h + T_c$	$15T_h + 3T_c$	960 bits	2
Islam *et al*. (2018)[4]	$7T_h + 2T_m + T_s$	$5T_h + 2T_m + T_s$	$12T_h + 4T_m + 2T_s$	768 bits	3
Wazid *et al*. (2017)[2]	$8T_h + 2T_m$	$5T_h + 4T_m + 2T_a$	$13T_h + 6T_m + 2T_a$	1140 bits	3
Liu *et al*. (2016)[21]	$6T_h + 3T_c$	$6T_h + 3T_c$	$12T_h + 6T_c$	1280 bits	3
Jiang *et al*. (2016)[23]	$6T_h + 3T_c$	$5T_h + 3T_c$	$11T_h + 6T_c$	768 bits	2
Zhu *et al*. (2015)[22]	$4T_h + 2T_c$	$6T_h + 2T_c$	$10T_h + 4T_c$	736 bits	2
Tsai *et al*. (2013)[29]	$5T_h + T_m$	$5T_h + 3T_m$	$10T_h + 4T_m$	960 bits	3
Our protocol	$10T_h + 3T_c$	$8T_h + 3T_c$	$18T_h + 6T_c$	1376 bits	2

CC means Communication Cost; CR means Number of Communication Message Flows. Some lightweight operations like exclusive-OR and "||" are ignored.

of the ciphertext of the symmetric encryption/decryption algorithm is 128 bits and the bit length of the timestamp is 16 bits. The total communication cost of our protocol is (160 +128+320+160)+(128+160×2+160)=1376 bits. From Table 4, we can observe that existing protocols cost slightly less communication than our protocol. But our protocol only needs two communication message flows, while [2], [4], [21] and [29] need three flows.

However, in summary, the proposed AKA protocol is best suitable for three-factor authentication and key agreement with regard of the trade-off between security and usability for mobile lightweight devices.

6 Conclusion

In this paper, we have designed a new chaotic-maps-based AKA protocol by adopting the techniques of "Fuzzy-Verifiers" and "Honeywords", and then prove its security from formal and evaluation criteria-based security analysis. The security analysis results indicate that the proposed AKA protocol is capable to achieve semantic security and meet the 13 evaluation criteria (EC_1-EC_{13}). By comparing the security, computation and communication costs of our protocol with the state-of-the-art protocols, we show that our new protocol is more practical for mobile lightweight devices. Its design ideas are generic and can be used as a guideline to build AKA protocols with a good balance of security and usability. Our future work is to develop concrete three-factor AKA protocols with robust security and high efficiency for scenarios where more than three or four parties are involved (e.g., cloud computing and edge computing environments).

Acknowledgments

The authors thank the anonymous reviewers for their invaluable comments. This work was supported by the National Natural Science Foundation of China (No. 61802006, 61897069), National Key Research and Development Program of China (No. 2018YFB0803605), Science and Technology Research Project of Education Department of Jiangxi Province (No. GJJ191680), and Doctoral Foundation of Jiangxi Normal University.

References

[1] The Statistics Portal, "Forecast number of mobile users worldwide from 2019 to 2023," Feb. 28, 2020, Accessed: Mar. 2020. [Online]. https://www.statista.com/statistics/218984/number-of-global-mobile-users-since-2010

[2] M. Wazid, A. K. Das, N. Kumar, and J. Rodrigues, "Secure three-factor user authentication scheme for renewable-energy-based smart grid environment," *IEEE Trans. Ind. Inform.*, vol. 13, no. 6, pp. 3144–3153, Dec. 2017.

[3] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, S. Kumari, and M. H. Jo, "Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2884–2895, Aug. 2018.

[4] S. H. Islam, P. Vijayakumar, M. Z. A. Bhuiyan, R. Amin, V. R. M., and B. Balusamy, "A provably secure three-factor session initiation protocol for multimedia big data Communications," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3408–3418, Oct. 2018.

[5] C. G. Ma, D. Wang, and S. D. Zhao, "Security flaws in two improved remote user authentication schemes using smart cards," *Int. J. Commun. Syst.*, vol. 27, no. 10, pp. 2215–2227, 2014.

[6] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[7] J. Arkko, V. Torvinen, G. Camarillo, A. Niemi, and T. Haukka, "Security mechanism agreement for SIP sessions," *IETF RFC 3329*, Jan. 2003. [Online]. Available: https://datatracker.ietf.org/doc/rfc3329/

[8] A. Durlanik and I. Sogukpinar, "SIP authentication scheme using ECDH," *World Enformatika Soc. Trans. Eng. Comput. Technol.*, vol. 8, pp. 350–353, 2005.

[9] D. Wang, H. B. Cheng, D. B. He, and P Wang, "On the challenges in designing identity-based privacy-preserving authentication schemes for mobile devices," *IEEE Syst. J.*, vol. 12, no. 1, pp. 916–925, Mar. 2018.

[10] S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment," *IEEE Trans. Dependable Secur. Comput.*, vol. 15, no. 5, pp. 824–839, Sep./Oct. 2018.

[11] P. Gope, J. Lee, and T. Q. S. Quek, "Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 11, pp. 2831–2843, Nov. 2018.

[12] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for Wireless sensor networks: Attacks, principle and solutions," *Comput. Netw.*, vol. 73, pp. 41–57, 2014.

[13] Z. Yang, J. He, Y. G. Tian, and J. Y. Zhou, "Faster authenticated key agreement with perfect forward secrecy for industrial internet-of-things," *IEEE Trans. Ind. Inform.*, vol. 16, no. 10, pp. 6584–6596, Oct. 2020.

[14] M. L. Das, A. Saxena, and V. P. Gulati, "A dynamic ID-based remote user authentication scheme," *IEEE Trans. Consum. Electron.*, vol. 50, no. 2, pp. 629–631, May 2004.

[15] L. Kocarev, and S. Lian, *Chaos-Based Cryptography: Theory, Algorithms and Applications*. Berlin, Germany: Springer, 2011.

[16] D. Xiao, X. F. Liao, and K. W. Wong, "An efficient entire chaos-based scheme for deniable authentication Chaos," *Solitons Fractals.*, vol. 23, pp. 1327–1331, 2005.

[17] G. Alvarez, "Security problems with a chaos-based deniable authentication scheme Chaos," *Solitons Fractals.*, vol. 26, pp. 7–11, 2005.

[18] D. Xiao, X. F. Liao, and S. J. Deng, "A novel key agreement protocol based on chaotic maps," *Inf. Sci.*, vol. 177, no. 4, pp. 1136–1142, 2007.

[19] L. D. Han, Q. Xie, W. H. Liu, and S. B. Wang, "A new efficient chaotic maps based three factor user authentication-and-key-agreement scheme," *Wireless Pers, Commun.*, vol. 95, no. 3, pp. 3391–3406, 2017.

[20] T. F. Lee, C. H. Hsiao, S. H. Hwang, and T. H. Lin, "Enhanced smart card-based password authenticated key agreement using extended chaotic maps," *PLoS ONE*, vol. 12, no. 7, 2017. Art. no. e0181744. [Online]. Available: https://doi.org/10.1371/journal.pone.0181744

[21] Y. Liu and K. Xue, "An improved secure and efficient password and chaos-based two-party key agreement protocol," *Nonlinear Dyn.*, vol. 84, no. 2, pp. 549–557, 2016.

[22] H. Zhu and X. Hao, "A provable authenticated key agreement protocol with privacy protection using smart-card based on chaotic maps," *Nonlinear Dyn.*, vol. 81, no. 1–2, pp. 311–321, 2015.

[23] Q. Jiang *et al.*, "Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy," *Nonlinear Dyn.*, vol. 83, no. 4, pp. 2085–2101, 2016.

[24] X. Li, F. Wu, M. K. Khan, L. L. Xu, J. Shen, and M. H. Jo, "A secure chaotic map-based remote authentication scheme for telecare-medicine-information-systems," *Future Gener. Comput. Syst.*, vol. 84. pp. 149–159, 2018.

[25] D. Wang, N. Wang, P. Wang, and S. H. Qing, "Preserving privacy for free: Efficient and provably secure two-factor authentication scheme with user anonymity," *Inf. Sci.*, vol. 321, pp. 162–178, 2015.

[26] C. Guo and C. C. Chang, "Chaotic maps-based password-authenticated key agreement using smart-cards," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, pp. 1433–1440, 2013.

[27] G. M. Yang, D. S. Wong, H. X. Wang, and X. T. Deng, "Two-factor mutual authentication based on smart-cards and passwords," *J. Comput. Syst. Sci.*, vol. 74, no. 7, pp. 1160–1172, 2008.

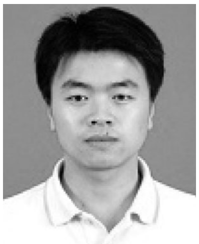
[28] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1086–1090, Mar. 2009.

[29] J. L. Tsai, N. W. Lo, and T. C. Wu, "Novel anonymous authentication scheme using smart cards," *IEEE Trans. Ind. Inform.*, vol. 9, no. 4, pp. 2004–2013, Nov. 2013.

[30] H. Y. Lin, "Improved chaotic maps-based password-authenticated key agreement using smart-cards," *Commun. Nonlinear Sci. Numerical Simul.*, vol. 20, pp. 482–488, 2015.

[31] D. Wang and P. Wang, "On the implications of Zipf's law in passwords," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2016, pp. 111–131.

[32] S. H. Islam, "Provably secure dynamic identity-based three-factor password authentication scheme using extended chaotic maps," *Nonlinear Dyn.*, vol. 78, no. 3, pp. 2261–2276, 2014.
[33] T. F. Lee, "Enhancing the security of password authenticated key agreement protocols based on chaotic maps," *Inf. Sci.*, vol. 290, pp. 63–71, 2015.
[34] X. Guo and J. Zhang, "Secure group key agreement protocol based on chaotic hash," *Inf. Sci.*, vol. 180. no. 20, pp. 4069–4074, 2010.
[35] C. M. Chen, W. C. Fang, K. H. Wang, and T. Y. Wu, "Comments on an improved secure and efficient password and chaos-based two-party key agreement protocol," *Nonlinear Dyn.*, vol. 87, no. 3, pp. 2073–2075, 2017.
[36] S. Shin and K. Kobara, "Security analysis of password-authenticated key retrieval," *IEEE Trans. Dependable Secur. Comput.*, vol. 14, no. 5, pp. 573–576, Sep./Oct. 2017.
[37] *IEEE Standard Specifications for Password-Based Public-Key Cryptographic Techniques*, IEEE Standard 1363.2$^{\text{TM}}$-2008, Jan. 2009.
[38] D. P. Jablon, "Password authentication using multiple servers," in *Proc. Conf. Topics Cryptol., The Cryptographer's Track at RSA*, 2001, pp. 344–360.
[39] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE Trans. Dependable Secur. Comput.*, vol. 15, no. 4, pp. 708–722, Jul./Aug. 2018.
[40] L. Kocarev and Z. Tasev, "Public-key encryption based on chebyshev maps," in *Proc. IEEE Symp. Circuits Syst.*, 2003, pp. 28–31.
[41] L. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Chaos Solitons Fract.*, vol. 37, no. 3, pp. 669–674, 2008.
[42] P. Bergamo, P. Arco, A. Santis, and L. Kocarev, "Security of public key cryptosystems based on Chebyshev polynomials," *IEEE Trans. Circuits Syst.*, vol. 52, pp. 1382–1393, Jul. 2005.
[43] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, 2004, pp. 523–540.
[44] D. Wang, D. He, P. Wang, and C. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Trans. Dependable Secur. Comput.*, vol. 12, no. 4, pp. 428–442, Jul./Aug. 2015.
[45] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, and M. T. Shalmani, "On the power of power analysis in the real world: A complete break of the KeeLoq code hopping scheme," in *Proc. Annu. Int. Cryptology Conf.*, 2008, pp. 203–220.
[46] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Annu. Int. Cryptol. Conf.*, 1999, pp. 388–397.
[47] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51. no. 5, pp. 541–552, May 2002.
[48] D. Wang D, Z. Zhang, and P. Wang, "Targeted online password guessing: An underestimated threat," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1242–1254.
[49] D. Wang, H. B. Cheng, P. Wang, X. Y. Huang, and G. P. Jian, "Zipf's law in passwords," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 11, pp. 2776–2791, Nov. 2017.
[50] S. Agrawal, M. L. Das, and J. López,"Detection of node capture attack in wireless sensor networks," *IEEE Syst. J.*, vol. 13, no. 1, pp. 238–247, Mar. 2019.
[51] D. B. He and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment," *IEEE Syst. J.*, vol. 9, no. 3, pp. 816–823, Sep. 2015.
[52] H. Krawczyk, "HMQV: A high-performance secure Diffie-Hellman protocol," in *Proc. Annu. Int. Cryptology Conf.*, 2005, pp. 546–566.
[53] D. Wang, W. T. Li, and P. Wang, "Measuring two-factor authentication schemes for real-rime data access in industrial Wireless sensor networks," *IEEE Trans. Ind. Inform.*, vol. 14, no. 9, pp. 4081–4092, Sep. 2018.
[54] R. Madhusudhan and R. Mittal, "Dynamic ID-based remote user password authentication schemes using smart-cards: A review," *J. Netw. Comput. Appl.*, vol. 35, no. 4, pp. 1235–1248, 2012.
[55] A. Juels and R. L. Rivest, "Honeywords: Making password-cracking detectable," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur*, 2013, pp. 145–160.
[56] D. Wang, H. B. Cheng, P. Wang, J. Yan, and X. Y. Huang, "A security analysis of honeywords," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018. [Online]. Available: http://dx.doi.org/10.14722/ndss.2018.23142
[57] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, 2000, pp. 139–155.
[58] E. Bresson, O. Chevassut, and D. Pointcheval, "Security proofs for an efficient password-based key exchange," in *Proc. ACM 10th ACM Conf. Comput. Commun. Secur.*, 2003, pp. 241–250.
[59] V. Shoup, "Sequences of games: A tool for taming complexity in security proofs," *IACR Cryptol. ePrint Archive.*, vol. 332, pp. 1–33, 2004. [Online]. Available: https://eprint.iacr.org/2004/332.pdf
[60] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Proc. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2001, pp. 453–474.



Shuming Qiu received the PhD degree from the Beijing University of Posts and Telecommunications, China, in 2019. He won the 2018 China Internet Development Foundation Network Security Scholarship. He is currently a lecturer with Jiangxi Normal University, China. His research interests include authentication protocol, lattice cryptography, fully homomorphic encryption, non-communicative cryptography, (semi-)group algebra, etc.

Ding Wang received the PhD degree in information security from Peking University, in 2017. He is currently a professor with the College of Cyber Science, Nankai University, and also serves as the deputy director of the Tianjin key laboratory of Network and Data Security Technology. As the first author, he has published more than 40 papers at venues like ACM CCS, Usenix Security, NDSS, IEEE DSN, ESORICS, the *ACM Transactions on Cyber-Physical Systems*, *IEEE Transactions on Dependable and Secure Computing*, and *IEEE Transactions on Information Forensics and Security*. Seven of them are recognized as "ESI highly cited papers". His PhD thesis receives the "ACM China Doctoral Dissertation Award" and "China Computer Federation (CCF) Outstanding Doctoral Dissertation Award". He has been involved in the community as a TPC member, AEC member, or PC Chair for more than 50 international conferences such as Usenix Security, ACSAC, ACM AsiaCCS, SEC, ISC, and ACISP. His research interests include password, multi-factor authentication, and provable security.

Guoai Xu received the PhD degree in signal and information processing from the Beijing University of Posts and Telecommunications, China, in 2002. He was awarded the Title of Professor in 2011. He is currently an associate director with the National Engineering Laboratory of Security Technology for Mobile Internet, School of Cyberspace Security, Beijing University of Posts and Telecommunications. His research interests inclide the areas of software security and data analysis.

Saru Kumari received the PhD degree in mathematics from Chaudhary Charan Singh University, Meerut, Uttar Pradesh, India, in 2012. She is currently an assistant professor with the Department of Mathematics, Chaudhary Charan Singh University, Meerut, Uttar Pradesh, India. She has published 190 papers in international journals and conferences including 170 research publications in SCI indexed journals. Her current research interests include information security, digital authentication, security of wireless sensor networks, and applied mathematics.

Practical and Provably Secure Three-Factor Authentication Protocol Based on Extended Chaotic-Maps for Mobile Lightweight Devices (Appendix File)

Shuming Qiu, Ding Wang, Guoai Xu, and Saru Kumari

Abstract—This appendix file consists of one part. In the Appendix A, we simulate the proposed protocol using AVISPA. The simulation results show that the proposed protocol is secure against the active and passive attacks under the Dolev-Yao model.

Index Terms—Appendix file.

Appendix A: Security Verification of The Proposed Protocol Using AVISPA

In this Appendix, we simulate the proposed protocol using AVISPA. AVISPA (Automated Validation of Internet Security Protocols and Applications) is a button software tool and widely accepted for automatically validating Internet security sensitive protocols and applications [1]. AVISPA supports high-level protocol specification language called HLPSL, which is usually used to provide formal security verification for the simulated protocol. The simulation results of AVISPA can verify whether the simulated protocol is secure against active and passive attacks. Now, we first describe the proposed protocol using HLPSL, and then execute the HLPSL specifications using SPAN. Lastly, the simulation results are presented.

HLPSL Specification of the Proposed Protocol

In this section, we translate the protocol into HLPSL language, and provide the standardized descriptions of user, server, session, environment and goal roles under HLPSL language (See Section , Section and Section for details). In the simulation, the following eight security goals and two authentication properties were verified.

Goal 1: The `secrecy_of subs1` means that ID_i is kept secret to (U_i, S).

Goal 2: The `secrecy_of subs2` means that PW_i, u and σ_i are kept secret to U_i only.

Goal 3: The `secrecy_of subs3` means that B_0 is kept secret to (U_i, S).

Goal 4: The `secrecy_of subs4` means that random key parameters $\{C_2, C_6\}$ are kept secret to (U_i, S).

Goal 5: The `secrecy_of subs5` means that the negotiated session key SK_u is known by (U_i, S).

Goal 6: The `secrecy_of subs6` means that the secret key s and the secret parameter r_i are permanently kept secret, known to only S.

Goal 7: The `secrecy_of subs7` means that the the random number v is only known to S.

Goal 8: The `secrecy_of subs8` means that the negotiated session key SK_s is known by (U_i, S).

Authentication Property 1: The `authentication_on alice_server_c2` means that U_i generates u and compute C_2. If S can computes C_2 by making use of the login request message C_1 and successfully verify C_4, then it authenticates U_i.

Authentication Property 2: The `authentication_on server_alice_c6` means that S generates v. If U_i can compute C_6 by making use of the respond message C_5 and successfully verify C_8, then it authenticates S.

Role specification of U_i in HLPSL

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role alice(Ui,S:agent,SKas:symmetric_key,
SKsa:symmetric_key,H,Chao,GEN,REP:hash_func,
Snd,Rcv:channel(dy))
played_by Ui
def= local State:nat,IDi,PWi,VPWi,BIOi,SIGi,TAUi,
N0,CIDi,B0,B1,Ts,X,Wi,UUi,C2,C6,CIDiN,B0N:text,
C1,C3,C4,C5,C7,C8,SKu:message,
Inc:hash_func
const alice_server,server_alice,subs1,subs2,
subs3,subs4,subs5,subs6,subs7,subs8:protocol_id
init State:=0
transition
1.State=0/\Rcv(start)=|>
State':=1
/\IDi':=new()/\Snd({IDi'}_SKas)
/\secret({IDi},subs1,{Ui,S})
2.State=1/\Rcv({Ts'.X'.CIDi'.B0'}_SKsa)=|>
```


- *Shuming Qiu is with the Elementary Educational College, Jiangxi Normal University, Nanchang 330022, China; and also with National Engineering Laboratory of Mobile Network Security, Beijing University of Posts and Telecommunications, Beijing 100876, China*
  E-mail: qiushuming2008@163.com, shumingqiu@bupt.edu.cn.
- *Ding Wang is with College of Cyber Science, Nankai University, Tianjin 300350, China; and also with Tianjin Key Laboratory of Network and Data Security Technology, Nankai University, Tianjin 300350, China*
  E-mail: wangding@nankai.edu.cn, wangding@pku.edu.cn.
  Corresponding author.
- *Guoai Xu is with the National Engineering Laboratory of Mobile Network Security and School of cyberspace security, Beijing University of Posts and Telecommunications, Beijing 100876, China*
  E-mail: xga@bupt.edu.cn.
- *Saru Kumari is with the Department of Mathematics, Ch. Charan Singh University, Meerut 250 005, Uttar Pradesh, India*
  E-mail: saryusiirohi@gmail.com, saru@ccsuniversity.ac.in.

```
State':=2
/\PWi':=new()/\SIGi':=GEN(BIOi)/\TAUi':=GEN(BIOi)
/\VPWi':=H(PWi'.IDi.CIDi'.SIGi')
/\Wi':=H(xor(H(IDi),VPWi').N0)
/\B1':=xor(B0',VPWi',SIGi')/\UUi':=new()
/\C1':=Chao(UUi'.X')/\C2':=Chao(C1'.Ts)
/\C3':=xor(IDi,C2')/\C4':=H(IDi.CIDi'.B0'.C2'.C3')
/\Snd(CIDi'.C1'.C3'.C4')
/\secret({PWi',UUi',SIGi'},subs2,Ui)
/\secret({B0'},subs3,{Ui,S})
/\witness(Ui,S,alice_server_c2,C2')
3.State=2/\Rcv(C5'.C7'.C8')=|>
State':=3
/\C6':=Chao(C1.C5')/\B0N':=xor(C7',H(C6'.B0))
/\SKu':=H(IDi.CIDi.B0.B0N'.C2.C6')
/\secret({C2,C6},subs4,{Ui,S})
/\secret({SKu},subs5,{Ui,S})
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Role specification of S in HLPSL

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role server(S,Ui:agent,SKas:symmetric_key,
SKsa:symmetric_key,H,Chao,GEN,REP:hash_func,Snd,
Rcv:channel(dy))
played_by S
def= local State:nat,
SS,Ri,CIDi,Ei,IDi,VS,B0,Ts,X,C2,C6,EiN,CIDiN,
B0N:text,C1,C3,C4,C5,C7,C8,SKs:message,
Inc:hash_func
const alice_server,server_alice,subs1,subs2,
subs3,subs4,subs5,subs6,subs7,subs8:protocol_id
init State:=0
transition
1.State=0/\Rcv({IDi'}_SKas)=|>
State':=1
/\Ei':=new()/\Ri':=new()/\CIDi':=H(IDi'.Ei')
/\B0':=H(IDi'.SS.Ri'.CIDi')/\Ts':=new()/\X':=new()
/\Snd({Ts'.X'.CIDi'.B0'}_SKsa)
/\secret({SS,Ri'},subs6,S)
2.State=1/\Rcv(CIDi'.C1'.C3'.C4')=|>
State':=2
/\C2':=Chao(Ts.C1')/\IDi':=xor(C3',C2')
/\B0':=H(IDi'.SS.Ri.CIDi')/\EiN':=new()
/\CIDiN':=H(IDi'.EiN')/\B0N':=H(IDi'.SS.Ri.CIDiN')
/\VS':=new()/\C5':=Chao(VS'.X)/\C6':=Chao(C5'.C1')
/\C7':=xor(B0N',H(C6'.B0'))
/\SKs':=H(IDi'.CIDi'.B0'.B0N'.C2'.C6')
/\C8':=H(IDi'.CIDi'.B0'.B0N'.C2'.SKs')
/\witness(S,Ui,server_alice_c6,C6')
/\secret({VS'},subs7,S)/\secret({SKs'},subs8,{Ui,S})
/\Snd(C5'.C7'.C8')
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Specification of the session, environment, and goal in HLPSL

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role session(S,Ui:agent,SKas:symmetric_key,
SKsa:symmetric_key,H,Chao,GEN,REP:hash_func)
def=local SI,SJ,RI,RJ:channel(dy)
composition
alice(Ui,S,SKas,SKsa,H,Chao,GEN,REP,SI,RI)
/\server(Ui,S,SKas,SKsa,H,Chao,GEN,REP,SJ,RJ)
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role environment()
def=const ui,s:agent,
skas:symmetric_key,sksa:symmetric_key,h,chao,gen,
rep:hash_func,idi,pwi,bioi,wi,taui,n0,b0,b1,ts,x,
wi,c1,c2,c3,c4,c5,c6,c7,c8,cidi,cidin,ein,b0n,ska,
sks:text,alice_server_c2,server_alice_c6,subs1,subs2,
subs3,subs4,subs5,subs6,subs7,subs8:protocol_id
intruder_knowledge={ui,s,h,chao,gen,rep,cidi,b1,wi,
taui,ts,x,n0,c1,c3,c4,c5,c7,c8}
composition
session(ui,s,skas,sksa,h,chao,gen,rep)
/\session(s,ui,skas,sksa,h,chao,gen,rep)
end role
goal
secrecy_of subs1 secrecy_of subs2 secrecy_of subs3
secrecy_of subs4 secrecy_of subs5 secrecy_of subs6
secrecy_of subs7 secrecy_of subs8
authentication_on alice_server_c2
authentication_on server_alice_c6
end goal
environment()
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results
  /TDSC_three_factor_smq.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.63s
  visitedNodes: 18 nodes
  depth: 4 plies
```

Fig. 1. Simulation result of the proposed protocol under OFMC model

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

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

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 2 states
  Reachable  : 0 states
  Translation: 0.16 seconds
  Computation: 0.00 seconds
```

Fig. 2. Simulation result of the proposed protocol under CL-AtSe mode

Simulation Results

In order to test the security of the proposed protocol by using of SPAN (Security Protocol ANimator for AVISPA) [2], our experimental environment is Inter(R) Core(TM) i5-5200U CPU@2.20GHz, RAM8.00+4.00GB, SPAN-Ubuntu 10.10-light, Ubuntu (32-bit), virtual machine memory 2GB. Since the OFMC (On-the-Fly-Model-Checker) and CL-AtSe (Constraint-Logic-based Attack Searcher) models accept Diffie Hellman and XOR operations, we use them to simulate the proposed protocol. After the implementation of OFMC and CL-ATSE, the simulation results of the proposed protocol are shown in Figs. 1 and 2. It can be observe that in the OFMC model, the search depth is 4, 18 nodes are accessed, the analysis time is almost 0 second, the search time is 0.63 second, and the simulation result shows SAFE; in the CL-ATSE model, two states are analyzed, where the translation takes 0.16 second and the computation time is almost 0 second, and the simulation result shows SAFE. Therefore, the simulation results show that the proposed protocol is secure against the active and passive attacks under the Dolev-Yao model [3].

REFERENCES

[1] AVISPA, "Automated validation of internet security protocols and applications," Accessed: March. 2020, Available online: http://www.avispaproject.org.

[2] SPAN, "A security protocol animator for avispa," Accessed: March. 2020, Available online: http://www.avispaproject.org.

[3] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory.*, vol. 29, no. 2, pp. 198–208, 2003.