Robust Three-Factor Lightweight Authentication Based on Extended Chaotic Maps for Portable Resource-Constrained Devices

Arijit Karati[a], Yu-Sheng Chang and Ting-Yu Chen
Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, Republic of China

Keywords: Authentication, Key Agreement, Extended Chaotic Maps (ECM), Physically Unclonable Function (PUF), Security and Privacy, Lightweight Cryptography.

Abstract: Public-key based authentication and key agreement (AKA) protocols have attracted considerable interest in providing secure access for various application scenarios. Although three-factor AKA (3FAKA) offers higher security than one- or two-factor ones, most existing 3FAKA are vulnerable, or their safety is reduced to the security of one- or two-factor authentication. Thus, finding a balance between security and usability and countering cloning risks with robust three-factor authentication is an ongoing problem. To mitigates such issues, we propose a lightweight 3FAKA for mobile devices. The suggested 3FAKA employs the physical unclonable function to withstand device cloning attacks and extended chaotic maps to preserve lightweight processes while ensuring essential cryptographic traits, such as unpredictability, unrepeatability, and uncertainty. It is secure under the intractability of extended chaotic maps computational Diffie-Hellman problem. Performance analysis exhibits that our protocol provides a comprehensive set of security and functional aspects accounting for adequate computation, storage, and communication costs compared to state-of-the-art alternatives.

1 INTRODUCTION

With the rapid advancement of technology, mobile applications are becoming increasingly popular. One such application is the intelligent healthcare system, where patient health conditions are further improved via continuous monitoring of patient data stored on the remote server. The identity authentication and key agreement (AKA) gains wide attention in providing safe access (Trivedi and Patel, 2021) in this setting. Specifically, the three-factor AKA (3FAKA) provides higher security than one- or two-factor authentication (1FA or 2FA) and is fit for healthcare applications.

Figure 1 depicts a situation in which patients' data, such as heart rate, blood pressure, or body temperature, are collected using multiple body-mounted sensors and stored in the medical database. By collecting such data, a supervising doctor accesses the database containing his/her patients' information for diagnostic reference, allowing him/her to make a timely and accurate medical decision. Also, each patient's information could be accessible to other responsible divisions. Here, each doctor accesses patients' data through robust authentication with secret credentials (smart card, ID and password, and biometrics), followed by attribute-based access control (ABAC) strategy (Hu et al., 2013). However, there could be various attacks due to improper authentication. For example, the patient's rights and health may be risked if an outside foe or a malevolent insider illegally accesses the database and tampers sensitive data. Although AKA mitigates these issues (Roy et al., 2021), most recent 3FAKA systems are vulnerable, or their security is reduced to 1FA or 2FA, making them inadequate for practical use. Thus, balancing security, usability, and device cloning threats with valid 3FAKA is a persistent challenge.

1.1 Security Requirements

3FAKA should support below critical traits between user U and the server $\mathcal{S}$ in the presence of a foe $\mathcal{A}$:

F1 (User anonymity): The U can access system and being identified uniquely by $\mathcal{S}$ while $\mathcal{A}$ cannot identify U during public communication.

F2 (Untraceability): It allows $\mathcal{S}$ to identify the source of an openly conveyed message while $\mathcal{A}$ cannot do it.

F3 (Mutual authentication)**:** It is a critical property that enables $\mathcal{S}$ to identify U and vice versa.

F4 (Session key agreement)**:** It offers both $\mathcal{S}$ and U to

[a] https://orcid.org/0000-0001-5605-7354

Karati, A., Chang, Y. and Chen, T.
Robust Three-Factor Lightweight Authentication Based on Extended Chaotic Maps for Portable Resource-Constrained Devices.
DOI: 10.5220/0012077600003555
In *Proceedings of the 20th International Conference on Security and Cryptography (SECRYPT 2023)*, pages 673-682
ISBN: 978-989-758-666-8; ISSN: 2184-7711

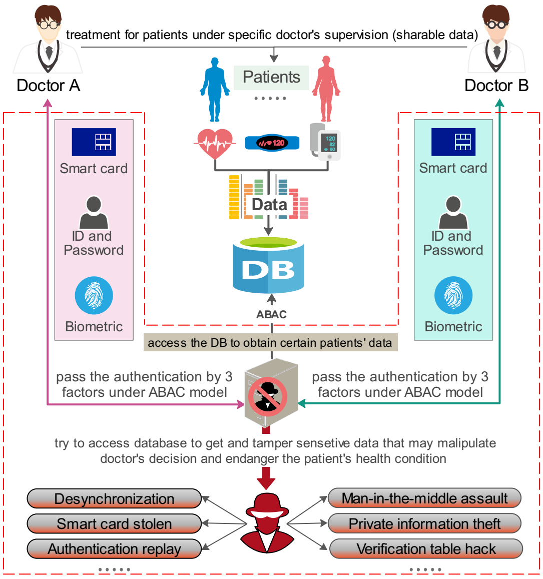
Doctor A
treatment for patients under specific doctor's supervision (sharable data)
Doctor B
Patients
Smart card
ID and Password
Biometric
Smart card
ID and Password
Biometric
Data
DB
ABAC
access the DB to obtain certain patients' data
pass the authentication by 3 factors under ABAC model
pass the authentication by 3 factors under ABAC model
try to access database to get and tamper sensetive data that may malipulate doctor's decision and endanger the patient's health condition
Desynchronization
Smart card stolen
Authentication replay
Man-in-the-middle assault
Private information theft
Verification table hack

Figure 1: A typical patient data access under mixed threat scenarios where attackers use public parameters and information-sharing flaws to sabotage medical services. Robust authentication (red box) prevents data invasion.

persuade a specific key for a single session.

F5 (Revoking smart cards remotely)**:** It permits $\mathcal{S}$ to revoke smart card of a suspected U remotely.

F6 (Dynamic User Addition Feature)**:** $\mathcal{A}$ may occasionally physically seize some Us' credentials due to a hostile environment. Therefore, allowing dynamic user addition (canceling ID but not the credentials enact) in the network is crucial to maintain safe services.

Besides, 3FAKA should withstand below attacks:

A1 (Resistant to desynchronization attack)**:** Upon effective authentication, the database stows certain values. The smart card may simultaneously alter specific values to sync the database, which aids U for the next login. During synchronization, $\mathcal{A}$ blocks communication and gains system access in the future. One should restrict such attacks or avoid synchronization.

A2 (Immune to stolen-data-and-password attack)**:** $\mathcal{A}$ cannot access the system even if U's password and data on the smartcard are known to $\mathcal{A}$.

A3 (Resistant to replay attack)**:** Valid data transmission in a replay attack is repeated maliciously, fraudulently, or delayed for system access. This kind of attack will be detected and refused.

A4 (Resistant to man-in-the-middle attack)**:** $\mathcal{A}$ sends and modifies messages between two parties who believe they are directly communicating in secret. Thus, only $\mathcal{S}$ and U use the same session key, regardless of the connection's weakness. Besides, $\mathcal{A}$ can only change the message after first informing $\mathcal{S}$ and U.

A5 (Resist stolen smartcard and biometric/password attack)**:** Even if $\mathcal{A}$ has U's smart card, $\mathcal{A}$ cannot pretend himself/herself as a specific U to access $\mathcal{S}$.

A6 (Stolen Verifier Table Attack)**:** Most protocols hold verifier tokens (such as hashed passwords) on $\mathcal{S}$, which is vulnerable to attack, rather than the users' real passwords or secret keys.

Note, we do not consider network traffic related attacks, such as jamming attack.

1.2 Related Works

Due to the open nature of wireless networks, an attacker may intercept, replay, and even modify shared data (Karati et al., 2021). In 2014, a 3FAKA protocol (Islam, 2014) was devised based on extended chaotic maps (ECM) and showed its security against various known attacks, such as smartcard loss attacks, while maintaining the three-factor security properties. However, the authors (Jiang et al., 2016) noted that the work (Islam, 2014) is susceptible to offline password guessing, smart card loss attacks, and biometric leaks due to incorrect password timely inspection mechanisms. To overcome the drawbacks, they devised a new 3FAKA with fuzzy verification (Jiang et al., 2016). In 2018, the authors (Das et al., 2018) developed a PUF-based authentication for wearable devices which necessitates a substantial number of computing operations and a longer execution time. Subsequently, the authors in (Roy et al., 2018) proposed a chaotic map-based anonymous authentication protocol with the fuzzy extractor for crowdsourcing Internet of Things (IoT); however, its verifier token was vulnerable to performing offline password guessing. In the multi-server setting, the authors (Chatterjee et al., 2018) designed a 3FAKA using the Chebyshev chaotic map, cryptographic hash, and symmetric key encryption/decryption. However, it does not provide proper three-factor security: the explicit password verifier enables attackers to guess passwords and identities offline when the other two factors (biometric and smart card) are compromised (Qiu et al., 2022). Shortly thereafter, a 3FAKA protocol is suggested in (Jiang et al., 2019) for cloud-assisted wearable devices based on the fuzzy extractor; however, it cannot provide PFS and X-security. In 2020, the authors in (Jiang et al., 2020) introduced a cloud-centric 3FAKA protocol that withstands many security measures, such as user anonymity, password confidentiality, and prompt typo detection. In addition, an IoT-based three-factor authentication method was developed in (Wang et al., 2020) to secure the system against all session key disclosure threats. Nonetheless, an enhanced anonymous authentication scheme for smart home was introduced by (Banerjee et al., 2020). In 2021, the authors in (Masud et al., 2021) presented a lightweight user authentication strategy for IoT-based healthcare to defend the network from impersonation and replay assaults and ensure data privacy and anonymity. Later, the authors in (Fakroon et al., 2021) applied PUF to safeguard IoT edge devices against cloning attacks and gave a logical analysis utilizing Burrows-Abadi-Needham (BAN) logic. Further, a mutual three-factor authentication using PUF was designed in (Kwon et al., 2021). It resists replay and man-in-the-middle attacks using AVISPA. The authors in (Yu et al., 2021) suggested an ECM-based AKA system resistant to verifier attacks for a multi-server scenario. Later, the authors in (Saqib et al., 2022) introduced a three-factor authentication system based on identity, password, and a digital signature mechanism for IoT-driven critical applications. It resists cryptographic attacks, such as man-in-the-middle, replay, and publisher impersonation attacks.

In 2022, a fast authentication for charging electric vehicles (EVs) was devised utilizing the ECM operation (Wang et al., 2022). The protocol ensured user anonymity while defending against insider assaults. Later, a blockchain-based privacy-preserving multi-factor device authentication was introduced for the cross-domain industrial IoT resistant to various known threats, including impersonation and replay attacks (Zhang et al., 2022). However, the scheme is vulnerable to user and server impersonation, data leakage, user traceability, and user revocation flaws (Ryu et al., 2022). Further, a three-factor authentication based on configurable ring oscillator (CRO)-PUF was presented by (Wang et al., 2023) that resists machine and password guessing attacks.

1.3 Motivation and Our Methodology

Technology for verifying identities has developed in numerous directions to accommodate comprehensive use cases. Every single approach to verification has its own set of flaws. Besides, most currently used authentication solutions do not provide adequate security to withstand modern assaults. Some systems based on single-factor (password-based) authentication (or reduced to single-factor) do not follow model security criteria. Nevertheless, many methods are secure yet inefficient due to inadequate cryptographic operations. This work suggests an efficient scheme for consumer devices to prevent sensitive data leakage. In the proposed scheme, we employ the chaotic map operation, which is feasible and more efficient than scalar multiplication and modular exponentiation. We achieve specific cryptographic traits, including unpredictability, non-repetition, and uncertainty. In addition, we use physically unclonable functions (PUFs) to provide hardware-based security through hardware fingerprints, which are essential for lightweight device authentication. It may be noted that the PUFs have specific characteristics, including randomness and physical unclonability, which increase the importance of the smart card's hardware entity. Thus, as long as the PUF readings are unavailable, capturing smartcard data and other secret information is insufficient for successful authentication.

1.4 Our Contributions

We design a safe 3FAKA protocol for low-power devices. Given the importance of hardware function, it offers a viable security solution. Within the context of this framework, we contribute the following:

- The system incorporates the ECM features to enable robust mutual authentication and session key agreement in lightweight devices.
- The suggested system uses PUFs to enhance the smartcard's hardware security and provides low-powered device-coupled authentication. Besides, it uses the Fuzzy Extractor to combat the growing prevalence of biometric verification in software and hardware to enhance usability.
- Under the chaotic-map computational Diffie-Hellman problem (CMCDHP), the scheme meets critical security aspects, such as mutual authentication, user anonymity, tight session key agreement, and untraceability. Besides, it resists desynchronization, stolen data and passwords, man-in-the-middle, replay, and stolen smartcard attacks.
- We estimate the efficacy of our scheme in authentication delay, transmission, and storage costs.

2 TECHNICAL PRELIMINARIES

This section describes the necessary backgrounds for comprehending the proposed work.

2.1 Fuzzy Extractor

With the prevalence of smart gadgets supporting biometric operations, the invariable traits of clients are gaining increasing attention. As a third authentication factor, biometrics such as fingerprints, facial characteristics, and iris are utilized to increase security. To improve usability, biometrics are stabilized with a fuzzy extractor so that the server can identify the user with an acceptable tolerance for error. A fuzzy extractor comprises two below functions, where l is the bit-length of the output string, χ is error tolerance, and $M \in \{0,1\}^v$ is the set of positive integers:

- $\mathsf{GEN}(\cdot)$: This probabilistic algorithm inputs a biometric $B_i \in M$ and outputs a secret $\sigma_i \in \{0,1\}^l$ and a support token T_i, where $\mathsf{GEN}(B_i) = \{\sigma_i, T_i\}$.
- $\mathsf{REP}(\cdot,\cdot)$: This deterministic algorithm inputs T_i, noisy biometric $B_i^l \in M$ and χ related to B_i, reproducing the biometric key σ_i. Moreover, we have $\mathsf{REP}(B_i^l, T_i) = \sigma_i$ where $d(B_i, B_i^l) \leq \chi$ is satisfied.

2.2 Physical Unclonable Functions

A physical unclonable function (PUF) is a chaotic system that maps a given set of challenges to a set of responses based on the device's microstructure. Cloning a PUF has proven extremely difficult, if not impossible. The functionality of Arbiter PUF may be modeled using an additive linear delay model (Rührmair et al., 2010). The total delay in each step is $\Delta = \vec{W}^T\vec{\Phi}$, where $\vec{W}$ is a feature vector representing the propagation delay of each MUX and $\vec{\Phi}$ is a function of n-bit challenge C= $\{c_i\}$. $\vec{\Phi}$ is denoted as:

$$\vec{\Phi}(\vec{C}) = \left(\vec{\Phi}^1(\vec{C}), \vec{\Phi}^2(\vec{C}), \vec{\Phi}^3(\vec{C}), \cdots, \vec{\Phi}^n(\vec{C}), 1\right)^T \quad (1)$$

where $\vec{\Phi}^j(\vec{C}) = \prod_{i=j}^{n}(i - 2c_i), \forall j \in [1,k]$. If $\Delta > 0$, the output r of Arbiter PUF is 1; else, 0. For convenience, use $e = (2*r) - 1$ to denote $e = \mathsf{sgn}(\Delta) = \mathsf{sgn}(\vec{W}^T\vec{\Phi})$, where sgn is a sign function. An XOR PUF with l individual Arbiter PUFs is denoted by l-COR PUF. The individual outputs of Aribiter PUF is denoted by $e_i \in \{-1,1\}$, where $i \in [0, l-1]$. The l-stage XOR gate is used to XOR all e_i to get the final response

$$e_{XOR} = \prod_{i=0}^{l-1}(e_i) \quad (2)$$

Consider each Arbiter PUF with a unique delay vector $\vec{W}_i$ shares Φ. Then, we have $e_{XOR} = \mathsf{sgn}(\prod_{i=0}^{l-1}\vec{W}^T\vec{\phi})$.

Moreover, based on the integrated circuit (IC) features, a PUF[1] generates unique challenge-response pair $R = PUF(C)$, where R is a response to a unique challenge C. An integrated PUF is a d-bit PUF composed of d PUFs, each responding with a single bit. The properties of an ideal d-bit PUF are as follows:

Fact 1. *For C_1 in two PUF_1 and PUF_2, the Hamming distance $HD(PUF_1(C_1), PUF_2(C_1)) \approx d/2$.*

Fact 2. *For two distinct challenges C_1 and C_2, we have $HD(PUF_1(C_1), PUF_2(C_2)) \approx d/2$.*

Fact 3. *For a random challenge C_1, we have the Hamming distance $HD(PUF_1(C_1), PUF_1'(\ C_1)) = 0$. (note, tampered PUF_1' contains a $d/10$ bit error.)*

Fact 4. *$HD(PUF_1(C_1), PUF_1(C_2)) \approx d/2$ for any d-bit PUF_1 with any two challenges C_1 and C_2.*

The proposed solution depends on the notion that these features of PUFs may be exploited to create physically secure protocols for device authentication.

2.3 Extended Chaotic Maps

Given $n \in Z^+$ and $-1 \leq x \leq 1$, the Chebyshev chaotic polynomial (CCP) $T_n(x) : [-1,1] \rightarrow [-1,1]$ is defined as $T_n = \cos(n \cdot cos^{-1}(x))$, or

$$T_n(x) = \begin{cases} 1, & \text{if } n = 0 \\ x, & \text{if } n = 1 \\ 2xT_{n-1}(x) - T_{n-2}(x), & \text{if } n \geq 2. \end{cases} \quad (3)$$

The CCP satifies the semi-group property: $T_r(T_s(x)) = T_s(T_r(x)) = cos(rs \cdot cos^{-1}(x))$ for $r, s \in N$. However, CCP based public key cryptography is not secure. Later, (Zhang, 2008) defined the *extended* Chebyshev polynomial as $T_n = 2xT_{n-1}(x) - T_{n-2}(x)$ mod p, $\forall n \geq 2$ that holds the semi-group property in interval $(-\infty, +\infty)$, where p is a large prime.

Definition 1 (Extended Chaotic Map)**.** *Given prime p and $x \in (-\infty, +\infty)$, the extended Chebyshev polynomial holds $T_r(T_s(x)) \equiv T_s(T_r(x)) \equiv T_{rs}(x) \pmod{p}$.*

Definition 2 (Chaotic map discrete logarithm problem (CMDLP))**.** *Given x and y, it is infeasible for $\mathcal{A}$ to find r such that $T_r(x) = y$. The advantage of $\mathcal{A}$ is:*

$$\varepsilon_{\mathcal{A}}^{\mathsf{CMDLP}} = \Pr[\mathcal{A}(x,y) = r : r \in Z_p^*, y = T_r(x) \ \% \ p] \quad (4)$$

Definition 3 (Chaotic map computational Diffie-Hellman problem (CMCDHP))**.** *Given (x, T_s, T_m), it is infeasible for $\mathcal{A}$ to find $T_{ms}(x)$. The advantage of $\mathcal{A}$ is:*

$$\varepsilon_{\mathcal{A}}^{\mathsf{CMCDHP}} = \Pr[\mathcal{A}(x, T_s, T_m) = r : r \equiv T_{ms}(x) \ \% \ p] \quad (5)$$

3 OUR CONSTRUCTION

Table 1 introduces necessary notations. We presume there must be a device capable of extracting biometrics features and a smart card capable of performing PUF and storing user-specific tokens. The proposed scheme comprises the six phases listed below:

3.1 System Setup

This phase is executed by the server $\mathcal{S}$ to initialize specific parameters prior to user registration and other

[1] While PUF is reliant on the features of the IC, its reaction may vary owing to environmental factors.

associated actions. It sets a large prime $p \equiv 3 \pmod 4$ and considers an admissible ECM as $\mathsf{T}_x(y)$ where $x, y \in Z_p^*$. Besides, it selects a one-way hash function $H(\cdot) \in \mathcal{H}$. Now, it generates a high entropy secret key s at random for performing standard symmetric encryption $\mathsf{SE}[\cdot]$ and decryption $\mathsf{SD}[\cdot]$. Then, it chooses a list $\mathcal{L}(ID, Valid\_CID)$, which is initially empty. To protect against thefts for actual ID and $Valid\_CID$ at the server side, $\mathcal{S}$ adopts a honey-list $\mathcal{L}'$ as shown in (Juels and Ristenpart, 2014) to safeguard $\mathcal{L}$ against potential attackers. After successful authentication, we assume that $\mathcal{S}$ employs the ABAC model to determine user authorization over sensitive data.

3.2 User Registration

As depicted in Fig. 2, on a viable request from a user U_i, the $\mathcal{S}$ provides an initially empty smartcard SC_i depending on U_i's level. Now, U_i selects PW_i and generates $\gamma = \mathsf{PUF}(PW_i)$. Next, it provides a biometric sample Bio_i and generates σ_i with the fuzzy extractor as $(\sigma_i, \Gamma_i) = \mathsf{GEN}(Bio_i)$. Besides, it computes feature value $FV_i = H(PW_i, \sigma_i, \gamma)$. Finally, U_i sends the $R_1 = (ID_i, \gamma, FV_i)$ to the server $\mathcal{S}$.

Upon obtaining R_1, $\mathcal{S}$ ABORTs the session if $ID_i \in \mathcal{L}$. Otherwise (i.e., $ID_i \notin \mathcal{L}$), $\mathcal{S}$ assigns a token CID_i where $CID_i = 1$ and inserts $(ID_i, 1) \in \mathcal{L}$. Now, $\mathcal{S}$ generates $y_i = \mathsf{SE}_s[ID_i, FV_i, CID_i]$ and an access token $T_i = \mathsf{T}_s(\gamma)$. Consequently, it updates honey-list $\mathcal{L}'$. Finally, $\mathcal{S}$ updates SC_i that contains (y_i, T_i, ID_i, CID_i).

Now, U_i generates $VPW = H(PW_i, ID_i, \sigma_i, CID_i)$ and updates the received smart card as $SC_i = (y_i, T_i, VPW, ID_i, CID_i, \Gamma_i)$. The communications between U_i and $\mathcal{S}$ occur securely.

3.3 Authentication and Key Agreement

$\boldsymbol{U_i \to \mathcal{S}}$ **:** When U_i wants to access the system, he/she provides his/her ID, password, and biometrics to compute $\gamma' = \mathsf{PUF}(PW_i^{'})$, $\sigma_i^{'} = \mathsf{REP}(Bio_i^{'}, \Gamma_i)$. Next, U_i generates $VPW' = H(PW_i^{'}, ID_i, \sigma_i^{'}, CID_i)$ and checks whether password and biometrics are correct as $VPW' \stackrel{?}{=} VPW$. Then, U_i computes $K = \mathsf{T}_u(T_i)$ and chooses r_1 at random. Next, it performs symmetric key encryption by $C_1 = \mathsf{SE}_K[ID_i || y_i || r_1]$. After that, U_i sends $L_1 = (C_1, T_i' = \mathsf{T}_u(\gamma'))$ to $\mathcal{S}$.

$\boldsymbol{\mathcal{S} \to \mathbf{U_i}}$ **:** Once $\mathcal{S}$ receives L_1, it computes $K' = \mathsf{T}_s(T_i')$ to decrypt C_1 and further y^* as

$$\mathsf{SD}_{K'}[C_1] \longrightarrow (ID_i^*, y_i^*, r_1^*) \tag{6}$$

$$\mathsf{SD}_{\,s}\,[y_i^*] \longrightarrow (ID_i^o, FV_i^o, CID_i^o) \tag{7}$$

Now, $\mathcal{S}$ checks whether $ID_i^o \stackrel{?}{=} ID_i^*$ exists in $\mathcal{L}$. If it does not exist or $(CID_i^o \neq Valid\_CID_i)$, $\mathcal{S}$ terminates the session. If it exists with $Valid\_CID_i = -1$, then $\mathcal{S}$ ABORTs the session notifying U_i that the SC_i has expired. Otherwise, $\mathcal{S}$ chooses r_2 at random and masks it as $\alpha = r_1 \oplus r_2$. Then, it computes $\beta = H(r_1 || r_2)$, and sends $L_2 = (\alpha, \beta)$ back to U_i.

Table 1: List of useful notations.

Notation	Description
p	A sufficiently large prime holds 3 *mod* 4
ID_i, PW_i	Identity and password of user U_i
Bio_i	Biometric data of user U_i
σ_i	Reproducible data generated by $\mathsf{GEN}(\cdot)$
Γ_i	Assisting token to output σ_i by $\mathsf{REP}(\cdot)$
γ	An error-tolerant PUF response
$\mathsf{T}_x(y)$	Extended chaotic map T for input (x, y)
y_i	$\mathcal{S}$ derived token for U_i with secret s
$\mathsf{SE}_x[M]$	Symmetric encoding of M for secret x
$\mathsf{SD}_x[C]$	Symmetric decoding of C for secret x
$H(\cdot)$	Secure cryptographic hash function
$\mathsf{PUF}(\cdot)$	Physical unclonable function for SC_i
$A \oplus B$	Bit-wise XOR operation A and B
$A \vert\vert B$	Concatenation between A and B

$\boldsymbol{U_i \to \mathcal{S}}$ **:** On receiving $L_2 = (\alpha, \beta)$, U_i discloses $r_2 = \alpha \oplus r_1$ and verifies whether $\beta \stackrel{?}{=} H(r_2 || r_1)$ for resisting replay attack. Finally, U_i performs symmetric encryption and sends L_3 to $\mathcal{S}$ where

$$L_3 = \mathsf{SE}_{H(K || r_2)}[H(\sigma_i^{'}, \gamma_i^{'}, PW_i^{'}) || r_2] \tag{8}$$

Since $\mathcal{S}$ has knowledge of K' and r_2, it reveals cleartext $\leftarrow \mathsf{SD}_{H(K'||r_2)}[L_3]$. Next, it checks whether $FV_i^o \stackrel{?}{=} FV_i'$. If it holds, $\mathcal{S}$ generates the session key as $SK = H(r_1 || r_2 || K')$. Note that U_i can compute SK. Finally, $\mathcal{S}$ allows U_i accessing sensitive data based on ABAC (Hu et al., 2013) for traced ID_i, and SK is used for any data exchange between $\mathcal{S}$ and U_i.

3.4 User Password Update

When U_i wants to change his/her password PW_i, U_i must pass the authentication successfully to communicate with $\mathcal{S}$ with the session key SK. For this, U_i follows the steps outlined in Section 3.3. Now, it chooses a password PW_i^{new} complies with password rules and sets $\gamma_i^{new} = \mathsf{PUF}(PW_i^{new})$ and updates feature value as $FV_i = H(PW_i^{new}, \sigma_i^{'}, \gamma_i^{new})$. When $\mathcal{S}$ reveals γ_i and FV_i^{new} using symmetric decryption for key SK, it updates $CID_i^{new} = Valid\_CID + 1$ to record the times that U_i changes the password. Then, $\mathcal{S}$ generates

$$y_i^{new} = \mathsf{SE}_s[ID_i, FV_i^{new}, CID_i^{new}] \tag{9}$$

Next, it computes $\mathsf{T}s(\gamma_i^{new})$ and updates the smartcard data $SC^* = (y_i{}^*, T_s(\gamma_i^*), ID_i, CID_i{}^*)$ securely with SK. Now, U_i sets $VPW_i^{new} = H(PW_i^{new}, ID_i, \sigma_i^{'}, CID_i^{new})$. The smartcard data will eventually be refreshed as $SC_i^{new} = (y_i^{new}, \mathsf{T}s(\gamma_i^{new}), VPW_i^{new}, ID_i, CID_i^{new}, \Gamma_i)$.

3.5 Dynamic Device Addition

At this phase, a device is considered an end user. Assume U_i^{new} has to be added to the network after the first deployment. U_i^{new} first sends $ID_i^{new}, \gamma_i^{new}, FV_i^{new}$ to $\mathcal{S}$. If ID_i^{new} does not already exist in $\mathcal{L}$, then $\mathcal{S}$ allows U_i^{new} to execute the registration phase. However, if $ID_i^{new} \in \mathcal{L}$, then the following actions are taken:

- If $Valid\_CID_i = -1$ which indicates that U_i^{new} has been revoked, $\mathcal{S}$ aborts the session of U_i^{new}.
- If $Valid\_CID_i = 0$ (expired smartcard), then $\mathcal{S}$ sets $CID_i^{new} = Valid\_CID_i + 1$. Next, $\mathcal{S}$ issues a new SC_i^{new} for user identifier $\gamma_i^{new} = PUF_i^{new}(PW_i^{new})$ and $FV_i^{new} = H(\sigma_i^{new}, \gamma_i^{new}, PW_i^{new})$. Next, $\mathcal{S}$ decodes y_i from expired SC_i^{old} and checks $CID_i \overset{?}{=} CID_i^{new}$. If it holds, $\mathcal{S}$ collects $y_i^{new} = SE_s(ID_i^{new}, FV_i^{new}, CID_i^{new})$ and sets $SC_i^{new} = (y_i^{new}, \mathsf{T}_s(\gamma_i^{new}), ID_i^{new}, CID_i^{new})$. Now, U_i^{new} sets new $VPW_i^{new} = H(PW_i^{new}, ID_i^{new}, \sigma_i, CID_i^{new})$ and updates SC_i^{new} as $\{y_i^{new}, \mathsf{T}_s(\gamma_i^{new}), VPW_i^{new}, ID_i^{new}, CID_i^{new}, \Gamma_i\}$.
- If $Valid\_CID_i = 1$, then U_i already exists and attempts to add itself again. For this, $\mathcal{S}$ permits U_i^{new} to execute the Password Change phase.

3.6 User Revocation

When an identity has been tracked for malicious conduct, one of the most pressing concerns is how to revoke its login credentials. Tracking hazardous behaviors may be discovered by repeatedly providing incorrect credentials, ABAC, $\mathcal{L}'$, or smart behavior analysis, which is not the focus of this work. We enable $\mathcal{S}$ to revoke U_i's authentication credentials remotely. To do this, $\mathcal{S}$ authentically access its own maintained $\mathcal{L}$ and updates tuple $(ID_i, Valid\_CID)$ to $(ID_i, -1)$.

4 SECURITY DISCUSSION

Our 3FAKA achieves several security properties as mentioned in Section 1.1. Here, we consider $\mathcal{S}$ as a trusted entity while the attacker $\mathcal{A}$ is a vulnerable entity. The PUF with the SC is considered a System on Chip (SoC). Any attempt to tamper or separate the PUF from a SC renders the PUF useless (Kirkpatrick et al., 2014). Besides, the communication between SC and PUF is considered secure (Aman et al., 2018).

Lemma 1. *The response (R) of a PUF cannot be envisioned. The advantage of $\mathcal{A}$ in finding R is $1/2^n$.*

Proof. An admissible PUF function is defined as $\mathsf{PUF} : \{0,1\}^m \rightarrow \{0,1\}^n$ for some positive integers m and n. The security of a PUF can be modeled using a game $G_{\mathcal{A}}^{\mathsf{PUF}}$ between a challenger $\mathcal{S}$ and a probabilistic polynomial-time (PPT) algorithm run by adversary $\mathcal{A}$. The details are provided below:

T1: $\mathcal{A}$ sends a random challenge C_i to $\mathcal{S}$. The $\mathcal{S}$ reveals $R_i = \mathsf{PUF}(C_i)$ to $\mathcal{A}$. This is an adaptive process executed t_1 times.

C: The $\mathcal{S}$ chooses a challenge C_x at random (not queried by $\mathcal{A}$ before). Then, it sends C_x to $\mathcal{A}$ while keeps $R_x = \mathsf{PUF}(C_x)$ safely.

T2: $\mathcal{A}$ can send any query C_j to $\mathcal{S}$ where $C_j \neq C_x$ and $C_j \neq C_i$. The $\mathcal{S}$ reveals $R_j = \mathsf{PUF}(C_j)$ to $\mathcal{A}$. This is an adaptive process executed t_2 times where the total number of queries is $q = t_1 + t_2$.

G: $\mathcal{A}$ outputs its guess R^* to $\mathcal{S}$ for the challenge C_x and wins the game if $R^* = R_x$.

$\mathcal{A}$ wins the above game with an advantage $\varepsilon_{\mathcal{A}}^{\mathsf{PUF}} = \Pr[R^* \leftarrow \mathcal{A}(C_x, (C_i, R_i)) : C_x \neq C_{i\in[1,q]} \wedge R^* = R_x]$. However, each PUF responds uniquely and cannot be cloned (Kirkpatrick et al., 2014). Thus, $\mathcal{A}$ guesses $R_x \in_R \{0,1\}^n$. Hence, $\mathcal{A}$ has $\varepsilon_{\mathcal{A}}^{\mathsf{PUF}} = 1/2^n$. □

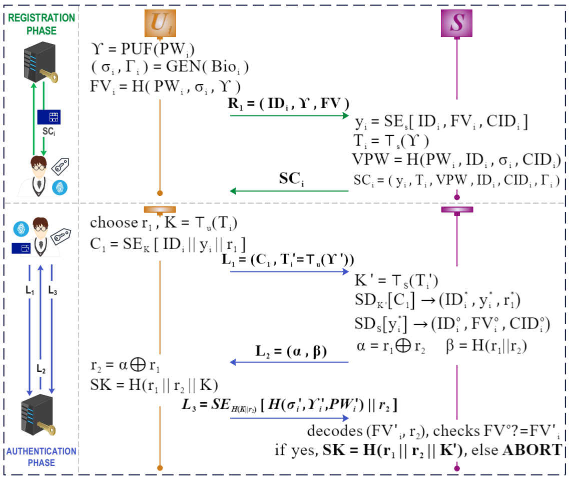
REGISTRATION PHASE
U_i
$\mathcal{S}$
SC$_i$
ϒ = PUF(PW$_i$)
(σ$_i$, Γ$_i$) = GEN(Bio$_i$)
FV$_i$ = H(PW$_i$, σ$_i$, ϒ)
R$_1$ = (ID$_i$, ϒ, FV)
y$_i$ = SE$_s$[ID$_i$, FV$_i$, CID$_i$]
T$_i$ = T$_s$(ϒ)
VPW = H(PW$_i$, ID$_i$, σ$_i$, CID$_i$)
SC$_i$
SC$_i$ = (y$_i$, T$_i$, VPW, ID$_i$, CID$_i$, Γ$_i$)
choose r$_1$, K = T$_u$(T$_i$)
C$_1$ = SE$_K$[ID$_i$ || y$_i$ || r$_1$]
L$_1$ = (C$_1$, T$_i$'=T$_u$(ϒ'))
K' = T$_S$(T$_i$')
SD$_{K'}$[C$_1$] → (ID$_i^*$, y$_i^*$, r$_1^*$)
SD$_S$[y$_i^*$] → (ID$_i^\circ$, FV$_i^\circ$, CID$_i^\circ$)
α = r$_1$ ⊕ r$_2$ β = H(r$_1$||r$_2$)
L$_2$ = (α, β)
r$_2$ = α ⊕ r$_1$
SK = H(r$_1$ || r$_2$ || K)
L$_3$ = SE$_{H(K||r_2)}$[H(σ$_i$', ϒ$_i$', PW$_i$') || r$_2$]
decodes (FV'$_i$, r$_2$), checks FV°?=FV'$_i$
if yes, SK = H(r$_1$ || r$_2$ || K'), else ABORT
L$_1$
L$_2$
L$_3$
AUTHENTICATION PHASE

Figure 2: Registration and authentication of our 3FAKA.

Theorem 1. *Our 3FAKA protocol supports critical security traits, namely user anonymity, untraceability, mutual authentication, and session key agreement.*

Proof. The theorem follows when Lemmas 2-5 according to Definitions 2-3 and Lemma 1 are hold. □

Lemma 2 (F1). *The proposed 3FAKA protocol maintains strong user anonymity.*

Proof. Valid U_i and server $\mathcal{S}$ exchange $\langle L_1, L_2, L_3 \rangle$ during authentication. To violate user anonymity, $\mathcal{A}$ must trace the user identifier hidden in $L_1 = (C_1, T_i')$. Specifically, $\mathcal{A}$ needs to decrypt $C_1 = \mathsf{SE}_K[ID_i||y_i||r_1]$

encrypted with $K = \mathsf{T}_u(\mathsf{T}_s(\gamma))$. Although $\mathcal{A}$ has $T_i' = \mathsf{T}_u(\gamma)$, it cannot divulge K since the server secret s is unavailable. As C_1 is seen as a secure trapdoor for K; user identification is infeasible and only visible to $\mathcal{S}$ who issued SC_i to U_i. Hence, the suggested protocol achieved strong user anonymity. □

Lemma 3 (F2). *The 3FAKA supports untraceability.*

Proof. Even though $\mathcal{A}$ cannot uniquely identify a user during authentication, it may determine the relevance between the user and data. This can be accomplished by monitoring the channel and recording the relationship between anonymous ID and data $AR_1 = \langle L_1, L_2, L_3 \rangle$ is coming from which anonymous ID. The suggested protocol randomizes each L_j during each session. Even if the same Ui initiates several authentication requests $(AR_1, \cdots, AR_n)$ at different times, $\mathcal{A}$ cannot trace such AR_i belongs to the same anonymous U_i. To achieve randomization in AR_i, U_i generates L_1 for some u, r_1 chosen at random and L_3 with a random secret key K. Besides, $\mathcal{S}$ generates L_2 with some random r_2. One may observe that u, r_1, r_2, K are random data for different sessions. Hence, our scheme meets untraceability. □

Lemma 4 (F3). *The proposed 3FAKA protocol maintains strong mutual authentication.*

Proof. During login, U_i sends $L_1 = (C_1, T_i' = \mathsf{T}_u(\gamma))$ to $\mathcal{S}$. Notably, we incorporate U_i's identity ID_i, identification token y_i and and random r_1, then utilize key $K = \mathsf{T}_u(\mathsf{T}_s(\gamma))$ to generate $C_1 = \mathsf{SE}_K(ID_i||y_i||r_1)$ to $\mathcal{S}$. With secret s, $\mathcal{S}$ generates $K' = T_s(T_u(\gamma))$ and discloses ID_i, y_i and r_1 from C_1 to authenticate U_i. Due to the hardness of CMCDHP, one may note that $\mathcal{A}$ cannot generate K' even if it reveals hard token $\mathsf{T}_s(\gamma)$. Following that, $\mathcal{S}$ computes $\alpha = r_1 \oplus r_2$ and $\beta = H(r_1||r_2)$ for some r_2. Finally, it returns $L_2 = (\alpha, \beta)$ to U_i. Now, U_i can authenticate $\mathcal{S}$ if it receives r_1 details from L_2. This is because r_1 can be derived from C_1 by the $\mathcal{S}$ with secret s. On receiving L_2, U_i finds $\mathcal{S}$'s credibility through β. Besides, $\mathcal{S}$ finds U_i's credibility when it receives valid FV_i through L_3. Further, $\mathcal{A}$ cannot disclose any sensitive information that allows $\mathcal{A}$ to authenticate on behalf of $\mathcal{S}$ and U_i. Thus, our scheme meets mutual authentication. □

Lemma 5 (F4). *Our 3FAKA protocol achieves strong session key agreement during authentication.*

Proof. The strong key agreement relies on the fact that $\mathcal{S}$ and U_i agree on some session-dependant random tokens that are never transmitted as cleartext throughout authentication exchange for a particular session. On viable authentication, both $\mathcal{S}$ and U_i agree on the same session key $SK = H(r_1||r_2||K)$, where r_1 and r_2 are random tokens generated by U_i and $\mathcal{S}$ respectively and K is session-dependant symmetric key generated as $T_s(T_u(\gamma))$ using U_i's nonce u and $\mathcal{S}$'s secret s. Besides, $\mathcal{S}$ agrees on r_1 to send its nonce r_2 in L_2. Similarly, when receives r_1 details in L_2, U_i agrees on r_2 to send L_3. When $\mathcal{S}$ reveals FV_i from L_3 and authenticate it, then $\mathcal{S}$ agrees on SK. Thus, the suggested protocol achieves the session key agreement. □

Lemma 6 (F5). *The proposed 3FAKA protocol revokes the issued cards remotely.*

Proof. Since $\mathcal{S}$ controls $\mathcal{L}$, it may officially revoke a smart card SC_i by setting $Valid\_CID_i = -1$ of a particular user U_i. It impedes $\mathcal{A}$'s capability to authenticate successfully. Because CID_i and $Valid\_CID_i$ are distinct, $\mathcal{A}$ cannot pass authentication. Hence, our method remotely revokes the smartcard. □

Theorem 2. *The proposed 3FAKA protocol resists essential security attacks for lightweight devices.*

Proof. The theorem follows when Lemmas 6-11 according to Definitions 2-3 and Lemma 1 are hold. □

Lemma 7 (A1). *The proposed 3FAKA protocol is resistant to desynchronization attacks.*

Proof. Synchronization of server-side data, including PUF responses and hash-chain values, is frequent in many authentications and key agreement systems. So, several potential vulnerabilities exist due to the severity of data synchronization. The proposed solution, however, avoids the possibility of covert data updates occurring with each user authentication. Hence, our technique withstands desynchronization attacks. □

Lemma 8 (A2). *The proposed 3FAKA protocol withstands stolen-data-and-password attacks.*

Proof. Suppose that $\mathcal{A}$ has access to U_i's password PW_i and the smartcard SC_i data. Because of the PUF characteristics and the unavailability of nonce r_1, $\mathcal{A}$ can still not access the system. In particular, $\mathcal{A}$ requires token $\gamma = \mathsf{PUF}_i(PW_i)$. According to Lemma 1, if $\mathcal{A}$ cannot obtain PUF_i, the probability of achieving the correct γ is $1/2^{|\gamma|}$, which corresponds to guessing a nonce with high-entropy. Thus, the suggested scheme resists stolen-data-and-password assaults. □

Lemma 9 (A3). *The proposed 3FAKA protocol is secure against replay attacks.*

Proof. Assume that $\mathcal{A}$ tries to replay any of L_1, L_2, and L_3 sent in session S_i to another session S_j. If $\mathcal{A}$ replays L_1, then it cannot continue sending L_3 as it cannot reveal r_2 due to the unavailability of r_1 hidden in L_1. Besides, it cannot reveal r_1 due to the hardness

Table 2: Security functionalities comparisons of the existing schemes.

Scheme	Security Traits (symbols as in Section 1.1)											
	F1	F2	F3	F4	F5	F6	A1	A2	A3	A4	A5	A6
W1 (Jiang et al., 2019)	✓	✓	✓	×	×	×	✓	✓	✓	×	×	×
W2 (Zhou et al., 2019)	×	×	✓	✓	×	×	✓	×	×	×	×	×
W3 (Banerjee et al., 2020)	✓	✓	×	×	×	×	×	×	✓	✓	✓	×
W4 (Kwon et al., 2021)	✓	×	✓	×	×	×	×	×	✓	✓	×	✓
W5 (Qiu et al., 2022)	✓	✓	✓	×	×	×	✓	×	✓	✓	×	×
W6 (Wang et al., 2023)	✓	×	✓	✓	×	×	✓	×	✓	✓	×	×
Our 3FAKA scheme	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

of symmetric trapdoor C_1. Similarly, if $\mathcal{A}$ replays L_2, then U_i can detect it due incorrect r_1 details in L_2. Nonetheless, replaying L_3 will not be successful as $\mathcal{S}$ can detect it for incorrect r_2. In the proposed protocol, $\mathcal{S}$ and $\mathcal{U}$ produce challenge tokens containing random integers based on the random responses from each entity during each session. Hence, the suggested protocol is resistant to replay assaults. □

Lemma 10 (A4). *The proposed 3FAKA protocol safeguards against a man-in-the-middle attack.*

Proof. Assume that $\mathcal{A}$ tries to alter the data sent by U_i during the session. If $\mathcal{A}$ wants to alter $L_1 = (C_1, T_i')$ where $C_1 = \mathsf{SE}_K[ID_i||y_i||r_1]$ and $T_i' = \mathsf{T}_u(\gamma)$. Suppose $\mathcal{A}$ tries to send $C_1' = \mathsf{SE}_K[ID_j||y_j||r_1]$ for ID_j along with T_i'. However, computation of a valid y_j requires $\mathcal{S}$'s secret s. For a random y_j^*, $\mathcal{S}$ detects $ID_j \neq ID_j^*(\leftarrow \mathsf{SD}_s[y_i^*])$. Thus, successful modification in L_1 is not feasible. Similarly, any data tampering in $L_2 = (\alpha = r_1 \oplus r_2, \beta = H(r_1||r_2))$ and $L_3 = SE_{H(K||r_2)}[H(\sigma^*, \gamma^*, PW^*)||r_2]$ is hard due to the CMCDHP assumption and Lemma 1. Besides, such changes will be detected by U_i with r_1 and $\mathcal{S}$ with r_2, respectively. Hence, the suggested protocol safeguards against the man-in-the-middle attack. □

Lemma 11 (A5). *The proposed 3FAKA protocol is resistant to a stolen smartcard and password attack.*

Proof. Suppose $\mathcal{A}$ has stolen U_is smartcard $SC_i = (y_i, T_i, VPW, ID_i, CID_i, \Gamma_i)$ and use it on behalf of U_i to access the system. To pass authentication by $\mathcal{S}$, $\mathcal{A}$ uses SC_i and generates $L_1 = (C_1, T_i')$ where $C_1 = \mathsf{SE}_K[ID_i||y_i||a_1]$ and $T_i' = \mathsf{T}_a(\gamma_i)$ for some a and $\gamma_i = \mathsf{PUF}_i(PW_i)$. Clearly, $\mathcal{S}$ accepts L_1 and $\mathcal{A}$ may receive r_2. However, $\mathcal{A}$ cannot produce a valid L_3 due to the unavailability of biometric Bio_i. If it tries to tamper it and compute as $L_3 = SE_{H(K||r_2)}[H(\sigma_i^f ake, \gamma_i, PW_i)||r_2]$, then $\mathcal{S}$ easily detect such issue as FV_i decrypted from y_i does not match with $H(\sigma_i^{fake}, \gamma_i, PW_i)||r_2]$. Hence, our scheme resists the stolen smartcard-and-password attack. □

Lemma 12 (A6). *The proposed 3FAKA protocol is resistant to stolen verifier table attack.*

Proof. A stolen-verifier attack occurs when $\mathcal{A}$ obtains a password or secret verifier from $\mathcal{S}$ to impersonate a legitimate user. Our 3FAKA employs a database, say $\mathcal{D}$, to assign IDs and Valid_CIDs to users. $\mathcal{D}$ stores no passwords and checks if a user is revoked during authentication. Thus, updating $\mathcal{D}$ for each authentication request is nonessential. Even if $\mathcal{A}$ breaches $\mathcal{D}$ security, it cannot mimic U_i during user authentication. Thus, our 3FAKA resists stolen verifier attacks. □

5 PERFORMANCE DISCUSSION

Table 2 compares the security features of the related schemes. One may note that the proposed 3FAKA achieves comprehensive security properties compared to others. Our protocol could be implemented with a lightweight 128-bit ASCON or AES cryptosystem with standard SHA-2 algorithms. Table 3 compares our scheme with existing related schemes considering specific overheads as discussed below.

5.1 Registration and Authentication

For user registration, U_i executes one $\mathsf{PUF}(\cdot)$, one FE_{GEN}, and two $H(\cdot)$ operations while server $\mathcal{S}$ runs one $SE(\cdot)$ and one $\mathsf{T}.(\cdot)$. Thus, the registration cost is estimated as $C_{reg} = T_{PUF} + T_{FE} + T_{ECM} + 2T_H + T_{SE}$.

During an interactive authentication process in our 3FAKA, U_i requires one $\mathsf{PUF}(\cdot)$, one FE_{REP}, one $\mathsf{T}.(\cdot)$, five $H(\cdot)$, two $\mathsf{SE}(\cdot)$ and three $\mathsf{SD}(\cdot)$. Thus, the user's burden is $C_1 = T_{PUF} + T_{FE} + T_{ECM} + 4T_H + 2T_{SE}$. On the other hand, $\mathcal{S}$ needs one $\mathsf{T}.(\cdot)$, three $H(\cdot)$, and three $\mathsf{SD}(\cdot)$ operations. Thus, the server's load is $C_2 = T_{ECM} + 3T_H + 3T_{SD}$. Hence, the overall authentication overhead is considered as $C_{auth} = C_1 + C_2 = T_{PUF} + T_{FE} + 2T_{ECM} + 7T_H + 2T_{SE} + 3T_{SD}$.

Further, W1 requires one T_{FE} and three T_H operations during registration. Thus the overhead is $T_{FE} + 3T_H$. Besides, it runs one T_{FE} and twenty T_H during user authentication. Thus, the overhead is $T_{FE} + 20T_H$. Similarly, one may compute costs for W2-W5. Work W6 uses PUF to provide hardware security. It takes one T_{PUF}, one T_{FE}, and six T_H to reg-

Table 3: Computation cost comparisons of related schemes.

Scheme	Registration Overhead	Authentication Overhead
W1 (Jiang et al., 2019)	$T_{FE}+3T_H$	$T_{FE}+20T_H$
W2 (Zhou et al., 2019)	$5T_H$	$42T_H$
W3 (Banerjee et al., 2020)	$T_{FE}+4T_H$	$T_{FE}+25T_H$
W4 (Kwon et al., 2021)	$T_{FE}+5T_H$	$2T_{FE}+T_{PUF}+32T_H$
W5 (Qiu et al., 2022)	$T_{FE}+T_{ECM}+4T_H$	$T_{FE}+6T_{ECM}+19T_H$
W6 (Wang et al., 2023)	$T_{PUF}+T_{FE}+6T_H$	$3T_{PUF}+2T_{BP}+T_{FE}+12T_H+3T_{PEA}$
Our 3FAKA Scheme	$T_{PUF}+T_{ECM}+T_{FE}+2T_H+T_{SE}$	$T_{PUF}+2T_{ECM}+T_{FE}+7T_H+2T_{SE}+3T_{SD}$

T_{PUF} : Time to execute one PUF$(\cdot)$; T_{FE} : Time to run either GEN$(\cdot)$ or REP$(\cdot)$; T_{ECM} : Time to execute one $T_{.}(\cdot)$; T_{SE} : Cost for one SE$(\cdot)$ T_{SD} : Cost for one SD$(\cdot)$; T_H : Cost for one $H(\cdot)$; T_{BP}: Cost for one bilinear pairing operation; T_{PEA}: Paillier Ecrypt/Decrypt operation.

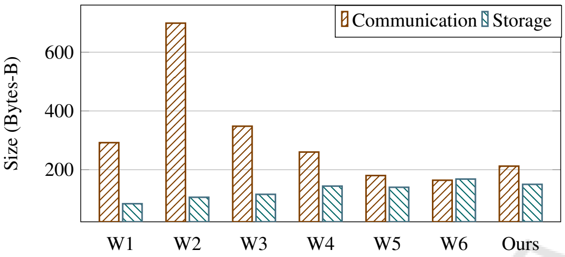

Figure 3: Computational and storage costs comparison.

ister a user. Thus, the total cost is $T_{PUF}+T_{FE}+6T_H$. Besides, it considers three T_{PUF}, two T_{BP}, one T_{FE} twelve T_H and three T_{PEA}. Hence, the overhead is considered as $T_{PUF}+2T_{BP}+T_{FE}+12T_H+3T_{PEA}$.

5.2 Token Transmission and Storage

During authentication, several data as authentication tokens are transmitted. To measure the various costs, we approximate the length of the security parameters as follows: $|ID_i| = 16$ bytes (B), $|Z_p^*| = 20$ B, hash $|h| = 32$ B, random $|r| = 16$ B and each component is of 16 B in the output of the function $GEN(\cdot)$. Authentication is valid when all L_1, L_2, L_3 are successfully transmitted. The U_i initiates $L_1 = (C_1, T_i')$, where C_1 is a 96 B ciphertext and T_i' is an element in Z_p^*. Thus, U_i sends $|L_1| = 116$ B. Now, $\mathcal{S}$ sends $L_2 = (\alpha, \beta)$ where α is 16 B masked data and $\beta = 32$ B integrity token. Finally, U_i sends L_3 which is of 48 B. Hence, the overall, token transmission overhead is considered as $|L| = |L_1|+|L_2|+|L_3| = 116+48+48 = 212$ B.

On successful interactive registration, U_i holds a smart card as $SC_i = (y_i, T_i, VPW, ID_i, CID_i, \Gamma_i)$ where y_i is a 64 B ciphertext, T_i is 20 B trapdoor, and Γ_i is 16 B featured value. Thus, the SC_i requires approximately 150 B to store essential security tokens.

Although the computation cost in our 3FAKA, as shown in Table 3, is high compared to some of the others, the growth is nominal for real-time applications. Besides, Fig. 3 depicts the comparisons of transmission and storage costs between the schemes. Despite W5 and W6 incurring lesser overheads, they fail to achieve comprehensive security traits as mentioned in Table 2. Hence, the suggested 3FAKA is an effective alternative for achieving all the F1-F6 and A1-A6 security traits with adequate overheads.

6 CONCLUSION AND FUTURE RESEARCH DIRECTION

This paper provides a novel secure three-factor authentication and key agreement based on extended chaotic maps and physical unclonable functions. It accomplishes several security features, including mutual authentication, session key agreement, dynamic device addition, user anonymity, efficient revocation, user untraceability, and remote smartcard revocation. Under the CMCDHP assumption, it is immune to various attacks, including device cloning, traceability, desynchronization, stolen data and passwords, replay, man-in-the-middle, and stolen smartcard attacks. Thus, the suggested protocol provides ample operational security while considering adequate computation, storage, and transmission costs.

Although it offers a broad set of features, the used fuzzy extractor may require more storage and processing power. In the future, we will focus on secure multi-factor authentication minimizing these issues in a trust-less multi-server scenario.

ACKNOWLEDGEMENTS

This work was supported by the National Science and Technology Council under grant 111-2222-E-110 - 008 and by the CANSEC-LAB@NSYSU in Taiwan.

REFERENCES

Aman, M. N., Sikdar, B., Chua, K. C., and Ali, A. (2018). Low power data integrity in IoT systems. *IEEE Internet of Things Journal*, 5(4):3102–3113.

Banerjee, S., Odelu, V., Das, A. K., Chattopadhyay, S., and Park, Y. (2020). An efficient, anonymous and robust authentication scheme for smart home environments. *Sensors*, 20(4):1215.

Chatterjee, S., Roy, S., Das, A. K., Chattopadhyay, S., Kumar, N., and Vasilakos, A. V. (2018). Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment. *IEEE Trans Dependable Secure Comput*, 15(5):824–839.

Das, A. K., Wazid, M., Kumar, N., Khan, M. K., Choo, K.-K. R., and Park, Y. (2018). Design of secure and lightweight authentication protocol for wearable devices environment. *IEEE Journal of Biomedical and Health Informatics*, 22(4):1310–1322.

Fakroon, M., Gebali, F., and Mamun, M. (2021). Multifactor authentication scheme using physically unclonable functions. *Internet of Things*, 13:100343.

Hu, V. C., Ferraiolo, D., Kuhn, R., Friedman, A. R., Lang, A. J., Cogdell, M. M., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K., et al. (2013). Guide to attribute based access control (ABAC) definition and considerations (draft). *NIST special publication*, 800(162):1–54.

Islam, S. (2014). Provably secure dynamic identity-based three-factor password authentication scheme using extended chaotic maps. *Nonlinear Dyn.*, 78(3):2261–2276.

Jiang, Q., Qian, Y., Ma, J., Ma, X., Cheng, Q., and Wei, F. (2019). User centric three-factor authentication protocol for cloud-assisted wearable devices. *Int. J. Commun. Syst.*, 32(6):e3900.

Jiang, Q., Wei, F., Fu, S., Ma, J., Li, G., and Alelaiwi, A. (2016). Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy. *Nonlinear Dyn.*, 83(4):2085–2101.

Jiang, Q., Zhang, N., Ni, J., Ma, J., Ma, X., and Choo, K.-K. R. (2020). Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles. *IEEE Trans. on Vehicular Technology*, 69(9):9390–9401.

Juels, A. and Ristenpart, T. (2014). Honey encryption: Security beyond the brute-force bound. In *Advances in Cryptology–EUROCRYPT 2014: 33rd Annual Intl. Conf. on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings 33*, pages 293–310. Springer.

Karati, A., Fan, C.-I., and Zhuang, E.-S. (2021). Reliable data sharing by certificateless encryption supporting keyword search against vulnerable KGC in industrial internet of things. *IEEE Trans. on Industrial Informatics*, 18(6):3661–3669.

Kirkpatrick, M. S., Kerr, S., and Bertino, E. (2014). System on chip and method for cryptography using a physically unclonable function. US Patent 8,750,502.

Kwon, D., Park, Y., and Park, Y. (2021). Provably secure three- factor-based mutual authentication scheme with PUF for wireless medical sensor networks. *Sensors*, 21(18):6039.

Masud, M., Gaba, G. S., Choudhary, K., Hossain, M. S., Alhamid, M. F., and Muhammad, G. (2021). Lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare. *IEEE Internet of Things Journal*, 9(4):2649–2656.

Qiu, S., Wang, D., Xu, G., and Kumari, S. (2022). Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices. *IEEE Trans Dependable Secure Comput*, 19(2):1338–1351.

Roy, S., Chatterjee, S., Das, A. K., Chattopadhyay, S., Kumari, S., and Jo, M. (2018). Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing internet of things. *IEEE IOT Journal*, 5(4):2884–2895.

Roy, S., Das, D., Mondal, A., Mahalat, M. H., Roy, S., and Sen, B. (2021). PUF based lightweight authentication and key exchange protocol for IoT. In *SECRYPT*, pages 698–703.

Rührmair, U., Sehnke, F., Sölter, J., Dror, G., Devadas, S., and Schmidhuber, J. (2010). Modeling attacks on physical unclonable functions. In *Proceedings of the 17th ACM conference on Computer and Communications Security (ACM-CCS)*, pages 237–249.

Ryu, J., Kang, D., and Won, D. (2022). Improved secure and efficient chebyshev chaotic map-based user authentication scheme. *IEEE Access*, 10:15891–15910.

Saqib, M., Jasra, B., and Moon, A. H. (2022). A lightweight three factor authentication framework for iot based critical applications. *Journal of King Saud University-Computer and Info. Sciences*, 34(9):6925–6937.

Trivedi, H. S. and Patel, S. J. (2021). Privacy preserving scalable authentication protocol with partially trusted third party for distributed internet-of-things. In *SECRYPT*, pages 812–818.

Wang, F., Xu, G., Xu, G., Wang, Y., and Peng, J. (2020). A robust IoT-based three-factor authentication scheme for cloud computing resistant to session key exposure. *Wirel Commun Mob Comput*, 2020.

Wang, W., Han, Z., Alazab, M., Gadekallu, T. R., Zhou, X., and Su, C. (2022). Ultra super fast authentication protocol for electric vehicle charging using extended chaotic maps. *IEEE Trans. on Industry Applications*, 58(5):5616–5623.

Wang, Z., Deng, D., Hou, S., Guo, Y., and Li, S. (2023). Design of three-factor secure and efficient authentication and key-sharing protocol for IoT devices. *Computer Communications*.

Yu, Y., Taylor, O., Li, R., and Sunagawa, B. (2021). An extended chaotic map-based authentication and key agreement scheme for multi-server environment. *Mathematics*, 9(8):798.

Zhang, L. (2008). Cryptanalysis of the public key encryption based on multiple chaotic systems. *Chaos, Solitons & Fractals*, 37(3):669–674.

Zhang, Y., Li, B., Wu, J., Liu, B., Chen, R., and Chang, J. (2022). Efficient and privacy-preserving blockchain-based multifactor device authentication protocol for cross-domain IIoT. *IEEE IOT J.*, 9(22):22501–22515.

Zhou, L., Li, X., Yeh, K.-H., Su, C., and Chiu, W. (2019). Lightweight IoT-based authentication scheme in cloud computing circumstance. *Future Generation Computer Systems*, 91:244–251.