Robust Multi-Factor Authentication for WSNs With Dynamic Password Recovery

Liufu Zhu and Ding Wang

***Abstract*— Multi-factor authentication (MFA) is crucial for Wireless Sensor Networks (WSNs) to ensure secure communication in security-critical applications such as smart homes, industrial control, and military defense due to the open nature of WSNs. Considerable efforts have been made to propose various MFA schemes with varied security goals and desirable properties. However, little attention has been given to the property of dynamic password recovery, and it still remains a question of *how to construct a robust MFA scheme with the desirable property of dynamic password recovery for WSNs*. In this paper, we first review two representative multi-factor authentication schemes proposed by Li-Tian (at IEEE Syst J'22) and Fatima et al. (at ACM TOSN'23) as case studies, and reveal that these two schemes fail to resist some known attacks and pay little attention to password forgetting and leakage issues. Accordingly, we employ the techniques of the honeywords method, fuzzy-verifier technique, and public key cryptosystem to construct a novel MFA scheme. Particularly, we propose the first dynamic password recovery method for MFA to address password forgetting and leakage issues. Key rotation is implemented to ensure the security of the long-term secret key. Our scheme is provably secure under the Random Oracle Model. Comparison results show the superiority of our new scheme.**

***Index Terms*— Multi-factor authentication, password recovery, provable security, random oracle model.**

I. Introduction

THE advancement of sensor devices and wireless communication has led to the rapid evolution of Wireless Sensor Networks (WSNs), which have been extensively implemented in various fields, such as smart homes, industrial control, and military defense. However, the open nature and resource-limited property of WSNs may pose serious security issues [1]. Therefore, it is urgent to take particular security measures to avoid these risks. Identity authentication serves as the first line of defense to protect communication and has been widely deployed. Usually, three types of information are treated as authentication factors: user-known information like password; user-unique biometrics like fingerprints; and user-owned physical information like smart card. Multi-factor authentication (MFA) schemes [2], [3], [4], [5], [6], [7], [8] utilize two or more factors to verify users to prevent security threats in case one or more authentication factors (not all of them) have been compromised.

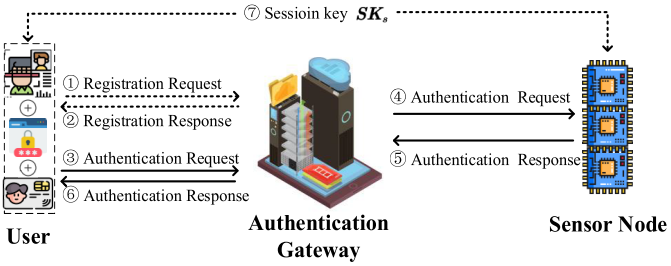

Fig. 1. Multi-factor authentication for WSNs. The serial number shows the order of message transmission. The dashed line represents the private channel, and the solid line represents the public channel.

Generally, there are three entities involved in a MFA scheme, namely user U_i, the authentication gateway AGW, and sensor node SN_j. As shown in Fig. 1, the system works as follows. U_i first sends a registration request to AGW. Upon receiving this request, AGW generates a registration response and sends it to U_i secretly. When U_i attempts to establish a session key with SN_j, she resubmits her password PW_i, smart card SC, and biometrics Bio_i and then generates an authentication request to AGW. If U_i is legitimate, AGW will transmit the request to SN_j. When mutual authentication is conducted successfully, a session key will be exchanged between U_i and SN_j.

Since Lamport [9] proposed the first authentication scheme in 1981, considerable efforts have been made to propose various MFA schemes with varied security goals and desirable properties for WSNs over the last 30 years. The relationships between the investigated MFA schemes are shown in Fig. 2. In 2009, Das [10] proposed a multi-factor authentication scheme for WSNs, requiring passwords and mobile devices to access the remote server. However, He et al. [11] pointed out that Das's scheme [10] is vulnerable to insider attacks and impersonation attacks. To address the above issues, He et al. [11] proposed an enhanced multi-factor authentication scheme for WSNs. However, Gope et al. [12] pointed out that He et al.'s scheme [11] is susceptible to privacy leakage and fails to provide anonymity and mutual authentication.

Received 21 February 2024; revised 5 June 2024 and 6 August 2024; accepted 12 August 2024. Date of publication 28 August 2024; date of current version 23 September 2024. This work was supported in part by the Natural Science Foundation of Tianjin, China, under Grant 21JCZXJC00100; in part by the National Natural Science Foundation of China under Grant 62222208; and in part by the Key Laboratory of Network Cryptography Technology under Grant LNCT2022-A02. The associate editor coordinating the review of this article and approving it for publication was Dr. Hossein Pishro-Nik. *(Corresponding author: Ding Wang.)*

The authors are with the College of Cyber Science, Nankai University, Tianjin 300350, China, also with the with Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450001, China, also with the Tianjin Key Laboratory of Network and Data Security Technology, Nankai University, Tianjin 300350, China, and also with the Key Laboratory of Data and Intelligent System Security (NKU), Ministry of Education, Tianjin 300350, China (e-mail: zhuliufu@mail.nankai.edu.cn; wangding@nankai.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIFS.2024.3451364, provided by the authors.

Digital Object Identifier 10.1109/TIFS.2024.3451364

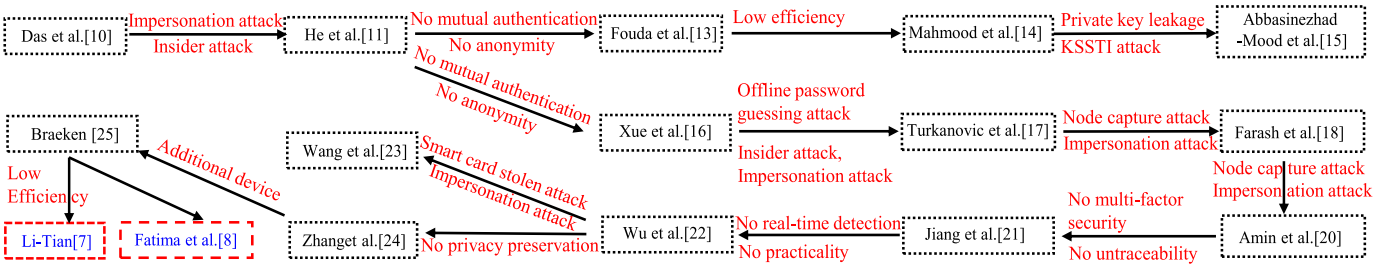

Fig. 2. The relationships between the investigated MFA schemes. A scheme associated with an arrow-tail means that it is prone to some defects as revealed around the arrow, while a scheme associated with an arrow-head means that it overcomes the related defects.

To ensure mutual authentication between users and sensor nodes, on the one hand, Fouda et al. [13] introduced a lightweight authentication scheme with key agreement and mutual authentication. However, Mahmood et al. [14] reported that Fouda et al.'s scheme [13] requires significant computational overhead because of the exponential-time complexity of some operations. To reduce the computational cost, Mahmood et al. [14] proposed a lightweight authentication scheme based on elliptic curve cryptography, achieving mutual authentication with low computation cost and resisting known attacks. However, Abbasinezhad-Mood and Nikooghadam [15] pointed out that Mahmood et al.'s scheme [14] is vulnerable to private key leakage and known session-specific temporary information attacks, and fails to provide perfect forward security.

On the other hand, Xue et al. [16] introduced a multi-factor authentication scheme for WSNs based on biometrics. However, Turkanovic et al. [17] pointed out that Xue et al.'s scheme [16] is vulnerable to offline password guessing attacks, privileged insider attacks, and impersonation attacks. As a repair, Turkanovic et al. [17] proposed a multi-factor authentication and key agreement scheme for WSNs, enabling remote users to negotiate the session key with common sensor nodes securely. Despite this improvement, Farash et al. [18], and Amin and Biswas [19] reported that Turkanovic et al.'s scheme [17] is unable to resist node capture attacks, impersonation attacks, and smart card stolen attacks. To tackle the aforementioned problems, Farash et al. [18] proposed an improved authentication scheme that enables users to communicate with any specific sensor node.

However, Amin et al. [20] found that Farash et al.'s scheme [18] is vulnerable to node capture attacks, offline password guessing attacks, impersonation attacks, and fails to achieve anonymity. As a remedy, Amin et al. [20] proposed an anonymity-preserving multi-factor authentication scheme with identity change and smart card revocation. However, Jiang et al. [21] pointed out that Amin et al.'s scheme [20] fails to provide true multi-factor security and cannot ensure untraceability. In order to address these issues, Jiang et al. [21] proposed an untraceable multi-factor authentication scheme using temporal-credential based on elliptic curve cryptography. But Wu et al. [22] pointed out some weaknesses in Jiang et al.'s scheme [21], such as the lack of user-friendliness, and no real-time detection for unauthorized logins. To solve these problems, Wu et al. [22] introduced an authentication scheme for WSNs, but it is vulnerable to smart card loss attacks, user impersonation attacks by Wang et al. [23].

Since most WSNs systems require users to register with private information, such as ID numbers, phone numbers, or biometrics, it is urgent and important to take privacy preservation seriously. To address this issue, Zhang et al. [24] proposed a blockchain-based authentication scheme for cross-domain WSNs with privacy preservation, but it requires an additional device to accumulate derived key materials. To address this issue, Braeken [25] proposed an anonymous authentication scheme using physically unclonable functions to ensure hardware security, but it involves complex computations to implement physically unclonable functions. To reduce the computation cost, Li and Tian [7] proposed a comprehensive efficient scheme that enables adaptive privacy protection. Recently, Fatima et al. [8] proceeded to propose a lightweight scheme with diverse security to support privacy preservation. However, neither Li-Tian [7] nor Fatima et al. [8] addresses password forgetting and leakage issues.

As known information, passwords are easy to deploy and change, making them widely applied to construct authentication schemes. However, due to the limited memory, users may only remember 5-7 different passwords [26]. If multiple passwords are set simultaneously, it may result in password forgetting or leakage issues. As far as we know, most schemes [2], [3], [4], [5], [7], [8], [27], [28] assume that passwords will not be forgotten or leaked, and they require users to resubmit previous passwords to conduct authentication and perform password updating. Therefore, these representative schemes fail to be deployed successfully in the password forgetting and leakage scenarios where password recovery is required..

To address the aforementioned issues, Ellison et al. [29] constructed a password recovery scheme based on the encryption algorithm. The password is encrypted locally and can be decrypted by users who own the secret key, which can be reconstructed through the correct answers to private questions. However, it fails to present a security analysis. To provide a formal security analysis, Frykholm and Juels [30] introduced a secure password recovery scheme with fault-tolerant code, where answers to private questions are formalized as low entropy secrets to recover passwords. However, it requires users to store all possible secret subsets locally.

The aforementioned schemes achieve password recovery through the application of private questions, yet fail to consider the security guaranteed by these questions. To tackle this problem, Just and Aspinall [31] evaluated the security of private questions for authentication and found that many user-selected questions have a low entropy, making them susceptible

to guessing attacks. Furthermore, Bonneau et al. [32] and Golla and Durmuth [33] revealed a number of weaknesses in private questions applications through extensive analysis of large-scale and real-world datasets and then provided crucial empirical evidence and recommendations for designing robust recovery methods using private questions securely.

In addition, remembering all correct answers may lead to an expensive memory burden. To address this issue, Dennig et al. [34] introduced a new method that applies implicit memory techniques based on users' behavior and emotion to minimize the memory burden. Moreover, Kharudin et al. [35] proposed a graphic-based password recovery method that applies graphical and visual elements to help users recover passwords. These approaches not only reduce the memory load but also protect password security due to the fact that implicit factors are more difficult to guess than private questions. However, these methods require additional devices to extract and store secret behavior information.

To make recovery methods more simple and practical, email is commonly employed to help users retrieve their passwords. However, its security issues have been insufficiently researched. In 2010, Jin et al. [36] analyzed the vulnerabilities of password recovery methods based on email, pointing out that they are vulnerable to security risks like phishing attacks. Furthermore, Al Maqbali and Mitchell [37] conducted a comprehensive analysis of email-based password recovery systems, emphasizing that reliance on email for password recovery still presents significant flaws. Additionally, Li et al. [38] found that numerous websites employ registered email accounts for password recovery and claimed that it is a convenient yet insecure method. As an improvement, they introduced a new protocol that enhances security by incorporating an additional verification phase.

The above password recovery methods cannot be directly deployed in MFA schemes. On the one hand, correct answers to private questions may be collected through phishing attacks during the interaction between users and servers, resulting in malicious recovery behavior on another system; On the other hand, when the server breaks down, users may not enjoy the recovery service promptly; even worse, building a password-related table on the server may lead to password guessing attacks. Therefore, further research is required to design dynamic password recovery methods in MFA schemes.

A. Motivations

Passwords are the most prevalent method in the construction of MFA schemes because of their convenient modification and deployment [39]. To prevent password reuse issues and password guessing attacks, more and more schemes require users to set different and complex passwords, which are composed of uppercase and lowercase letters, numbers, and special characters [40]. However, users may forget or leak passwords due to the limited memory [41]. Recently, Wang et al. [42], [43] have found that human-chosen passwords follow Zipf's law, which reveals that it will be easier than expected to perform password guessing attacks on human-chosen passwords. Although setting complex passwords may reduce the risk of password guessing attacks [44], [45], it may lead to inconvenience in remembering these complex passwords and make it easier for users to forget. Therefore, it is necessary to address password forgetting and leakage issues and construct MFA schemes with dynamic password recovery.

Furthermore, as shown in Fig. 2, the efforts in constructing MFA schemes have fallen into the "*break-fix-break-fix*" circle. Precedent schemes claim to be secure but later all have also been pointed out by successive schemes to be vulnerable to severe security flaws, such as password guessing attacks, impersonation attacks, and node capture attacks. Particularly, many existing schemes fail to achieve multi-factor security when two out of three (one out of two, etc.) authentication factors are compromised. To fairly and scientifically evaluate MFA schemes, considerable efforts have been devoted to addressing this issue. In general, a multi-factor authentication scheme needs to satisfy the following security goals [23], [46], [47], [48], [49], [50]: Multi-factor security; Resistance against password guessing attacks; Anonymity and untraceability; Security against known attacks; and Session key security. However, these existing evaluation criteria fail to take password forgetting and leakage issues into consideration. Above all, it is necessary and urgent to address the question *"How to construct a robust MFA scheme with the desirable property of dynamic password recovery for WSNs?"*.

B. Contributions

This paper will review two representative multi-factor authentication schemes proposed by Li and Tian [7] and Fatima et al. [8]. Cryptoanalysis of these schemes shows that they fail to resist some known attacks. Furthermore, we propose a robust MFA scheme with dynamic password recovery that meets Wang-Wang's evaluation criteria [48]. The key contributions are outlined as follows:

(1) We investigate two representative MFA schemes (Li and Tian [7] and Fatima et al. [8]) and show that they fail to achieve some security goals, such as securing against password guessing attacks and impersonation attacks. Particularly, these two schemes also fail to provide dynamic password recovery and key rotation.

(2) We propose a robust Multi-Factor Authentication scheme for WSNs with Dynamic Password Recovery and Update (MFA-DPRU). To address password forgetting and leakage issues, a dynamic password recovery method based on personal private questions and the secret sharing technique is designed to assist users in finding their passwords and then permits them to update passwords, private questions and secret values locally. To provide perfect forward security, key rotation is provided to ensure the security of the long-term secret key. To prevent password guessing attacks, we employ the fuzzy verifier technique that can mix up the distribution of passwords and help to resist offline password guessing attacks; To resist online password guessing attacks, the honeywords method is employed to provide real-time detection of malicious login issues.

(3) We prove the security of MFA-DPRU under the random oracle model and further provide a heuristic analysis to

show that MFA-DPRU can resist known attacks. We further apply automated tools Tamarin to perform security analysis and show that MFA-DPRU is secure against password guessing attacks and has forward security and session key security. The comparison results to some state-of-the-art works demonstrate that MFA-DPRU is superior in security and efficiency.

II. Preliminaries

We introduce some necessary preliminaries, including fuzzy extractor, adversary model, and system evaluation criteria.

A. Fuzzy Extractor

The fuzzy extractor of biometrics [51] has two algorithms, $Gen(\cdot)$ and $Rep(\cdot)$. The definitions are presented as follows.

(1) $\boldsymbol{Gen(Bio)=(\sigma,\tau)}$. $Gen(\cdot)$ is a probabilistic algorithm, that is, given biometrics Bio, it will output $(\sigma,\ \tau)$, where σ is a secret biometric key extracted from Bio and τ is an auxiliary element related to σ.

(2) $\boldsymbol{Rep(Bio',\tau)=\sigma}$. $Rep(\cdot)$ is a deterministic algorithm, that is, given biometrics and auxiliary element $(Bio',\ \tau)$, it will output the σ related to Bio'.

B. Adversary Model

Referring to the previous work [23], [47], [48], [50], the comprehensive adversary models are described as follows.

Adv1: An adversary can offline access all elements in identity and password space in polynomial time and then enumerate all pairs $(PW_i,\ ID_i)$. This is reasonable because the space sizes satisfy $|ID_i| \leq |PW_i| \leq 10^6$ and the human-chosen passwords have low entropy.

Adv2: An adversary can control the communications channels. For example, she can intercept, modify, eavesdrop, and replay any message transmitted in a public manner.

Adv3: An adversary can not only retrieve all secret information stored in the smart card but also obtain secret value about biometrics through malicious devices. Generally speaking, it can compromise any two of the three authentication factors.

Adv4: An adversary can compromise sensor nodes and extract secret information. It makes use of secret information to conduct sensor node impersonation attacks.

Adv5: An adversary can achieve the long-term secret key and attempt to rebuild the session key.

Adv6: An adversary can register and log in as a legitimate party, and attempt to extract other secret information.

C. System Evaluation Criteria

The community has made more and more remarkable contributions (see [48], [52], [53]) to establishing systemic and comprehensive evaluation criteria. Among them, Wang-Wang's evaluation criteria [48] is the most comprehensive and concrete, and it has been widely adopted (see [5], [6], [7], [8], [27], [49]). Here we introduce it and make some modifications (i.e., dynamic password recovery and no passwords and biometrics exposure) to meet the following requirements.

C1 **No password verifier table**. No password tables are generated by authentication gateway and sensor nodes.

C2 **Dynamic password recovery and updating**. Users can recover and update passwords dynamically even if they forget and leak their previous passwords.

C3 **No password and biometrics exposure**. The publicly transmitted messages reveal no valid information about passwords and biometrics.

C4 **Resistance to smart card loss attacks**. It is infeasible for adversary to perform other attacks or retrieve other secrets through a stolen smart card.

C5 **Resistance to known attacks**. The scheme is secure against various attacks, like node capture attacks, replay attacks, password guessing attacks, impersonation attacks, insider attacks, known session-specific temporary information attacks, clone card attacks, and key compromised impersonation attacks.

C6 **Sound flexibility**. It is feasible to revoke accounts for users when they want to exit the session.

C7 **Key-agreement provision**. A session key is shared between users and sensor nodes.

C8 **No clock synchronization**. The proposed scheme should keep clock asynchronous.

C9 **Timely detection**. A timely report is necessary when false passwords are entered.

C10 **Mutual-authentication**. Mutual authentication should be performed among parties.

C11 **User anonymity and untraceability**. It should provide privacy preservation for all parties.

C12 **Forward-security**. The shared session key is secure even if the long-term secret key is compromised.

Note that, we have enhanced **C2** to employ dynamic password recovery, effectively addressing password forgetting and leakage issues. Moreover, **C3** has been modified to ensure that publicly transmitted messages will not reveal any valid information about passwords and biometrics, thereby reinforcing multi-factor security and privacy preservation.

D. Security Model

Let Ω_U^i denotes the user U_i, Ω_S^j denotes sensor node SN_j, and Ω_{AGW}^k denotes the authentication gateway AGW. Suppose the existence of an adversary $\mathcal{A}$, then $\mathcal{A}$ will use $\Delta = \{\Omega_U^i,\ \Omega_S^j,\ \Omega_{AGW}^k\}$ to perform the following oracle queries and try to retrieve the secret key.

(1) $Execute(\Delta)$. This oracle enables examination of all passive eavesdropping behaviors performed by $\mathcal{A}$ and then outputs each session transcripts during the login and authentication phase from Ω_U^i, Ω_S^j, and Ω_{AGW}^k.

(2) $Send(\Omega_U^i,\ Start)$. This query is defined to be the initialization phase of the proposed scheme.

(3) $Send(\Delta,\ M)$. This oracle can capture active attacks, that is, $\mathcal{A}$ can block and intercept messages transmitted in public and then forge M which will be sent to Δ as a challenge. Then, $\mathcal{A}$ could receive a response from Δ.

(4) $Corrupt(\Delta,\ b)$. According to the systematic evaluation criteria introduced above, $\mathcal{A}$ is allowed to obtain any

TABLE I
SYMBOL AND NOTATION

Symbol	Notation	Symbol	Notation
U_i	i^{th} User entity	AGW	Authentication Gateway
SN_j	j^{th} Sensor Node	ID_i	Identity of U_i
PW_i	Password of U_i	Bio_i	Biometrics of U_i
σ_i	Biometrics secret key	θ_i	Biometrics auxiliary information
r_u	Random value chosen by U_i	K_{gu}	Temporary key between U_i and AGW
Y_i	A nonce chosen by U_i	AGW	Authentication Gateway
SID_j	Identity of SN_j	r_s	Random value chosen by SN_j
p	A large prime	$Honey\_list$	Honeyword list
$\oplus$	Bitwise XOR operation	$\|\|$	String concatenation operation
Y_j	A nonce chosen by SN_j	K_{gs}	Temporary key between SN_j and AGW
x	Long term secret key of AGW	r_g	Random value chosen by AGW
SK_s	Session key shared between U_i and SN_j	$Gen(\cdot)$	Extraction function of fuzzy extractor
$Rep(\cdot)$	Reconstruction function of fuzzy extractor	$H(\cdot)$	Collision resistant hash function

two of the three authentication factors such as (SC, PW_i), (SC, Bio_i), and (PW_i, Bio_i), except for all of them. There exist four cases in this oracle query:

(a) (Ω_U^i, 1). It denotes that $\mathcal{A}$ achieves the password PW_i and smart card SC_i that belong to U_i.
(b) (Ω_U^i, 2). It denotes that $\mathcal{A}$ obtains the biometrics Bio_i and smart card SC_i that belong to U_i.
(c) (Ω_U^i, 3). This shows that $\mathcal{A}$ captures the password PW_i and biometrics Bio_i that belong to U_i.
(d) (Ω_{AGW}^k, 1). In addition, the long-term secret key x of AGW can be exposed to $\mathcal{A}$.

(5) $Hash(\cdot)$. This oracle will output a random hash value as a response when a string is input by $\mathcal{A}$ as a query. During this query, $\mathcal{A}$ will attempt to perform private questions guessing attacks to get the correct answers.
(6) $Test(\Delta)$. If $\mathcal{A}$ performs this query, first of all, a coin b will be flipped. If $b = 0$, a randomly chosen session key SK_s' will be sent to $\mathcal{A}$. If $b = 1$, then the corrrect session key SK_s is returned to $\mathcal{A}$. It is computationally indistinguishable between the random SK_s' and the real SK_s for $\mathcal{A}$ in polynomial time.

Definition 1: If there exist an adversary $\mathcal{A}$ who is able to $\mathcal{A}$ can perform $Execute(\Delta)$, $Send(\Omega_U^i, Start)$, $Send(\Delta, M)$, $Corrupt(\Delta, b)$, $Hash(\cdot)$, and $Test(\Delta)$ queries in polynomial time under the ROM model and she can win the above interaction game with non-negligible advantage $Adv_{\mathcal{A}}^{Suc}$, then she can break the proposed scheme efficiently with non-negligible advantage.

III. REVIEW OF LI-TIAN'S SCHEME

In this section, we briefly review Li-Tian's scheme [7], which includes Sensor Node Registration, User Registration, Login and Authentication, Password and Biometrics renewal, and Revocation Phase. Table I shows commonly used symbols and their notations in this paper.

A. Sensor Node Registration Phase

When AGW receives a registration request from SN_j, it chooses a unique identity SID_j, a random number y_j, and then computes $K_{gs} = SID_j \oplus y_j$ as the preshared session key. Finally, AGW transmits (SID_j, K_{gs}) to SN secretly.

B. User Registration Phase

1) UR1: U_i randomly selects a unique identity string ID_i, a password PW_i, and extracts biometrics Bio_i through specific devices. She proceeds to choose y_i randomly and computes $Gen(Bio_i) = (\sigma_i, \theta_i)$, $HPW_i = h\ (PW_i \ || \ \sigma_i)$. Finally, $\{ID_i, HPW_i, y_i\}$ will be sent to AGW via a secure channel.

2) UR2: Upon the receipt of registration request, AGW selects a random nonce SCN_i and then calculates $C_1 = H(ID_i \ || \ HPW_i)$, $PID_i = SCN_i \oplus ID_i \oplus HPW_i$, $K_{gu} = h(x \ || \ ID_i \ || \ y_i)$, and $C_2 = K_{gu} \oplus h(HPW_i \ || \ ID_i)$, where x represents the long-term secret key of AGW, K_{gu} denotes the preshared session key between AGW and U_i. At last, AGW generates the smart card SC_i and injects $\{C_1, C_2, PID_i\}$ into it, and sends SC_i to U_i secretly.

3) UR3: After obtaining SC_i from AGW, U_i further computes $SCN_i = PID_i \oplus ID_i \oplus HPW_i$, and then injects $\{SCN_i, \theta_i, Gen(\cdot), Rep(\cdot)\}$ into SC_i.

C. Login and Authentication Phase

1) UL1: U_i inserts SC_i into the specified device and inputs ID_i^*, PW_i^*, and Bio_i^*. The device calculates $\sigma_i^* = Rep(Bio_i^*, \theta_i)$, $HPW_i^* = h(PW_i^*, \sigma_i^*)$, and $C_1^* = h(ID_i^* \ || \ HPW_i^*)$. If the equation $C_1^* = C_1$ holds, then it computes $K_{gu} = C_2 \oplus h(HPW_i^* \ || \ ID_i^*)$, and $PID_i^* = SCN_i \oplus ID_i^* \oplus HPW_i^*$ for U_i. Furthermore, U_i selects a random number PID_i^{new} and computes $r_u = PID_i^{new} \oplus PID_i^*$, $M_1 = ID_i^* \oplus h(PID_i^* \ || \ K_{gu})$, $M_2 = r_u \oplus h(K_{gu} \ || \ ID_i^* \ || \ T_1)$, $M_3 = SID_j \oplus h(K_{gu} \ || \ r_u \ || \ PID_i^* \ || \ T_1)$, and $M_4 = h(ID_i^* \ || \ SID_j \ || \ r_u \ || \ PID_i^* \ || \ T_1)$, where T_1 represents the current timestamp. Finally, U_i transmits $\{M_1, M_2, M_3, M_4, PID_i^*, T_1\}$ to AGW publicly.

2) UL2: Once receiving authentication request from U_i, AGW checks the freshness of T_1, where $|T_1' - T_1| \leq \Delta T$. It further computes $K_{gu} = h(x \parallel ID_i \parallel y_i)$ and $ID_i^* = M_1{}^* \oplus h(PID_i^* \parallel K_{gu})$. If $ID_I^* = ID_i$, AGW continues to compute $r_u = M_2^* \oplus h(K_{gu} \ || \ ID_i \ || \ T_1)$, $SID_j = M_3^* \oplus h(K_{gu} \ || \ r_u \ || \ PID_i^* \ || \ T_1)$, and $M_4' = h(ID_i \ || \ SID_j \ || \ r_u \ || \ PID_i^* \ || \ T_1)$. If $M_4' = M_4^*$, AGW successfully authenticates U_i, and then computes $K_{gs} = SID_j \oplus y_i$, $M_5 = r_u \oplus h(K_{gs} \ || \ SID_j \ || \ T_2)$, $M_6 = r_g \oplus h(r_u \ || \ K_{gs} \ || \ T_2)$, and $M_7 = h(SID_j \ || \ r_u \ || \ PID_i \ || \ r_g || \ T_2)$, where T_2 denotes the timestamp. AGW transmits $\{M_5, M_6, M_7, T_2\}$ to SN_j in public manner.

3) UL3: Upon the receipt of $\{M_5, M_6, M_7, T_2'\}$, SN_j checks whether $|T_2' - T_2| \leq \Delta T$. Further, SN_j calculates $r_u = M_5^* \oplus h(K_{gs} \ || \ SID_j \ || \ T_2)$, $r_g = M_6^* \oplus h(r_u \ || \ K_{gs} \ || \ T_2)$ and then examines whether the equation $M_7^* = h(SID_j \ || \ r_u \ || \ PID_i^* \ || \ r_g \ || \ T_2)$ hold or not. If the above equation holds, then SN_j continues to compute $SK_s = h((PID_i^* \oplus SID_j) \ || \ r_u \ || \ r_g \ || \ r_s)$, $M_8 = r_s \oplus h(r_u \ || \ r_g)$, $M_9 = h(SID_j \ || \ r_g \ || \ SK_s \ || \ r_s \ || \ T_3)$, and $K_{gs}^{new} = K_{gs} \oplus r_g$, where r_s is chosen randomly by SN_j, T_3 represents the current timestamp. Finally, SN_j returns $\{M_8, M_9, T_3\}$ back as a response to AGW.

4) UL4: After receving $\{M_8, M_9, T_3'\}$, AGW checks the freshness of T_3, where $|T_3' - T_3| \leq \Delta\ T$. AGW calculates $r_s = M_8^* \oplus h(r_u \ || \ r_g)$, $SK_s = h((PID_i^* \oplus SID_j) \ || \ r_u \ || \ r_g \ || \ r_s)$, and $M_9' = h(SID_j \ || \ r_g \ || \ SK_s \ || \ r_s \ || \ T_s)$. If the equation

$M'_9 = M_9$ holds, AGW computes $PID_i^{new} = PID_i \oplus r_u$, $y_j^{new} = y_i \oplus r_g$, $M_{10}^* = r_g \oplus h(PID_i^{new} \| K_{gu} \| T_4)$, and $M_{11}^* = h(PID_i^{new} \| K_{gu} \| T_4)$, where T_3 denotes the timestamp. Finally, AGW transmits $\{M_8, M_{10}, M_{11}, T_4\}$ to U_i as a authentication response publicly.

5) UL5: U_i receives $\{M_8, M_{10}, M_{11}, T'_4\}$ and checks the freshness of T_4, that is, $|T'_4 - T_4| \leq \Delta T$. Furthermore, U_i computes $r_g = M_{10}^* \oplus h(PID_i^* \| K_{gu} \| T_4)$, $r_s = M_8^* \oplus h(r_u \| r_g)$, $SK_s = h((PID_i^* \oplus SID_j) \| r_u \| r_g \| r_s)$, and $M'_{11} = h(PID_i^{new} \| K_{gu} \| T_4)$. U_i checks whether the equation $M'_{11} = M_{11}$ holds. If it holds, SK_s is the correct session key shared between U_i and SN_j.

IV. Cryptanalysis of Li-Tian's Scheme

In this section, we conduct a cryptanalysis of Li-Tian's scheme [7]. A scheme with multi-factor security should remain secure even if any two out of three authentication factors are available to adversary $\mathcal{A}$. We analyze the security of Li-Tian's scheme [7] on the assumption that biometrics Bio_i and smart card SC_i are exposed to $\mathcal{A}$.

A. Offline/Online Password Guessing Attacks

Since Bio_i and SC_i have been exposed to $\mathcal{A}$, it can extract $\{C_1, C_2, PID_i, SCN_i, \theta_i, Gen(\cdot), Rep(\cdot)\}$ from SC_i. Further, $\mathcal{A}$ retrieves σ_i by computing $\sigma_i = Gen(Bio_i, \theta_i)$. $\mathcal{A}$ can launch offline password guessing attacks as follows:

Step1 $\mathcal{A}$ can adaptivly choose a pair of (ID'_i, PW'_i) from identity and password space $D_{id} \times D_{pw}$ every time.

Step2 Then, $\mathcal{A}$ proceeds to compute $HPW'_i = h(PW'_i, \sigma_i)$, and $C'_1 = h(ID'_i \| HPW_i^{*'})$ as described in the protocol.

Step3 Futhermore, $\mathcal{A}$ checks whether the equation $C'_1 = C_1$ holds or not. If the above equation holds, it implies that $\mathcal{A}$ guesses the target values (ID_i^*, PW_i^*) correctly.

Remarks. $\mathcal{A}$ can also conduct offline password guessing attacks according to the following steps.

Step1 $\mathcal{A}$ adaptively chooses a pair of (ID'_i, PW'_i) from identity and password space $D_{id} \times D_{pw}$ every time.

Step2 $\mathcal{A}$ further computes $HPW'_i = h(PW'_i, \sigma_i)$, and $PID'_i = SCN_i \oplus ID'_i \oplus HPW'_i$ following the scheme.

Step3 $\mathcal{A}$ compares PID'_i with PID_i extracted from the smart card SC_i. If the equation $PID'_i = PID_i$ holds, it shows that $\mathcal{A}$ guesses (ID_i^*, PW_i^*) correctly.

Since no mechanism is provided to prevent online password guessing attacks, then $\mathcal{A}$ can guess (ID_i^*, PW_i^*) as follows:

Step1 $\mathcal{A}$ adaptivly choose a pair of (ID'_i, PW'_i) as a guess from identity and password space $D_{id} \times D_{pw}$.

Step2 Then $\mathcal{A}$ continues to calculate $HPW'_i = h(PW'_i, \sigma_i)$, $C'_1 = h(ID'_i \| HPW'_i)$, $K'_{gu} = C_2 \oplus h(HPW'_i \| ID'_i)$, and $PID_i = SCN_i \oplus ID'_i \oplus HPW'_i$. Later, $\mathcal{A}$ chooses a random number PID_i^{new} and then computes $r'_u = PID_i^{new} \oplus PID_i$, $M'_1 = ID'_i \oplus h(PID_i \| K'_{gu})$, $M_2 = r_u \oplus h(K_{gu} \| ID'_i \| T_1)$, $M'_3 = SID_j \oplus h(K'_{gu} \| r'_u \| PID_i \| T_1)$, and $M'_4 = h(ID'_i \| SID_j \| r'_u \| PID_i \| T_1)$. $\mathcal{A}$ transmits $\{M'_1, M'_2, M'_3, M'_4, PID_i, T_1\}$ to AGW as a login request.

Step3 If the above session does not abort and a message will be returned from AGW as a response, it implies that $\mathcal{A}$ guesses (ID_i^*, PW_i^*) successfully.

B. User Impersonation Attacks

Suppose that, $\mathcal{A}$ has performed the offline/online password guessing attacks and obtained the correct (ID_i, PW_i). Consequently, $\mathcal{A}$ could perform user impersonation attacks to communicate with AGW and SN_j as a legitimate user.

Step1 $\mathcal{A}$ obtains the correct (ID_i, PW_i) by performing offline/online password guessing attacks. Furthernore, she computes $K_{gu} = C_2 \oplus h(HPW_i \| ID_i)$, then selects a random number PID_i^{new} and continues to compute $r'_u = PID_i^{new} \oplus PID_i$, $M_1 = ID_i \oplus h(PID_i \| K_{gu})$, $M'_2 = r'_u \oplus h(K_{gu} \| ID_i \| T_1)$, $M'_3 = SID_j \oplus h(K_{gu} \| r'_u \| PID_i \| T_1)$, and $M'_4 = h(ID_i{}^* \| SID_j \| r'_u \| PID_i \| T_1)$. Finally, $\mathcal{A}$ transmits $\{M_1, M'_2, M'_3, M'_4, PID_i, T_1\}$ to AGW as a login and authentication request.

Step2 When AGW receives the request $\{M_1, M'_2, M'_3, M'_4, PID_i, T_1\}$, it computes $K_{gu} = h(x \| ID_i \| y_i)$, $ID_i = M_1 \oplus h(PID_i \| K_{gu})$, $r'_u = M_2 \oplus h(K_{gu} \| ID_i \| T_1)$, $SID_j = M_3 \oplus h(K_{gu} \| r'_u \| PID_i \| T_1)$, and $M_4^* = h(ID_i \| SID_j \| r'_u \| PID_i \| T_1)$. The equation $M_4^* = M'_4$ shows that AGW authenticates $\mathcal{A}$ successfully.

Step3 Later, $\mathcal{A}$ obtains $\{M_8, M_{10}, M_{11}, T_4\}$ and computes $r_g = M_{10} \oplus h(PID_i \| K_{gu} \| T_4)$, $r_s = M_8 \oplus h(r'_u \| r_g)$. Finally, it enables to rebuild the session key $SK_s = h((PID_i \oplus SID_j) \| r'_u \| r_g \| r_s)$.

V. Review of Fatima Et Al.'s Scheme

In this section, we review Fatima et al's schme [8], which includes Initialization, Pre-deployment, User Registration, and Key Agreement phases.

A. Initialization Phase

At first, AGW selects a random number x as its long-term secret key. In addition, two large distinct prime numbers z, y will be chosen randomly as its top secrets, where $n = z \cdot y$. Finally, AGW makes n public and keeps (x, y, z) secret.

B. Pre-Deployment Phase

1) SP1: SN_j selects a unique identity ID_j, a random number Y_j, and then sends $\{ID_j, Y_j\}$ to AGW secretly.

2) SP2: Upon the receipt of $\{ID_j, Y_j\}$ from SN_j, AGW proceeds to compute $SID_j = h(ID_j \| Y_j)$, and $K_{gs} = h(SID_j \| \oplus Y_j)$, where K_{gs} denotes the preshared key. At last, AGW transmits SID_j to SN_j via a secure channel.

3) SP3: When obtaining SID_j, SN_j stores it in memory.

C. User Registration Phase

1) UR1: U_i chooses a unique identity ID_i, a password PW_i, and extracts biometrics Bio_i through specified device. The device proceeds to compute $Gen(Bio_i) = (\sigma_i, \theta_i)$, $MU_i = h(ID_i \| n_i)$, where $n_i \in Z_p^*$ is chosen randomly. At last, U_i sends MU_i to AGW as a registration request.

2) UR2: Upon the receipt of registration request MU_i, AGW selects a pesudonym PID_i, and then computes $Y_i = h(MU_i \| x)$. It continues to inject $\{MU_i, Y_i\}$ into the smart card SC_i, and sends SC_i to U_i via a secure channel.

3) UR3: When U_i achieves SC_i, she calculates $p_i = MU_i \oplus h(PID_i \,||\, PW_i \,||\, ID_i \,||\, Bio_i)$, and $q_i = h(ID_i \,||\, PW_i \,||\, PID_i \,||\, MU_i \,||\, \sigma_i)$. Finally, U_i keeps $\{PID_i, p_i, q_i\}$ secret and add them into SC_i.

D. Key Agreement Phase

1) KE1: U_i inserts SC_i into the specified device, and then inputs ID_i^*. PW_i^*, Bio_i^*. The device further computes $\sigma_i^* = Rep(Bio_i^*, \theta_i)$, $MU_i = p_i \oplus h(PID_i \,||\, PW_i^* \,||\, ID_i^* \,||\, Bio_i^*)$, $q_i^* = h(ID_i^* \,||\, PW_i^* \,||\, PID_i \,||\, MU_i \,||\, \sigma_i^*)$. Assume that U_i has known the identity SID_j, then she selects a random number $r_u \in Z_p^*$ and computes $MID_i = (MU_i \,||\, PDI_i \,||\, r_u)^2 \bmod n$, $F_1 = SID_j \oplus (MU_i \,||\, PID_i) \oplus r_u$, $F_2 = h(MU_i \,||\, Y_i \,||\, r_u \,||\, MID_i \,||\, SID_j)$. Finally, U_i sends $\{F_1, F_2, MID_i\}$ to AGW as login and authentication request publicly.

2) KE2: When AGW receives $\{F_1, F_2, MID_i\}$, it performs decryption to recover MU_i, PID_i, and r_u. It then computes $SID_j = F_1 \oplus (MU_i \,||\, PDI_i) \oplus r_u$, $Y_i = h(MU_i \,||\, x)$, and $F_2^* = h(MU_i \,||\, Y_i \,||\, r_u \,||\, MID_i \,||\, SID_j)$. It checks whether or not the equation $F_2^* = F_2$ holds. If it holds, AGW continues to calculate $K_{gs} = h(SID_j \,||\, Y_j)$, $K_j = h(K_{gs} \,||\, SID_j) \oplus r_g$, $F_3 = h(PID_i \,||\, SID_j \,||\, K_{gs} \,||\, F_2 \,||\, r_g)$, and $K_m = h(K_{gs} \,||\, F_3) \oplus r_u$, where r_g is chosen randomly. Finally, AGW sends $\{F_3, K_j, K_m\}$ to SN_j publicly.

3) KE3: Upon the receipt of $\{F_3, K_j, K_m\}$, SN_j computes $K_{gs} = h(SID_j \,||\, Y_j)$, $r_g = K_j \oplus h(K_{gs} \,||\, SID_j)$, $F_3' = h(PID_i \,||\, SID_j \,||\, K_{gs} \,||\, F_2 \,||\, r_g)$. It checks whether the equation $F_3' = F_3$ holds or not. If it holds, SN_j chooses a random number $r_s \in Z_p^*$ and calculates $r_u = h(K_{gs} \,||\, F_3) \oplus K_m$, $F_4 = r_s \oplus (SID_j \,||\, F_3)$, $SK_s = h(SID_j \,||\, PID_i \,||\, r_u \,||\, r_g \,||\, r_s \,||\, K_j \,||\, F_4)$, and $F_5 = h(K_j \,||\, SK_s \,||\, F4)$. Finally, it forwards $\{F_4, F_5\}$ to AGW through a public channel.

4) KE4: After receving $\{F_4, F_5\}$, AGW computes $r_s = F_4 \oplus (SID_j \,||\, F_3)$, $SK_s = h(SID_j \,||\, PID_i \,||\, r_u \,||\, r_g \,||\, r_s \,||\, K_j \,||\, F_4)$, and $F_5' = h(K_j \,||\, SK_s \,||\, F4)$. AGW examines whether $F_5' = F_5$ holds. If it holds, AGW computes $F_6 = r_g \oplus MU_i$, $F_7 = K_j \oplus F_2$, and $F_8 = h(SK_s \,||\, r_g \,||\, F_2)$. AGW transimits $\{F_4, F_6, F_7, F_8\}$ to U_i publicly.

5) KE5: Upon obtaining $\{F_4, F_6, F_7, F_8\}$, U_i computes $r_s = F_4 \oplus (SID_j \,||\, F_3)$, $r_g = F_6 \oplus MU_i$, $K_j = F_7 \oplus F_2$, and $SK_s = h(SID_j \,||\, PID_i \,||\, r_u \,||\, r_g \,||\, r_s \,||\, K_j \,||\, F_4)$. In the end, a session key is established between U_i and SN_j.

VI. Cryptanalysis of Fatima Et Al.'s Scheme

In this section, we analyze Fatima et al.'s three-factor scheme [8] and point out that it fails to resist password guessing attacks and impersonation attacks. Multi-factor security requires security even if any two out of three factors are compromised. Here, we suppose that biometrics Bio_i and smart card SC_i are exposed to $\mathcal{A}$.

A. Offline/Online Password Guessing Attacks

Since Bio_i and SC_i have been exposed to $\mathcal{A}$, she can extract message $\{MU_i, Y_i, PID_i, p_i, q_i, Gen(\cdot), Rep(\cdot), \theta_i\}$ from SC_i. Furthermore, $\mathcal{A}$ retrieves σ_i by computing $\sigma_i = Gen(Bio_i, \theta_i)$. $\mathcal{A}$ performs offline password guessing attacks according to the following steps:

Step1 $\mathcal{A}$ adaptively chooses a pair of (ID_i, PW_i) from identity and password space $D_{id} \times D_{pw}$ every time.

Step2 $\mathcal{A}$ calculates $MU_i' = p_i \oplus h(PID_i \,||\, PW_i^* \,||\, ID_i^* \,||\, Bio_i)$, and $q_i' = h(ID_i^* \,||\, PW_i^* \,||\, PID_i \,||\, MU_i' \,||\, \sigma_i)$.

Step3 At last, $\mathcal{A}$ proceeds to check whether the equation $q_i' = q_i$ holds or not. It obviously shows that $\mathcal{A}$ guesses (ID_i^*, PW_i^*) successfully when $q_i' = q_i$.

On the other hand, $\mathcal{A}$ is able to guess (ID_i^*, PW_i^*) by performing online password guessing attacks as follows:

Step1 $\mathcal{A}$ can compute $MU_i' = p_i \oplus h(PID_i \,||\, PW_i^* \,||\, ID_i^* \,||Bio_i)$ as normal, and chooses a random number r_u'.

Step2 $\mathcal{A}$ computes $MID_i' = (MU_i' \,||\, PDI_i \,||\, r_u')^2 \bmod n$, $F_1' = SID_j \oplus (MU_i' \,||\, PID_i) \oplus r_u'$, and $F_2' = h(MU_i' \,||\, Y_i \,||\, r_u' \,||\, MID_i' \,||\, SID_j)$. Afterwards, $\mathcal{A}$ sends (MID_i', F_1', F_2') to AGW as a login request.

Step3 If this session doesn't terminate and a response is returned to $\mathcal{A}$, then $\mathcal{A}$ guesses (ID_i^*, PW_i^*) correctly.

B. User Impersonation Attacks

Suppose that, $\mathcal{A}$ has performed the offline/online password guessing attacks and obtained (ID_i, PW_i) successfully. Then $\mathcal{A}$ proceeds to launch user impersonation attacks to communicate with AGW and SN_j according to the following steps.

Step1 $\mathcal{A}$ obtains (ID_i, PW_i) by performing password guessing attacks. Then, $\mathcal{A}$ chooses r_u' randomly and computes $MID_i' = (MU_i \,||\, PDI_i \,||\, r_u')^2 \bmod n$, $F_1' = SID_j \oplus (MU_i \,||\, PID_i) \oplus r_u'$, $F_2' = h(MU_i \,||\, Y_i \,||\, r_u' \,||\, MID_i' \,||\, SID_j)$. U_i sends $\{F_1', F_2', MID_i'\}$ to AGW via a public manner.

Step2 When AGW receives request $\{F_1', F_2', MID_i'\}$, it performs decryption to recover MU_i, PID_i, and r_u' as usual and then computes $SID_j = F_1 \oplus (MU_i \,||\, PDI_i') \oplus r_u'$, $Y_i = h(MU_i \,||\, x)$, $F_2^* = h(MU_i \,||\, Y_i \,||\, r_u' \,||\, MID_i' \,||\, SID_j)$. The equation $F_2^* = F_2'$ holds so that $\mathcal{A}$ can be authenticated as a legitimate user.

Step3 After $\mathcal{A}$ obtains $\{F_4, F_6, F_7, F_8\}$, she computes $r_s = F_4 \oplus SID_j$, $r_g = F_6 \oplus MU_i$, and $K_j = F_7 \oplus F_2'$. $\mathcal{A}$ rebuilds $SK_s = h(SID_j \,||\, PID_i \,||\, r_u \,||\, r_g \,||\, r_s \,||\, K_j \,||\, F_4)$.

VII. The Proposed Protocol

The following section introduces the proposed MFA scheme named MFA-DPRU, which is robust and suitable for WSNs. Notably, MFA-DPRU allows for dynamic password recovery and key rotation. The main advantages of this scheme are detailed as follows:

(1) The application of the fuzzy verifier technique and the honeywords method makes it harder to perform password guessing attacks efficiently.
(2) The proposed scheme enables dynamic password recovery when users forget and leak their passwords. It also provides key rotation for the long-term secret key.
(3) The analysis results show that the proposed scheme keeps a balance between security and efficiency.

The proposed scheme consists of seven phases: Initialization, Sensor Node Registration, User Registration, Login and Authentication, Dynamic Password Recovery, Password and Smart Card Updating, and Key Rotation. There are three

entities, named user U_i, authentication gateway AGW, and sensor node SN_j. If mutual authentication is performed successfully, a session key will be shared between U_i and SN_j.

A. Initialization Phase

AGW first randomly chooses $x \in Z_p^*$ and two large distinct primes $y_g, z_g \in Z_p^*$, where $n = z_g \cdot y_g$. It is required that x, y_g, and z_g are kept secret. Then, it chooses hash functions $H_1(\cdot) : \{0,1\}^* \rightarrow \{0,1\}^*$, $H_2(\cdot) : \{0,1\}^* \rightarrow Z_p^*$, $H_3(\cdot) : Z_p^* \rightarrow \{0,1\}^*$. Additionlly, a medium intergity n_p is defined as $2^4 \leq n_p \leq 2^8$, determining the size of $(ID,\ PW)$ space. An integrity n_a is also chosen to represent the size of the answer space. Then, AGW keeps (x, y_g, z_g) secret and makes $\{H_1(\cdot),\ H_2(\cdot),\ H_3(\cdot),\ n,\ n_p,\ n_a\}$ public.

B. Sensor Node Registration Phase

1) SR1: First of all, SN_j selects a unique identity SID_j and a random nonce Y_j. Then, it proceeds to transmit $\{SID_j,\ Y_j\}$ to AGW as a registration secretly.

2) SR2: Upon the receipt of registration request $\{SID_j,\ Y_j\}$, AGM computes a preshared session key $K_{gs} = H_1(SID_j \ ||\ Y_j) \oplus H_1(H_3(x) \ ||\ Y_j)$ and keeps $\{SID_j,\ Y_j,\ K_{gs}\}$ secretly. Finally, AGW forwards $\{SID_j,\ K_{gs}\}$ to SN_j via a secure channel.

3) SR3: Upon the receipt of $\{SID_j,\ K_{gs}\}$, SN_j stores these messages securely.

C. User Registration Phase

The user registration phase is shown in Fig. 3.

1) UR1: U_i selects identity ID_i, password PW_i, and extracts biometrics Bio_i. She chooses random numbers $z_u, y_u \in Z_p^*$ satisfying $N_u = y_u \cdot z_u$, where y_u and z_u are distinct large prime and kept secret. A pair of RSA cryptosystem key is generated as $pk_u = (e_u,\ N_u)$, $sk_u = (d_u,\ N_u)$. Then $\{ID_i,\ PW_i,\ Bio_i\}$ are input to the device. The device chooses a nonce $Y_i \in \{0,1\}^*$ randomly, and calculates $Gen(Bio_i) = (\sigma_i,\ \theta_i)$, $HPW_i = H_1(PW_i \ ||\ \sigma_i)$, $MID_i = H_1(ID_i \ ||\ r_i)$, where r_i is chosen randomly. U_i forwards $\{HPW_i,\ MID_i,\ Y_i,\ \theta_i\}$ to AGW as a registration request.

2) UR2: Upon the receipt of registration request, AGW calculates $C_1 = H_1(\theta_i) \oplus H_1(H_2(MID_i) \oplus H_2(HPW_i)\ mod\ n_p)$, $K_{gu} = H_1(MID_i \ ||\ Y_i) \oplus H_1(H_3(x) \ ||\ Y_i)$, and $C_2 = K_{gu} \oplus H_1(H_2(HPW_i) \oplus H_2(MID_i)\ mod\ n_p) \oplus H_1(PID_i)$, where PID_i is a random nonce. If it is the first registration request sent by U_i, AGW will set an empty table and append $\{MID_i,\ a_i,\ Honey\_List = NULL\}$ to it, where $a_i \in \{0,1\}^*$ is chosen randomly and $Honey\_List$ denotes the list of honeywords. Otherwise, it just needs to update the elements $\{a_i,\ Honey\_List\}$. Furthermore, AGW proceeds to compute $Upd_i = H_1(H_2(MID_i \oplus H_2(HPW_i)\ mod\ n_p))^{e_u}$. AGW injects $\{PID_i,\ C_1,\ C_1 \oplus a_i,\ C_2,\ Upd_i\}$ into the SC_i and then transmits SC_i to U_i in a secret manner.

3) UR3: Once obtaining SC_i from AGW, U_i proceeds to calculate $P_i = MID_i \oplus H_1(H_2(Bio_i) \oplus H_2(ID_i)\ mod\ n_p) \oplus H_1(PID_i)$ and then selects a random polynomial $f(x) = a_0 + a_1x_1 + a_2x_2^2 + \ldots + a_{N-1}x_{N-1}^{N-1}\ mod\ p$, where $a_i \in [0, p-1]$. U_i proceeds to compute $PWC = AES^{256}.Enc_{a_0}(PW_i)$ and sets N questions $\{Que_i\}_{1 \leq N}$, such as "*What is your previous phone number?*", "*Frequent flyer number?*", then she sets the answer to each question as Ans_i. Let $(x_i, \beta_i) = (x_i, f(x_i))$ and compute $\delta_i = \beta_i \oplus H_1(H_2(Ans_i) \| H_2(ID_i)\ \ mod\ n_a)$. One of these N questions are related to ID_i and Bio_i and can be set as $\delta_j = \beta_j \oplus H_1(ID_i || Bio_i)$. U_i injects $\{P_i,\ \theta_i,\ Gen(\cdot),\ Rep(\cdot),\ \{Que_i\}_{1 \leq N},\ PWC,\ \{\delta_i\}_{1 \leq i \leq N}\}$ into SC_i.

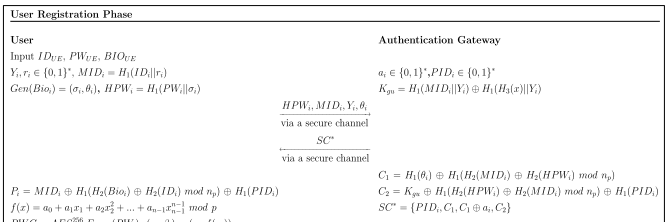

Fig. 3. User Registration Phase. This phase is implemented between U_i and AGW to generate registration credentials in a secure manner.

D. Login and Authentication Phase

The login and authentication phase is shown in Fig. 4.

1) UL1: U_i inputs ID_i^*, PW_i^*, and Bio_i^* after she inserts SC_i into the device. The device computes $\sigma_i^* = Rep(Bio_i^*,\ \theta_i)$, $HPW_i^* = H_1(PW_i^* \ ||\ \sigma_i)$, $MID_i^* = P_i \oplus H_1(PID_i) \oplus H_1(H_2(Bio_i) \oplus H_2(ID_i)\ mod\ n_p)$, and $C_1^* = H_1(\theta_i) \oplus (H_1(H_2(MID_i^*) \oplus H_2(HPW_i^*))\ mod\ n_p)$. It checks whether the equation $C_1^* = C_1$ holds. If not, the login request is rejected. Otherwise, U_i chooses a random number r_u and computes $QR_i = (H_1(MID_i) \ ||\ r_u||\ T_1)^2\ mod\ n$, $K_{gu}^* = C_2 \oplus H_1(PID_i) \oplus H_1(H_2(HPW_i) \oplus H_2(MID_i)\ mod\ n_p)$, $a_i = (C_1 \oplus a_i) \oplus C_1^*$, $CAK_i = (a_i \ ||\ K_{gu}^*) \oplus H_1(r_u \ ||\ T_1)$, $M_1 = SID_j \oplus H_1(K_{gu}^* \ ||\ r_u||\ PID_i \ ||\ T_1)$, and $M_2 = H_1(PID_i \ ||\ MID_i \ ||\ SID_j \ ||\ r_u \ ||T_1)$, where T_1 denotes the current timestamp. U_i then forwards $\{QR_i,\ CAK_i,\ M_1,\ M_2,\ T_1\}$ to AGW via a public manner.

2) UL2: Upon receving $\{QR_i',\ CAK_i',\ M_1',\ M_2',\ T_1'\}$, AGW first checks the freshness of T_1, that is $|T_1' - T_1| \leq \Delta T$. If T_1 doesn't meet the requirement, AGW will reject this request. Otherwise, it decrypts QR_i to obtain $\{H_1(MID_i'),\ r_u',\ T_1\}$. If $H_1(MID_i') = H_1(MID_i)$, it continues to compute $K_{gu}' = H_1(MID_i \ ||\ Y_i) \oplus H_1(x \ ||\ Y_i)$, $SID_j' = M_1' \oplus H_1(K_{gu}' \ ||\ r_u' \ ||\ PID_i \ ||\ T_1)$, and $M_2^* = H_1(PID_i \ ||\ MID_i \ ||\ SID_j' \ ||\ r_u' \ ||\ T_1)$. AGW further checks whether the equation $M_2^* = M_2'$ holds or not. If the above equation doesn't hold, the session will abort. Otherwise, it derives $a_i' \ ||\ K_{gu}^* = CAK_i \oplus H_1(r_u \ ||\ T_1)$. If $a_i' = a_i$, this implies that the login request is legitimate. Otherwise, the session will be rejected.

Furthermore, AGW appends K_{gs}' to the $Honey\_List$ and distrusts SC_i when the number of values appended in $Honey\_List$ exceeds the fixed threshold. Otherwise, AGW proceeds to randomly select r_g and compute $K_{gs} = H_1(SID_j \ ||\ Y_j) \oplus H_1(H_3(x) \ ||\ Y_j)$, $M_3 = r_u \oplus H_2(SID_j \ ||\ MID_i \ ||\ K_{gs} \ ||\ T_2)$, $M_4 = r_g \oplus H_2(SID_j \ ||\ MID_i \ ||\ K_{gs} \ ||\ r_u \ ||\ T_2)$,

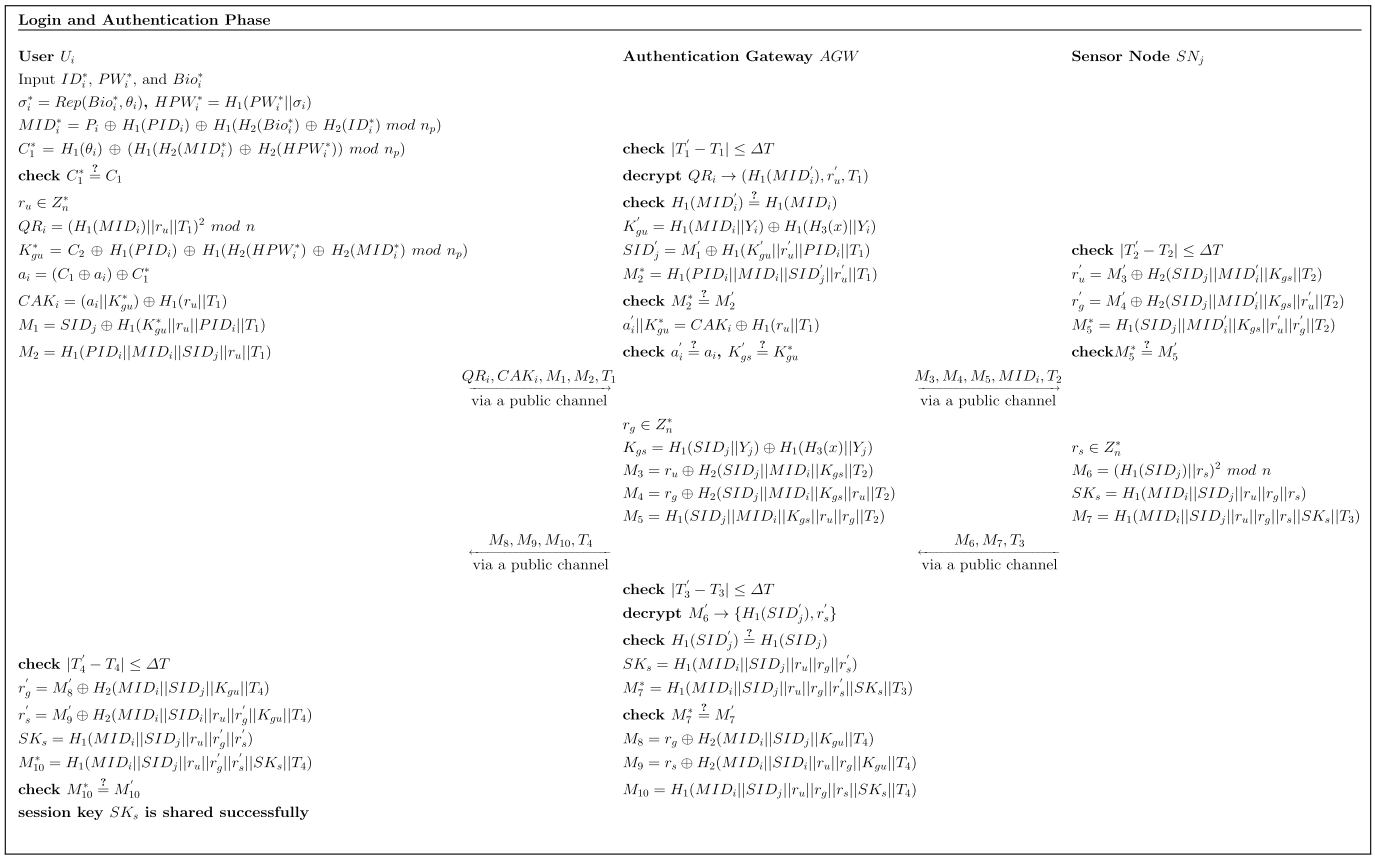

Fig. 4. Login and Authentication Phase. This phase deploys mutual authentication among U_i, AGW, and SN_j to establish a session key.

and $M_5 = H_1(SID_j \ || \ MID_i \ || \ K_{gs} \ || \ r_u \ || \ r_g \ || \ T_2)$, where T_2 denotes the current timestamp. Finally, AGW sends $\{M_3, \ M_4, \ M_5, \ MID_i, \ T_2\}$ to SN_j.

3) UL3: Upon the receipt of $\{M_3^{'}, \ M_4^{'}, \ M_5^{'}, \ MID_i^{'}, \ T_2^{'}\}$, SN_j checks the freshness of $T_2^{'}$, where $|T_2^{'} - T_2| \leq \Delta \ T$. If it isn't fresh, SN_j will reject this request. Otherwise, it continues to calculate $r_u^{'} = M_3^{'} \oplus H_2(SID_j \ || \ MID_i^{'} \ || \ K_{gs} \ || \ T_2)$, $r_g^{'} = M_4^{'} \oplus H_2(SID_j \ || \ MID_i^{'} \ || \ K_{gs} \ || \ r_u^{'} \ || \ T_2)$, and $M_5^* = H_1(SID_j \ || \ MID_i^{'} \ || \ K_{gs} \ || \ r_u^{'} \ || \ r_g^{'} \ || \ T_2)$. If the equation $M_5^* = M_5^{'}$ doesn't hold, the session will be rejected. Otherwise, SN_j selects a random number r_s and then computes $M_6 = (H_1(SID_j) \ || \ r_s)^2 \ mod \ n$, $SK_s = H_1(MID_i \ || \ SID_j \ || \ r_u \ || \ r_g \ || \ r_s)$, $M_7 = H_1(MID_i \ || \ SID_j \ || \ r_u \ || \ r_g \ || \ r_s \ || \ SK_s \ || \ T_3)$, where T_3 denotes the current timestamp. SN_j proceeds to forward $\{M_6, \ M_7, \ T_3\}$ to AGW.

4) UL4: Upon receving $\{M_6^{'}, \ M_7^{'}, \ T_3^{'}\}$, AGW first checks whether $T_3^{'}$ is fresh. If it doesn't meet the requirement, the session will be rejected. Otherwise, AGW decrypts $M_6^{'}$ to obtain $\{H_1(SID_j^{'}), \ r_s^{'}\}$. If $H_1(SID_j^{'}) = H_1(SID_j)$, it continues to compute $SK_s = H_1(MID_i \ || \ SID_j \ || \ r_u \ || \ r_g \ || \ r_s^{'})$, and $M_7^* = H_1(MID_i \ || \ SID_j \ || \ r_u \ || \ r_g \ || \ r_s^{'} \ || \ SK_s \ || \ T_3)$. AGW further examines whether the equation $M_7^* = M_7^{'}$ holds. If it doesn't hold, the session will abort. Otherwise, AGW calculates $M_8 = r_g \ \oplus \ H_2(MID_i \ || \ SID_j \ || \ K_{gu} \ || \ T_4)$, $M_9 = r_s \oplus H_2(MID_i \ || \ SID_i \ || \ r_u \ || \ r_g \ || \ K_{gu} \ || \ T_4)$, and $M_{10} = H_1(MID_i \ || \ SID_j \ || \ r_u \ || \ r_g \ || \ r_s \ || \ SK_s \ || \ T_4)$, where T_4 denotes the current timestamp. Fianlly, AGW forwards $\{M_8, \ M_9, \ M_{10}, \ T_4\}$ to U_i.

5) UL5: When U_i recevies $\{M_8^{'}, \ M_9^{'}, \ M_{10}^{'}, \ T_4^{'}\}$, she first checks the freshness of $T_4^{'}$. If $T_4^{'}$ is not fresh, then U_i rejects this session. Otherwise, U_i computes $r_g^{'} = M_8^{'} \oplus H_2(MID_i \ || \ SID_j \ || \ K_{gu} \ || \ T_4)$, $r_s^{'} = M_9^{'} \oplus H_2(MID_i \ || \ SID_i \ || \ r_u \ || \ r_g^{'} \ || \ K_{gu} \ || \ T_4)$, $SK_s = H_1(MID_i \ || \ SID_j \ || \ r_u \ || \ r_g^{'} \ || \ r_s^{'})$, and $M_{10}^* = H_1(MID_i \ || \ SID_j \ || \ r_u \ || \ r_g^{'} \ || \ r_s^{'} \ || \ SK_s \ || \ T_4)$. U_i further checks whether $M_{10}^* = M_{10}^{'}$ holds. If the above equation holds, it shows that a secret session key SK_s has been established for further communication.

E. Dynamic Password Recovery Phase

The password recovery phase is shown in Fig. 5.

As shown in Fig. 5, when forgetting and leaking her password, U_i injects SC_i and then inputs ID_i^* and Bio_i^*. If $ID_i^* = ID_i$, $\sigma_i^* = Rep(Bio_i^*, \ \theta_i) = \sigma_i$, she can achieve $\{\delta_i\}_{1 \leq i \leq N}$. Once answering questions that were set during the registration phase, she gets N answers Ans_i and then calculates $\beta_i = \delta_i \oplus H_1(H_2(Ans_i) \| H_2(ID_i) \mod n_a)$. It enables U_i to reconstruct $f(x)$ as $f(x) = \sum_{i=1}^{N} \beta_i \prod_{1 \leq j \leq N, j \neq i} (x - x_i)/(x_j - x_i) \ mod \ p$. She further computes $a_0 = f(0)$, and performs decryption algorithm $AES^{256}.Dec_{a_0}(PWC)$ to recover PW_i. Additionally, U_i can reset private questions $\{Ques_i\}_{1 \leq i \leq N} \rightarrow \{Ques_i\}_{1 \leq i \leq N}^{new}$ to resist private question attacks, and she is able to select a new polynomial to reset a_0^{new} as $f(x)^{new} = a_0^{new} + a_1^{new} x_1 + \ldots + a_{N-1}^{new} x_{N-1}^{N-1} \mod p$. Note that, dynamically updating private questions and secret values can further help

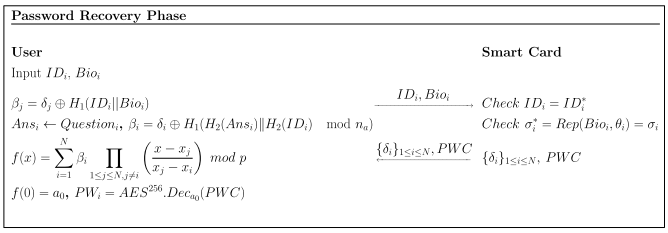

Fig. 5. Dynamic Password Recovery Phase. This phase achieves dynamic password recovery through private questions to help U_i recover her password.

resist private questions guessing attacks. That's why we call it dynamic password recovery.

F. Password and Smart Card Updating Phase

U_i inputs ID_i, PW_i and Bio_i after inserting SC_i into the specific device. It computes $\sigma_i = Rep(Bio_i, \theta_i)$, $HPW_i = H_1(PW_i \,||\, \sigma_i)$, $MID_i^* = P_i \oplus H_1(PID_i) \oplus H_1(H_2(Bio_i) \oplus H_2(ID_i) \; mod \; n_p)$, and $C_1^* = H_1(\theta_i) \oplus H_1(H_2(MID_i) \oplus H_2(HPW_i) \; mod \; n_p)$. If $C_1^* = C_1$, U_i resubmits a new password PW_i^{new} to the device. It computes $HPW_i^{new} = H_1(PW_i^{new} \,||\, \sigma_i)$, $C_1^{new} = H_1(\theta_i) \oplus H_1(H_2(MID_i) \oplus H_2(HPW_i^{new}) \; mod \; n_p)$, $C_2^{new} = K_{gu} \oplus H_1(H_2(HPW_i^{new}) \oplus H_2(MID_i) \; mod \; n_p) \oplus H_1(PID_i)$, and $C_1^{new} \oplus a_i$, and then updates related values in the SC_i.

G. Key Rotation Phase

AGW first picks a new secret key x^{new} randomly, and computes $K_{gu}^{new} = K_{gu} \oplus H_1(H_3(x) \,||\, Y_i) \oplus H_1(H_3(x^{new}) \,||\, Y_i)$, $K_{gs}^{new} = K_{gs} \oplus H_1(H_3(x) \,||\, Y_j) \oplus H_1(H_3(x^{new}) \,||\, Y_j)$ to replace the previous preshared key. AGW proceeds to generate rotation tokens denoted as $Tok_i^{new} = H_1(H_3(x) \,||\, Y_i) \oplus H_1(H_3(x^{new}) \,||\, Y_i)$, $Tok_j^{new} = H_1(H_3(x) \,||\, Y_j) \oplus H_1(H_3(x^{new}) \,||\, Y_j)$, and then forwards each of them to U_i and SN_j respectively. Upon the receipt of the rotation token, U_i and SN_j rotate previous preshared key as $K_{gu}^{new} = K_{gu} \oplus Tok_i^{new}$, $K_{gs}^{new} = K_{gs} \oplus Tok_j^{new}$.

VIII. Formal Analysis Under the ROM Model

Theorem 1: Assuming there exists an efficient adversary $\mathcal{A}$, who can perform $Execute(\Delta)$, $Send(\Omega_U^i, Start)$, $Send(\Delta, M)$, $Corrupt(\Delta, b)$, $Hash(\cdot)$, and $Test(\Delta)$ queries in polynomial time, the advantage of adversary $\mathcal{A}$ successfully breaking the security of the proposed scheme is $Adv_{\mathcal{A}}^{Suc} \le \frac{q_{H_1}^2}{2^{L_1-2}} + \frac{(q_S+q_{Exe})^2}{N_0} + \frac{q_S}{2^{L_1-2}} + 2C'q_S^{S'} + \frac{2N\lambda_{Ans_i}}{G_{Ans_i}} + \frac{q_{H_2}^2}{2^{L_2-1}} + \frac{q_{H_1}^3}{2^{L_3-2}}$, where q_S, q_{H_i}, q_{Exe} represent the number of $Send(\Delta, M)$ queries, $Hash(\cdot)$ and $Execute(\Delta)$ queries, respectively. L_i represents the length of hash function $H_i(\cdot)$, C', S', λ_{Ans_i} and G_{Ans_i} are predefined constants. The detailed formal analysis can be seen in Appendix A of the full version at https://bit.ly/4ezkQzN.

Proof: A sequence of games $GAME_i$ is defined to prove the security of MFA-DPRU. $\mathcal{A}$ wins the games if and only if she guesses b in each game $GAME_i$ successfully. The proof is presented as follows.

$\boldsymbol{GAME_1}$: This game simulates the real word attacks without performing any queries. Therefore, the advantage of $\mathcal{A}$ breaking the scheme is $Adv_{\mathcal{A}}^{Suc}(t) = |2Pr[Suc_{\mathcal{A}}^{G_1}] - 1|$.

$\boldsymbol{GAME_2}$: This game is identical to $GAME_1$ except that it can perform $Execute(\Delta)$ queries. However, only messages $\{QR_i, CAK_i, M_1, M_2, T_1, M_3, M_4, M_5, MID_i, T_2, M_6, M_7, T_3, M_8, M_9, M_{10}, T_4\}$ can be provided to $\mathcal{A}$. Notice that, due to the application of random values r_u, r_g, and r_s, it is difficult to guess b efficiently. The advantage of $\mathcal{A}$ to break the scheme is $Pr[Suc_{\mathcal{A}}^{G_1}] = Pr[Suc_{\mathcal{A}}^{G_2}]$.

$\boldsymbol{GAME_3}$: This game is identical to $GAME_2$ except that it will terminate immediately if a collision exists in the $Hash(\cdot)$ queries for SK_s or a collision exists in transcripts. The probability of collision in $Hash(\cdot)$ queries is at most $\frac{q_{H_1}^2}{2^{L_1+1}}$, and the probability of collision in transcripts is at most $\frac{(q_S+q_{Exe})^2}{2N_0}$. The advantage of $\mathcal{A}$ in this game is $Pr[Suc_{\mathcal{A}}^{G_3}] - Pr[Suc_{\mathcal{A}}^{G_2}] \le \frac{q_{H_1}^2}{2^{L_1+1}} + \frac{(q_S+q_{Exe})^2}{2N_0}$.

$\boldsymbol{GAME_4}$: This game will abort if $\mathcal{A}$ passes authentication without performing $Hash(\cdot)$ queries. The advantage satisfies $Pr[Suc_{\mathcal{A}}^{G_4}] - Pr[Suc_{\mathcal{A}}^{G_3}] \le \frac{q_S}{2^{L_1}}$.

$\boldsymbol{GAME_5}$: This game outlines the security measures of the session key SK_s, covering two specific cases. In the first case, $\mathcal{A}$ is allowed to conduct $Corrupt(\Omega_{AGW}^k, 1)$ queries to possess the long-term secret key x and then attempts to break the forward security. Note that, although in possession of x, it is infeasible for $\mathcal{A}$ to establish the session key SK_s because r_u, r_g, and r_s are confidential. In the second case, $\mathcal{A}$ has in possession of only one of the above three random values to check whether or not the scheme can resist KSSTI attacks. It is difficult to calculate any two of the three values from the other. As a consequence, the only way to break the security of the session key for $\mathcal{A}$ is to perform $Send(\Omega_U^i, Start)$, $Send(\Delta, M)$, and $Hash(\cdot)$ queries in both cases. The advantage of $\mathcal{A}$ to break the security of session key is $Pr[Suc_{\mathcal{A}}^{G_5}] - Pr[Suc_{\mathcal{A}}^{G_4}] \le \frac{q_S}{2^{L_1}} + \frac{q_{H_1}^2}{2^{L_1+1}}$.

$\boldsymbol{GAME_6}$: This game is equal to $GAME_5$, except that $\mathcal{A}$ attempts to perform password guessing attacks and private questions guessing attacks. Note that, the premise for making private questions guesses is that the smart card is lost and $\mathcal{A}$ can extract the private data. Firstly, it is required that if $Corrupt(\Omega_U^i, 2)$ is queried, $Corrupt(\Omega_U^i, 1)$ and $Corrupt(\Omega_U^i, 3)$ are forbidden to query. Through q_s queries of $Send(\Omega_U^i, Start)$ and $Send(\Delta, M)$, the probability in guessing the correct password is no more than $C' \cdot q_S^{S'}$, where C' and S' are constants related to the size of password space. Secondly, it is required that only smart cards SC can be obtained by $\mathcal{A}$. The advantage of $\mathcal{A}$ successfully guessing private questions when the adversary performs q_H hash query is $\frac{N\lambda_{Ans_i}}{G_{Ans_i}}$, where λ_{Ans_i} and G_{Ans_i} are constants related to the guessing entropy of the private question. Accordingly, the advantage of $\mathcal{A}$ in this game is $Pr[Suc_{\mathcal{A}}^{G_6}] - Pr[Suc_{\mathcal{A}}^{G_5}] \le C' \cdot q_S^{S'} + \frac{N\lambda_{Ans_i}}{G_{Ans_i}} + \frac{q_{H_1}^2}{2^{L_1+1}} + \frac{q_{H_2}^2}{2^{L_2+1}}$.

$\boldsymbol{GAME_7}$: In order to check whether the protocol can prevent impersonation attacks, it will terminate when $\mathcal{A}$ makes $Hash(\cdot)$ queries. The advantage of $\mathcal{A}$ satisfies $Pr[Suc_{\mathcal{A}}^{G_7}] - Pr[Suc_{\mathcal{A}}^{G_6}] \le \frac{q_{H_1}^2}{2^{L_1+1}} + \frac{q_{H_2}^2}{2^{L_2+1}} + \frac{q_{H_3}^2}{2^{L_3+1}}$.

According to the analysis from $GAME_1$ to $GAME_7$, the advantage breaking the proposed protocol can be calculated as

$$\begin{aligned}
&Adv_{\mathcal{A}}^{Suc}(t) = |2Pr[Succ_{\mathcal{A}}^{G_1}] - 1| \\
&= 2Pr[Succ_{\mathcal{A}}^{G_6}] - 1 + 2(Pr[Succ_{\mathcal{A}}^{G_1}] - Pr[Succ_{\mathcal{A}}^{G_6}]) \\
&\leq 2 \times \frac{1}{2} - 1 + 2\sum_{i=1}^{6}(Pr[Succ_{\mathcal{A}}^{G_{i+1}}] - Pr[Succ_{\mathcal{A}}^{G_i}]) \\
&= 2 \times [4\frac{q_{H_1}^2}{2^{L_1+1}} + \frac{(q_S + q_{Exe})^2}{2N_0} + 2\frac{q_S}{2^L} + 2\frac{q_{H_2}^2}{2^{L_2+1}} \\
&\quad + C' \cdot q_S^{S'} + \frac{q_{H_3}^2}{2^{L_3+1}} + \frac{N\lambda_{Ans_i}}{G_{Ans_i}}] \\
&= \frac{q_{H_1}^2}{2^{L_1-2}} + \frac{(q_S + q_{Exe})^2}{N_0} + \frac{q_S}{2^{L_1-2}} + 2C' q_S^{S'} + \frac{2N\lambda_{Ans_i}}{G_{Ans_i}} \\
&\quad + \frac{q_{H_2}^2}{2^{L_2-1}} + \frac{q_{H_1}^3}{2^{L_3-2}}.
\end{aligned}$$

IX. Formal Analysis of Password Recovery

Theorem 2: Assuming the existence of an efficient adversary $\mathcal{A}$ who can extract $\{Que_i\}_{1\leq N}$, PWC and $\{\delta_i\}_{1\leq i\leq N}$ from smart card SC_i, the advantage of $\mathcal{A}$ successfully recovering the password is no more than $Adv_{\mathcal{A}}^{AES^{256}} + \frac{1}{2^{Nq}} + \frac{N\lambda_{Ans_i}}{G_{Ans_i}}$, where $Adv_{\mathcal{A}}^{AES^{256}}$ denotes the advantage of $\mathcal{A}$ successfully breaking the AES^{256} algorithm and $\frac{\lambda_{Ans_i}}{G_{Ans_i}}$ denotes the advantage of guessing the private question. The complete analysis can be seen in Appendix B of the full version at https://bit.ly/4ezkQzN.

Proof: According to the definition, $f(x)$ is a randomly chosen polynomial in the finite field F_q. (x_i, β_i) and $\delta_i = \beta_i \oplus H_1(H_2(Ans_i)\|H_2(ID_i) \mod n_a)$ are uniformally distributed in F_q. Let $F_q[x]$ be a set of all polynomials with order q defined in F_q and let $S_{\delta_i}[x]$ be a set of polynomials related to δ_i, where $S_{\delta_i}[x] \subseteq F_q[x]$. When achieving δ_i, the maximum advantage of $\mathcal{A}$ in recovering the password depends on $S_{\delta_i}[x]$. Define the maximum advantage as follows: $Pr_{\mathcal{A}}^{Max}[\delta] = \frac{Pr[Ans_i]}{\sum_{Ans_i' \leftrightarrow S_{\delta_i}[x]} Pr[Ans_i']}$, where $Pr[Ans_i]$ represents the probability of $H_1(H_2(Ans_i) \parallel H_2(ID_i) \mod n_a)$ being inferred from $S_{\delta_i}[x]$.

When viewing $H_1(\cdot)$ as a random oracle, $Pr[Ans_i]$ is uniformally distributed and different from $Pr[Ans_j]$ if $i \neq j$. Therefore, $Pr_{\mathcal{A}}^{Max}[\delta] \approx 1/|S_{\delta_i}[x]| \approx 1/|F_q[x]|$, which denotes that $\mathcal{A}$ successfully reconstructs the polynomial $f(x)$. According to the construction, $f(x) = a_0 + a_1x_1 + a_2x_2^2 + \ldots + a_{N-1}x_{N-1}^{N-1} \ mod \ p$ is defined in F_q, where $a_i \in [0, p-1]$. If $\mathcal{A}$ reconstructs $f(x)$ successfully, it infers that a_i has been disclosed. Above all, $Pr_{\mathcal{A}}^{Max}[\delta] \approx 1/2^{pN}$. In order to analyze the maximum and minimum bound of $Pr[Ans_i]$, we define Min_i and Max_i as follows, $Min_i = Min_{Ans_i' \leftrightarrow S_{\delta_i}} Pr[Ans_i]$, $Max_i = Max_{Ans_i' \leftrightarrow S_{\delta_i}} Pr[Ans_i]$.

Then we get the following inequality $\sum Pr[Ans_i] \geq \frac{1}{|S_{\delta_i}[x]|} Min_{Ans_i' \leftrightarrow S_{\delta_i}[x]} Pr[Ans_i] \approx \frac{1}{2^{pN}} \prod_{i=1}^{N} Min_i$. Due to the independence of each question, $Pr_{\mathcal{A}}^{Max}[\delta] \leq \frac{1}{2^{pn}} \prod_{i=1}^{N} \frac{Max_i}{Min_i}$. Additionally, it is infeasible for $\mathcal{A}$ to distinguish $H_1(H_2(Ans_i) \parallel H_2(ID_i) \mod n_a)$ and $H_1(H_2(Ans_j) \parallel H_2(ID_i) \mod n_a)$ under the random oracle, which infers $Min_i \approx Max_i$. As a result, when achieving δ_i from smart card, the maximum advantage $Pr_{\mathcal{A}}^{Max}[\delta]$ in recovering the password is less than $\frac{1}{2^{pN}}$.

If $\mathcal{A}$ fails to achieve any valid information from δ_i, she can only obtain PWC from the smart card. If breaking the AES^{256} algorithm successfully, the advantage in recovering the password is nearly $Adv_{\mathcal{A}}^{AES^{256}}$. Due to the assumption that $\mathcal{A}$ can extract $\{Que_i\}_{1\leq N}$ from SC_i, then she attempts to perform private questions attacks to retrieve the correct answers. The advantage of $\mathcal{A}$ successfully guessing privacy questions is $\frac{N\lambda_{Ans_i}}{G_{Ans_i}}$.

Above all, the advantage of $\mathcal{A}$ successfully recovering the password is no more than $Adv_{\mathcal{A}}^{AES^{256}} + \frac{1}{2^{Nq}} + \frac{N\lambda_{Ans_i}}{G_{Ans_i}}$.

X. Informal Analysis of Security Properties

With the incorporation of the fuzzy verifier technique, honeywords method, and public key cryptosystem, the proposed scheme is immune to known attacks, such as password guessing attacks, known session-specific temporary information attacks, and key compromised impersonation attacks. Furthermore, it ensures forward security, dynamic password recovery, key rotation, and anonymity. We also model MFA-DPRU using Tamarin and then conduct automated analysis. The detailed informal analysis can be seen in Appendix C of the full version at https://bit.ly/4ezkQzN.

A. Resist Password Guessing Attacks

An adversary $\mathcal{A}$ obtains a smart card SC_i and enables to extract information $\{PID_i, C_1, C_1 \oplus a_i, C_2, P_i, \theta_i, Upd_i, Gen(\cdot), Rep(\cdot)\}$. Assuming that the biometrics Bio_i have been exposed to $\mathcal{A}$, which enables $\mathcal{A}$ to derive σ_i through the calculation of $Rep(Bio_i, \theta_i)$. Upon calculating σ_i, $\mathcal{A}$ attempts to guess PW_i and ID_i. Due to the fact that fuzzy verifier technique is applied to compute $C_1 = H_1(\theta_i) \oplus H_1(H_2(MID_i) \oplus H_2(HPW_i) \ mod \ n_p)$, it implies that there exist almost $\frac{|ID_i| \cdot |PW_i|}{N_0} \approx 2^{32}$ candidate pairs of (ID_i, PW_i) to thwart $\mathcal{A}$, where $N_0 \approx 2^8$, $|ID_i| \approx 10^6$ and $|PW_i| \approx 10^6$ represent the size of identity space and password space respectively. It is computationally infeasible to pinpoint the correct identity ID_i and password PW_i from the candidates without interacting with AGW. Fortunately, the introduction of the honeywords method reports malicious login behaviors with an accuracy of $1 - \frac{1}{2^{80}}$. Above all, it is difficult for $\mathcal{A}$ to launch password guessing attacks.

We consider another scenario where $\mathcal{A}$ compromises SN_j and obtains SID_j, K_{gs}, Y_j, but it's still difficult to rebuild the session key SK_s' established before (suppose that SN_j doesn't record SK_s after each communication). $\mathcal{A}$ intercepts $\{M_3', M_4', T_2'\}$ through public channel and proceeds to calculate $r_u' = M_3' \oplus H_2(SID_j \parallel MID_i' \parallel K_{gs} \parallel T_2)$, $r_g' = M_4' \oplus H_2(SID_j \parallel MID_i' \parallel K_{gs} \parallel r_u' \parallel T_2)$. However, it is infeasible to extract r_s' since it is transmitted in ciphertext $M_6 = (H_1(SID_j) \parallel r_s)^2 \ mod \ n$. As a result, $\mathcal{A}$ can't retrieve the previous session key even if she compromises SN_j.

B. Resist Known Session-Specific Temporary Information (KSSTI) Attacks

According to the reference [15], KSSTI attacks describe that $\mathcal{A}$ attempts to calculate the session key using random numbers like r_u, r_g, r_s. It is required that only one random value can be obtained by $\mathcal{A}$, that is, $\mathcal{A}$ is restricted to be a legitimate user, authentication gateway, or sensor node during each KSSIT attack. According to the construction of the proposed scheme, the session key is calculated as $SK_s = H_1(MID_i||SID_j||r_u||r_g||r_s)$, showing that it is computationally infeasible to rebuild the session key in the case where only one of the random values is exposed.

C. Resist Key Compromised Impersonation (KCI) Attacks

Assuming the leakage of the long-term secret key x, $\mathcal{A}$ attempts to impersonate the legitimate user. Due to $\{HPW_i, MID_i, Y_i\}$ are transmitted secretly, it's difficult to compute $K_{gu} = H_1(MID_i||Y_i) \oplus H_1(H_3(x)||Y_i)$. Additionally, because of the security of z_g and y_g, $\mathcal{A}$ fails to retrieve $\{H_1(MID_i'), r_u', T_1\}$ from QR_i'. Therefore, $\mathcal{A}$ cannot impersonate a user through KCI attacks.

D. Perfect Forward Security

Notice that, the session key is computed as $SK_s = H_1(MID_i \,||\, SID_j \,||\, r_u \,||\, r_g \,||\, r_s)$, where r_u, r_g, r_s are chosen randomly and transmitted securely. Although x is available to $\mathcal{A}$, it is still computationally infeasible for $\mathcal{A}$ to compute SK_s because she cannot decrypt the ciphertext to obtain r_u, r_s without the knowledge of y, z. Moreover, when AGW performs the key rotation to achieve a new long-term secret key x^{new}, the previous key becomes invalid. Despite the current $x^{current}$ is exposed, there is no way to rebuild the previous session key.

E. Dynamic Password Recovery and Update

When forgetting and leaking passwords, it enables users to recover passwords dynamically and then perform password updating. Once injecting the smart card and extracting $\{Ques_i\}_{1\le i\le N}$, U_i answers these questinos and gets answers Ans_i, calculates $\beta_i = \delta_i \oplus H_1(H_2(Ans_i)\|H_2(ID_i) \mod n_a)$. $f(x)$ can be reconstructed as $f(x) = \sum_{i=1}^{N} \beta_i \prod_{1\le j\le N, j\ne i} (x - x_i)/(x_j - x_i) \mod p$. Then she computes $a_0 = f(0)$, and performs decryption algorithm $AES^{256}.Dec_{a_0}(PWC)$ to recover PW_i. Note that the password can be recovered by correctly answering N private questions, and can be updated to enhance the security. These N private questions $\{Ques_i\}_{1\le i\le N}$ can also be dynamically changed, $\{Ques_i\}_{1\le i\le N} \rightarrow \{Ques_i\}_{1\le i\le N}^{new}$ to resist private question attacks. Additionally, U_i selects a new polynomial to reset a_0^{new} as follows $f(x)^{new} = a_0^{new} + a_1^{new}x_1 + \ldots + a_{N-1}^{new}x_{N-1}^{N-1} \mod p$.

F. Anonymity and Untraceability

MFA-DPRU provides anonymity for U_i and sensor node SN_j, respectively. It generates pseudo-identity MID_i for U_i, where $MID_i = H_1(ID_i \,||\, r_i)$ and then encrypts MID_i as $QR_i = (H_1(MID_i)||r_u||T_1)^2 \mod n$. It proceeds to compute the following value as $M_1 = SID_j \oplus H_1(K_{gu}^*||r_u||PID_i||T_1)$ which can be sent in public.

G. Key Rotation

When the long-term secret key x is required to be rotated to enhance the security, AGW can choose x^{new} randomly and computes $K_{gu}^{new} = K_{gu} \oplus H_1(H_3(x)||Y_i) \oplus H_1(H_3(x^{new})||Y_i)$, $K_{gs}^{new} = K_{gs} \oplus H_1(H_3(x)||Y_j) \oplus H_1(H_3(x^{new})||Y_j)$. Due to the randomness of x^{new}, K_{gu}^{new} and K_{gs}^{new} are uniformly distributed.

H. Security Analysis on Tamarin

Based on Ubuntu 22.04.4, we conduct an automated analysis of MFA-DPRU using Tamarin on a microcomputer with Intel (R) Core (TM) i5-12500H 2.50 GHz processor. The results show that MFA-DPRU is secure against password guessing attacks, and further keeps session key and forward security. The complete analysis on Tamarin can be seen in Appendix D of the full version at http://ieeexplore.ieee.org/10.1109/TIFS.2024.3451364.

XI. Performance Analysis

In this section, we conduct a comprehensive comparison of thirty-seven representative authentication schemes under the system evaluation criteria. The analysis results are provided in Table II, including security and efficiency comparison. To simplify the analysis, we assume that the size of identity, password, biometrics, random value, output of physical unclonable functions function, and fuzzy extractor value are all 160 bits. SHA-256 and AES-256 are introduced as the hash function and encryption scheme respectively.

We further perform experiments on a micro-computer deployed with Intel(R) Core(TM) i5-12500H 2.50 GHz processor. Some cryptographic primitive operations are performed based on the Pairing-Based Cryptography (PBC) library, including bilinear pairing operation, hash operation, modular exponential operation, message authentication code, symmetric encryption scheme, etc. Table IV presents the computational cost of executing these operations once. To evaluate the performance of MFA-DPRU in the same environment, some state-of-the-art MFA schemes are chosen ([7], [8], [54], [55], [56], [57], [58]) as comparative schemes and comparative results are presented in Fig. 6 and Fig. 7.

Table II shows that these representative schemes fail to satisfy the systematic evaluation criteria **C2** except MFA-DPRU, which requires dynamic password recovery. In addition, almost 90% of these schemes are unable to meet both **C4** and **C5**, which indicates that they are vulnerable to some known attacks, such as user impersonation attacks, password guessing attacks, and node capture attacks. Table II also shows that nearly 80% of them fail to provide forward security.

Li et al. [55] propose a multi-factor authentication scheme, which meets all systematic evaluation criteria except **C2**. However, Fig. 6 and Fig. 7 reveal that MFA-DPRU performs better in both communication and computation performance than Li et al. [55] due to the fact that we avoid applying

TABLE II

SECURITY AND EFFICIENCY COMPARISON AMONG RELEVANT AUTHENTICATION SCHEMES

Protocol	Year	Ref.	Evaluation Criteria*											Computational Cost*			Communication Cost* (bit)			Round
			C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	U_i	AGW	SN_j	U_i	AGW	SN_j	
Our scheme	–	–	√	√	√	√	√	√	√	√	√	√	√	√	$2T_B + 2T_C + T_E + 11T_H$	$2T_E + 19T_H$	$5T_H + T_E$	1088	2112	672	1
Jiang et al.	2024	[59]	×	×	×	×	×	×	√	√	×	√	×	×	$2T_H + (7 + 7n_s)T_E + 4T_S$	–	$5T_H + (7 + 4n_s)T_E + T_S$	11584	-	13728	1
Bera et al.	2024	[56]	√	×	√	√	×	√	√	√	×	√	×	×	$30T_H + 3T_M + 2T_B$	$27T_H + 9T_M$	$10T_H + 2T_M$	2464	8384	800	1
Wang et al.	2023	[57]	√	×	√	×	×	√	√	√	√	√	×	×	$21T_H + 2T_M + 2T_B$	$24T_H + 2T_M$	$6T_H + 2T_M$	2464	8448	1280	1
Fatima et al.	2023	[8]	√	×	√	×	×	√	√	×	√	√	√	×	$2T_B + T_E + 4T_H$	$T_E + 8T_H$	$5T_H$	1024	2592	992	1
Aljanah et al.	2023	[60]	√	×	√	×	×	√	√	√	√	√	×	×	$T_H + 4T_S$	$3T_H + 7T_S$	$T_H + T_S$	1984	1536	228	2
Chaudhary et al.	2023	[61]	√	×	√	×	×	√	√	√	√	√	√	×	$7T_H + 4T_U$	$2T_H + 2T_U$	$T_H + T_U$	2464	3744	1568	2
Braeken et al.	2023	[25]	√	×	√	×	×	×	√	×	√	√	×	×	$3T_H + 4T_P + 3T_S + 2T_U$	–	$4T_H + 3T_P + T_S + T_U$	288	–	768	1
Chen et al.	2023	[62]	√	×	√	×	×	√	√	√	√	√	×	√	$5T_H + 4T_U + 2T_B$	–	$8T_H + T_B$	2048	–	2576	2
Amir et al.	2023	[63]	√	×	√	×	×	√	√	×	√	√	√	√	$28T_H + T_U$	$2T_H$	$21T_H$	4096	6944	2816	2
Tahir et al.	2023	[64]	√	×	√	×	×	√	√	×	√	√	×	×	$3T_H + T_B$	$3T_H$	$10T_H + 2T_S$	864	1376	1984	1
Deebak et al.	2023	[65]	√	×	√	×	×	√	√	√	√	√	×	×	$6T_H$	$5T_H + T_C$	$4T_H$	1312	3018	823	2
Jiang et al.	2022	[58]	√	×	√	×	×	×	√	√	×	√	×	×	$22T_H + 5T_M + 2T_B$	$15T_H + 5T_M$	$4T_H$	3840	3328	512	1
Guo et al.	2022	[66]	√	×	√	×	×	√	√	×	√	√	×	×	$28T_H$	$46T_H$	$22T_H$	2880	2368	480	1
Xia et al.	2022	[67]	√	×	√	×	×	√	√	×	√	√	×	×	$T_B + 11T_H + T_S$	$10T_H + 4T_S$	$T_B + 2T_S + 6T_H$	1248	2688	928	1
Sutrala et al.	2022	[68]	√	×	√	×	×	√	√	×	√	√	×	√	$16T_H + T_B + 5T_P$	$9T_H + 3T_P$	$8T_H + 4T_P$	2016	1504	248	1
Rafique et al.	2022	[69]	√	×	√	×	×	√	√	×	√	√	×	×	$T_S + T_B + 10T_H$	$4T_S + 8T_H$	$4T_H + 2T_S$	1248	2368	416	2
Zhang et al.	2022	[70]	√	×	√	×	×	×	√	√	×	√	√	×	$T_B + 5T_H + 2T_S$	$T_B + 4T_H + 6T_S$	$T_B + 2T_H + 2T_S$	1056	768	1184	1
Li-Tian.	2022	[7]	√	×	√	×	×	√	√	×	√	√	×	×	$T_B + 11T_H$	$13T_H$	$6T_H$	1920	2688	672	1
Srinivas et al.	2021	[54]	√	×	√	×	×	√	√	×	√	×	×	×	$T_B + 17T_H + 6T_P$	$12T_H + 3T_P$	$8T_H + 4T_P$	1856	1760	1184	1
Li et al.	2021	[71]	√	×	√	×	×	√	√	×	√	×	×	×	$10T_H$	$9T_H$	$7T_H$	1920	3168	2176	1
Meshram et al.	2021	[72]	√	×	√	×	×	√	√	√	√	×	×	×	$12T_H + T_B$	$11T_H$	$11T_H$	2976	2208	768	1
Chaudhry et al.	2021	[73]	√	×	√	×	×	×	√	×	√	√	×	×	$17T_H + T_B$	$13T_H$	$8T_H$	1344	3296	928	1
Wu et al.	2021	[74]	√	×	√	×	×	×	√	√	√	√	×	×	$11T_H + T_B$	$14T_H$	$6T_H$	2112	2784	992	1
Sahoo et al.	2021	[75]	√	×	√	×	×	√	√	×	√	√	×	×	$T_B + 8T_H + 2T_S + 3T_P$	$37T_H + T_E$	$4T_H + 2T_S + 2T_P$	1952	1888	832	1
Guo et al.	2021	[76]	×	×	×	×	×	×	√	×	√	√	×	×	$21T_H$	$18T_H$	$8T_H$	1440	5132	1696	1
Li et al.	2020	[55]	√	×	√	√	√	√	√	√	√	√	√	√	$3T_P + 10T_H + T_B$	$T_P + 8T_H$	$2T_P + 4T_H$	1600	2944	768	1
Mo et al.	2020	[77]	√	×	√	√	×	√	√	×	√	√	√	×	$11T_H + T_E$	$13T_H + T_E$	$7T_H$	992	4032	672	1
Fotouhi et al.	2020	[78]	×	×	×	×	×	√	√	√	×	√	×	√	$10T_H$	$17T_H$	$6T_H$	1504	3104	672	1
Hajian et al.	2020	[79]	√	×	√	×	×	√	√	×	√	√	√	×	$13T_H$	$7T_H$	$9T_H$	1824	2656	928	1
Wazid et al.	2020	[80]	√	×	√	×	×	√	√	×	√	√	√	×	$16T_H + T_B$	$8T_H$	$8T_H$	1344	1696	828	1
Lee et al.	2020	[81]	√	×	√	×	×	√	√	×	√	√	√	×	$9T_H$	$7T_H$	$7T_H + 2T_S$	1504	3264	928	1
Vinoth et al.	2020	[3]	√	×	√	×	×	√	√	√	√	√	√	×	$9T_H + T_B + T_S$	$6T_H + 3T_S + 2T_E$	$4T_H + 2T_S$	1152	2688	544	1
Srinivas et al.	2020	[4]	√	×	√	×	×	√	√	×	√	√	×	√	$2T_C + 14T_H + T_B$	$10T_H$	$2T_C + 6T_H$	1952	1344	672	1
Li et al.	2019	[82]	√	×	√	×	×	×	√	√	×	×	×	×	$15T_H + (2n_s + 14)T_E + (2n_s + 2)T_M + 2T_B$	–	$6T_H + (n_s + 2)T_M + (n_s + 9)T_E$	10912	-	10496	1
Lin	2019	[83]	√	×	√	×	×	√	√	×	√	√	√	×	$3T_H + 2T_E$	–	$3T_H + T_E$	1248	–	800	1
Lu et al.	2019	[2]	√	×	√	×	×	√	√	×	√	√	×	×	$T_B + 6T_H + T_S + 3T_P$	$T_P + 4T_H + T_S$	$2T_P + 2T_H + 2T_S$	1408	800	544	1
Zhang et al.	2019	[27]	×	×	√	×	×	√	√	√	×	√	×	√	$6T_P + 2T_M + T_B$	–	$T_S + 8T_P + 2T_M$	1216	–	1420	1

The details of criteria **C1-C12** are referred to Sec.12. *: "√" denotes the scheme can provide the corresponding attribute. "×" denotes that the scheme cannot provide the corresponding attribute; "-" means that it fails to apply to the scheme. T_E, T_P, T_C, T_B, T_H, T_S, T_M, T_U denote the operation time for modular exponentiation, bilinear pairing, Chebysev chaotic-map, fuzzy extractor, hash, symmetric encryption, message authentication code and PUF operation respectively.

TABLE III

PERFORMANCE AND FUNCTIONALITY COMPARISON AMONG RELEVANT AUTHENTICATION SCHEMES

Scheme	Year	Ref.	Authentication factor	Computation cost (ms)			Communication cost (bits)			Key rotation	Password updating	Dynamic password recovery	Multi-factor security
				User	Sensor node	Total	User	Sensor node	Total				
Our scheme	-	-	Password+Biometric+Smart Card	$2T_B + 2T_C + T_E + 11T_H$	$5T_H + T_E$	46.164	1088	672	1760	√	√	√	√
Bera et al.	2024	[56]	Identity+Biometric	$30T_H + 3T_M + 2T_B$	$10T_H + 2T_M$	78.351	2464	800	3264	×	√	×	×
Wang et al..	2023	[57]	Password+Biometric+Smart Card	$21T_H + 2T_M + 2T_B$	$6T_H + 2T_M$	58.439	2464	1280	3744	×	√	×	√
Fatima et al.	2023	[8]	Password+Biometric+Smart Card	$2T_B + T_E + 4T_H$	$5T_H$	24.916	1024	992	2016	×	×	×	×
Li-Tian	2022	[7]	Password+Biometric+Smart Card	$T_B + 11T_H$	$6T_H$	26.847	1920	672	2592	×	√	×	×
Srinivas et al.	2021	[54]	Password+Biometric+Smart Card	$T_B + 17T_H + 6T_P$	$8T_H + 4T_P$	105.631	1856	1184	3040	×	√	×	×
Jiang et al.	2020	[58]	Password+Biometric+Smart Card	$22T_H + 5T_M + 2T_B$	$4T_H$	58.398	3840	512	4352	×	√	×	×
Li et al.	2020	[55]	Password+Biometric+Smart Card	$3T_P + 10T_H + T_B$	$2T_P + 4T_H$	56.834	1600	768	2368	×	√	×	×

† Note that "√" means achieving the corresponding goal, while "×" means failing to achieving the corresponding goal.

TABLE IV

OPERATION COST AND DESCRIPTION

Operation Cost	Description	Operation Cost	Description
$T_H \approx 1.413$	One Hash operation	$T_B \approx 3.251$	One Fuzzy Extractor function
$T_P \approx 6.827$	One Bilinear Paring operation	$T_F \approx 1.665$	One Physical Unclonable function
$T_M \approx 3.852$	One Message Authentication Code	$T_S \approx 2.172$	One Symmetric Encryption operation
$T_E \approx 2.863$	One Modular Exponential operation	$T_S \approx 2.359$	One Extendable Chebysev Chaotic Map

complex operations like pairing. Although it is poor in communication and computation performance compared to Li and Tian [7] and Fatima et al. [8], we achieve desirable security properties like dynamic password recovery, and key rotation. To be equipped with these security properties, MFA-DPRU incorporates the fuzzy verifier technique, honeywords method, and public key algorithm to prevent some known attacks, such as password guessing attacks, user impersonation attacks, and node capture attacks. Practically, it provides dynamic password recovery to address password forgetting and leakage issues using the secret-sharing method and supports key rotation to achieve perfect forward security when the current long-term key is compromised.

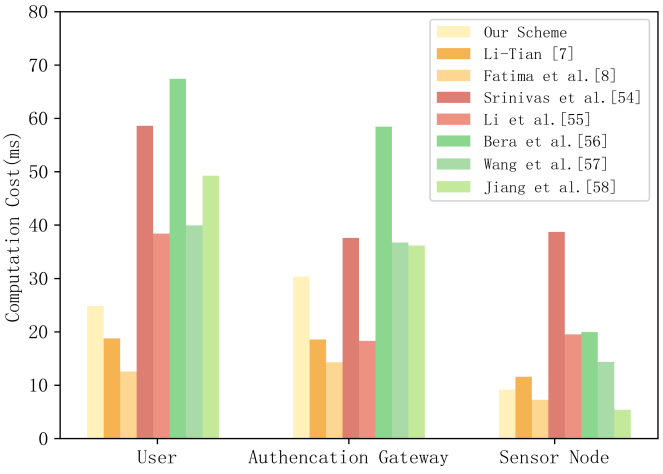

Fig. 6. Comparison on computation cost of schemes.

Table III shows comparative results of MFA-DPRU and the state-of-the-art works in terms of performance and functionality. It can be seen from Table III that most of the existing

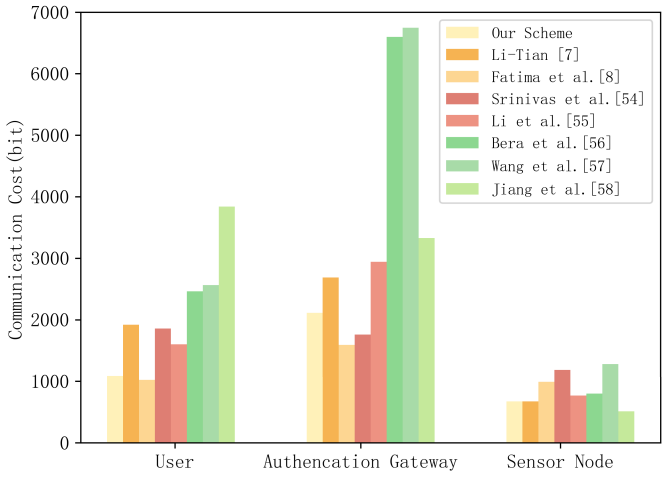

Fig. 7. Comparision on communication cost of schemes.

schemes are based on passwords, biometrics, and smart cards, but few of them can achieve multi-factor security. Although these schemes take password updating into consideration, they are based on the assumption that users will not forget or leak their passwords. As a result, none of them can perform password updating when users fail to provide previous passwords because they fail to address password forgetting and leakage issues. Additionally, these works fail to provide key rotation for the long-term secret key.

Compared to these works, MFA-DPRU can achieve multi-factor security and dynamic password recovery. It also provides key rotation to ensure the perfect forward security for the secret key. The performance analysis shows that the computation cost of the proposed scheme is more expensive than works [7] and [8], but is lower than references [54], [55], [56], [57], and [58]. In addition, the communication cost of the proposed scheme is the lowest among these schemes. Therefore, MFA-DPRU is superior in efficiency and security.

XII. Conclusion

Wireless Sensor Networks (WSNs) have greatly impacted social life and have been widely deployed. In this paper, we first review two representative schemes proposed by Li and Tian [7] and Fatima et al. [8], and point out that both of them fail to resist some known attacks and are unable to address the passwords forgetting and leakage issues. In order to solve the above problems, We propose a robust MFA scheme for WSNs with dynamic password recovery and updating, named MFA-DPRU, that has been proven to be secure under the random oracle model. Furthermore, heuristic analysis results show that MFA-DPRU can resist known attacks. Compared to the state-of-the-art works, MFA-DPRU is superior in security and efficiency for WSNs.

References

[1] N. Mahlake, T. E. Mathonsi, D. D. Plessis, and T. Muchenje, "A lightweight encryption algorithm to enhance wireless sensor network security on the Internet of Things," *J. Commun.*, vol. 18, pp. 47–57, Jul. 2023.

[2] Y. Lu, G. Xu, L. Li, and Y. Yang, "Anonymous three-factor authenticated key agreement for wireless sensor networks," *Wireless Netw.*, vol. 25, no. 4, pp. 1461–1475, May 2019.

[3] R. Vinoth, L. J. Deborah, P. Vijayakumar, and N. Kumar, "Secure multifactor authenticated key agreement scheme for industrial IoT," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3801–3811, Mar. 2021.

[4] J. Srinivas, A. K. Das, M. Wazid, and N. Kumar, "Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial Internet of Things," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 6, pp. 1133–1146, Nov. 2020.

[5] Q. Wang, D. Wang, C. Cheng, and D. He, "Quantum2FA: Efficient quantum-resistant two-factor authentication scheme for mobile devices," *IEEE Trans. Dependable Secur. Comput.*, vol. 20, no. 1, pp. 193–208, Jan. 2023.

[6] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, "Understanding node capture attacks in user authentication schemes for wireless sensor networks," *IEEE Trans. Depend. Secur. Comput.*, vol. 19, no. 1, pp. 507–523, Sep. 2022.

[7] Y. Li and Y. Tian, "A lightweight and secure three-factor authentication protocol with adaptive privacy-preserving property for wireless sensor networks," *IEEE Syst. J.*, vol. 16, no. 4, pp. 6197–6208, Dec. 2022.

[8] M. N. Fatima, M. S. Obaidat, K. Mahmood, S. Shamshad, M. A. Saleem, and M. F. Ayub, "Privacy-preserving three-factor authentication protocol for wireless sensor networks deployed in agricultural field," *ACM Trans. Sensor Netw.*, vol. 1, pp. 1–25, Jul. 2023, doi: 10.1145/3607142.

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

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

[11] D. He, Y. Gao, S. Chan, C. Chen, and J. Bu, "An enhanced two-factor user authentication scheme in wireless sensor networks," *Ad Hoc Sensor Wireless Netw.*, vol. 10, no. 4, pp. 361–371, 2010.

[12] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 4957–4968, Sep. 2019.

[13] M. M. Fouda, Z. M. Fadlullah, N. Kato, R. Lu, and X. S. Shen, "A lightweight message authentication scheme for smart grid communications," *IEEE Trans. Smart grid*, vol. 2, no. 4, pp. 675–685, Dec. 2011.

[14] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li, and A. K. Sangaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Generat. Comput. Syst.*, vol. 81, pp. 557–565, Apr. 2018.

[15] D. Abbasinezhad-Mood and M. Nikooghadam, "Design and hardware implementation of a security-enhanced elliptic curve cryptography based lightweight authentication scheme for smart grid communications," *Future Gener. Comput. Syst.*, vol. 84, pp. 47–57, Jul. 2018.

[16] K. Xue, C. Ma, P. Hong, and R. Ding, "A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 316–323, 2013.

[17] M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Netw.*, vol. 20, pp. 96–112, Sep. 2014.

[18] M. S. Farash, M. Turkanović, S. Kumari, and M. Hölbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment," *Ad Hoc Netw.*, vol. 36, pp. 152–176, Jan. 2016.

[19] R. Amin and G. Biswas, "A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," *Ad Hoc Netw.*, vol. 36, pp. 58–80, Jan. 2016.

[20] R. Amin, S. H. Islam, G. Biswas, M. K. Khan, L. Leng, and N. Kumar, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Comput. Netw.*, vol. 101, pp. 42–62, Jun. 2016.

[21] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, and Y. Yang, "An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 76, pp. 37–48, Dec. 2016.

[22] F. Wu, L. Xu, S. Kumari, and X. Li, "A new and secure authentication scheme for wireless sensor networks with formal proof," *Peer Peer Netw. Appl.*, vol. 10, no. 1, pp. 16–30, Jan. 2017.

[23] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4081–4092, Sep. 2018.

[24] Y. Zhang, B. Li, J. Wu, B. Liu, R. Chen, and J. Chang, "Efficient and privacy-preserving blockchain-based multifactor device authentication protocol for cross-domain IIoT," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22501–22515, Nov. 2022.

[25] A. Braeken, "Highly efficient bidirectional multi-factor authentication and key agreement for real-time access to sensor data," *IEEE Internet Things J.*, vol. 10, no. 23, pp. 21089–21099, Dec. 2023.

[26] M. Keith, B. Shao, and P. J. Steinbart, "The usability of passphrases for authentication: An empirical field study," *Int. J. Hum.-Comput. Stud.*, vol. 65, no. 1, pp. 17–28, Jan. 2007.

[27] R. Zhang, Y. Xiao, S. Sun, and H. Ma, "Efficient multi-factor authenticated key exchange scheme for mobile communications," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 4, pp. 625–634, Jul. 2019.

[28] I. Ul Haq, J. Wang, and Y. Zhu, "Secure two-factor lightweight authentication protocol using self-certified public key cryptography for multi-server 5G networks," *J. Netw. Comput. Appl.*, vol. 161, Jul. 2020, Art. no. 102660.

[29] C. Ellison, C. Hall, R. Milbert, and B. Schneier, "Protecting secret keys with personal entropy," *Future Gener. Comput. Syst.*, vol. 16, no. 4, pp. 311–318, Feb. 2000.

[30] N. Frykholm and A. Juels, "Error-tolerant password recovery," in *Proc. 8th ACM Conf. Comput. Commun. Secur.*, Nov. 2001, pp. 1–9.

[31] M. Just and D. Aspinall, "Personal choice and challenge questions: A security and usability assessment," in *Proc. 5th Symp. Usable Privacy Secur.*, Jul. 2009, pp. 1–11.

[32] J. Bonneau, E. Bursztein, I. Caron, R. Jackson, and M. Williamson, "Secrets, lies, and account recovery: Lessons from the use of personal knowledge questions at Google," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 141–150.

[33] M. Golla and M. Dürmuth, "Analyzing 4 million real-world personal knowledge questions (short paper)," in *Proc. Passwords*, 2015, pp. 39–44.

[34] T. Denning, K. Bowers, M. van Dijk, and A. Juels, "Exploring implicit memory for painless password recovery," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, May 2011, pp. 2615–2618.

[35] W. M. Kharudin, N. F. M. Din, and M. Z. Jali, "Password recovery using graphical method," in *Proc. WICT*, 2014, pp. 11–20.

[36] L. Jin, H. Takabi, and J. B. D. Joshi, "Security and privacy risks of using E-Mail address as an identity," in *Proc. IEEE 2nd Int. Conf. Social Comput.*, Aug. 2010, pp. 906–913.

[37] F. A. Maqbali and C. J. Mitchell, "Email-based password recovery–risking or rescuing users?" in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2018, pp. 1–5.

[38] Y. Li, Z. Chen, H. Wang, K. Sun, and S. Jajodia, "Understanding account recovery in the wild and its security implications," *IEEE Trans. Dependable Secur. Comput.*, vol. 19, no. 1, pp. 620–634, Jan. 2022.

[39] V. Zimmermann, "From the quest to replace passwords towards supporting secure and usable password creation," Ph.D. dissertation, Technische Universität Darmstadt, Germany, 2021.

[40] S. Alroomi and F. Li, "Measuring website password creation policies at scale," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, vol. 800, Nov. 2023, pp. 3108–3122.

[41] E. Stobert and R. Biddle, "The password life cycle," *ACM Trans. Privacy Secur.*, vol. 21, no. 3, pp. 1–32, Aug. 2018.

[42] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Trans. Inf. Forensics Security*, vol. 12, pp. 2776–2791, 2017.

[43] Z. Hou and D. Wang, "New observations on Zipf's law in passwords," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 517–532, 2023.

[44] R. Shay et al., "Encountering stronger password requirements: User attitudes and behaviors," in *Proc. 6th Symp. Usable Privacy Secur.*, Jul. 2010, pp. 1–20.

[45] S. Yang, S. Ji, and R. Beyah, "DPPG: A dynamic password policy generation system," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 3, pp. 545–558, Mar. 2018.

[46] R. Madhusudhan and R. C. 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, Jul. 2012.

[47] 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. 2015.

[48] 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. 2018.

[49] Q. Wang and D. Wang, "Understanding failures in security proofs of multi-factor authentication for mobile devices," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 597–612, 2022, doi: 10.1109/TIFS.2022.3227753.

[50] M. Pernpruner, R. Carbone, G. Sciarretta, and S. Ranise, "An automated multi-layered methodology to assist the secure and risk-aware design of multi-factor authentication protocols," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 4, pp. 1935–1950, 2024.

[51] N. Li, F. Guo, Y. Mu, W. Susilo, and S. Nepal, "Fuzzy extractors for biometric identification," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 667–677.

[52] J. Bonneau, C. Herley, P. C. v. Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *Proc. IEEE Symp. Secur. Privacy*, May 2012, pp. 553–567.

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

[54] J. Srinivas, A. K. Das, M. Wazid, and A. V. Vasilakos, "Designing secure user authentication protocol for big data collection in IoT-based intelligent transportation system," *IEEE Internet Things J.*, vol. 8, no. 9, pp. 7727–7744, May 2021.

[55] X. Li, J. Peng, M. S. Obaidat, F. Wu, M. K. Khan, and C. Chen, "A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems," *IEEE Syst. J.*, vol. 14, no. 1, pp. 39–50, Mar. 2020.

[56] B. Bera et al., "BioKA-ASVN: Biometric-based key agreement scheme for air smart vehicular networks using blockchain service," *IEEE Trans. Veh. Technol.*, vol. 73, no. 7, pp. 9478–9494, Jul. 2024, doi: 10.1109/TVT.2024.3380392.

[57] C. Wang, D. Wang, Y. Duan, and X. Tao, "Secure and lightweight user authentication scheme for cloud-assisted Internet of Things," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 2961–2976, 2023, doi: 10.1109/TIFS.2023.3272772.

[58] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K. R. Choo, "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9390–9401, Sep. 2020.

[59] C. Jiang, C. Xu, Y. Han, Z. Zhang, and K. Chen, "Two-factor authenticated key exchange from biometrics with low entropy rates," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 3844–3856, Apr. 2024, doi: 10.1109/TIFS.2024.3372812.

[60] S. AlJanah, N. Zhang, and S. Wah Tay, "A multi-factor homomorphic encryption based method for authenticated access to IoT devices," 2023, *arXiv:2307.03291*.

[61] R. R. Kumar Chaudhary and K. Chatterjee, "A lightweight PUF based multi-factor authentication technique for intelligent smart healthcare system," *Peer Peer Netw. Appl.*, vol. 16, no. 4, pp. 1975–1992, Aug. 2023.

[62] Z. Chen et al., "FSMFA: Efficient firmware-secure multi-factor authentication protocol for IoT devices," *Internet Things*, vol. 21, Apr. 2023, Art. no. 100685.

[63] A. M. A. Modarres and G. Sarbishaei, "An improved lightweight two-factor authentication protocol for IoT applications," *IEEE Trans. Ind. Informat.*, vol. 19, no. 5, pp. 6588–6598, May 2023.

[64] H. Tahir, K. Mahmood, M. F. Ayub, M. A. Saleem, J. Ferzund, and N. Kumar, "Lightweight and secure multi-factor authentication scheme in VANETs," *IEEE Trans. Veh. Technol.*, vol. 5, no. 1, pp. 1–10, Jul. 2023.

[65] B. D. Deebak and S. O. Hwang, "Intelligent drone-assisted robust lightweight multi-factor authentication for military zone surveillance in the 6G era," *Comput. Netw.*, vol. 225, Apr. 2023, Art. no. 109664.

[66] Y. Guo, Z. Zhang, and Y. Guo, "SecFHome: Secure remote authentication in fog-enabled smart home environment," *Comput. Netw.*, vol. 207, Apr. 2022, Art. no. 108818.

[67] Y. Xia, R. Qi, and S. Ji, "Puf-assisted lightweight group authentication and key agreement protocol in smart home," *Wireless Commun. Mob. Comput.*, vol. 2022, no. 1, 2022, Art. no. 8865158.

[68] A. K. Sutrala, M. S. Obaidat, S. Saha, A. K. Das, M. Alazab, and Y. Park, "Authenticated key agreement scheme with user anonymity and untraceability for 5G-enabled softwarized industrial cyber-physical systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2316–2330, Mar. 2022.

[69] F. Rafique, M. S. Obaidat, K. Mahmood, M. F. Ayub, J. Ferzund, and S. A. Chaudhry, "An efficient and provably secure certificateless protocol for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 8039–8046, Nov. 2022.

[70] L. Zhang, Y. Zhu, W. Ren, Y. Zhang, and K.-K. R. Choo, "Privacy-preserving fast authentication and key agreement for e-health systems in IoT, based on three-factor authentication," *IEEE Trans. Services Comput.*, vol. 16, no. 2, pp. 1324–1333, Nov. 2023, doi: 10.1109/TSC.2022.3149940.

[71] J. Li, Z. Su, D. Guo, K. R. Choo, and Y. Ji, "PSL-MAAKA: Provably secure and lightweight mutual authentication and key agreement protocol for fully public channels in Internet of Medical Things," *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13183–13195, Sep. 2021.

[72] C. Meshram, M. S. Obaidat, C.-C. Lee, and S. G. Meshram, "An efficient, robust, and lightweight subtree-based three-factor authentication procedure for large-scale DWSN in random Oracle," *IEEE Syst. J.*, vol. 15, no. 4, pp. 4927–4938, Dec. 2021.

[73] S. A. Chaudhry, A. Irshad, K. Yahya, N. Kumar, M. Alazab, and Y. B. Zikria, "Rotating behind privacy: An improved lightweight authentication scheme for cloud-based IoT environment," *ACM Trans. Internet Technol.*, vol. 21, no. 3, pp. 1–19, Aug. 2021.

[74] T. Wu and L. Yang, "A provably secure three-factor authentication protocol for wireless sensor networks," *Wireless Commun. Mob. Comput.*, vol. 2021, no. 1, pp. 1–15, 2021.

[75] S. S. Sahoo, S. Mohanty, and B. Majhi, "A secure three factor based authentication scheme for health care systems using IoT enabled devices," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 1, pp. 1419–1434, 2021.

[76] Y. Guo, Z. Zhang, and Y. Guo, "Anonymous authenticated key agreement and group proof protocol for wearable computing," *IEEE Trans. Mobile Comput.*, vol. 21, no. 8, pp. 2718–2731, Aug. 2022.

[77] J. Mo, W. Shen, and W. Pan, "An improved anonymous authentication protocol for wearable health monitoring systems," *Wireless Commun. Mobile Comput.*, vol. 2020, no. 1, pp. 1–13, Apr. 2020.

[78] M. Fotouhi, M. Bayat, A. K. Das, H. A. N. Far, S. M. Pournaghi, and M. A. Doostari, "A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT," *Comput. Netw.*, vol. 177, Aug. 2020, Art. no. 107333.

[79] R. Hajian, S. ZakeriKia, S. H. Erfani, and M. Mirabi, "SHAPARAK: Scalable healthcare authentication protocol with attack-resilience and anonymous key-agreement," *Comput. Netw.*, vol. 183, Dec. 2020, Art. no. 107567.

[80] M. Wazid, A. K. Das, V. Bhat K, and A. V. Vasilakos, "LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment," *J. Netw. Comput. Appl.*, vol. 150, Jan. 2020, Art. no. 102496.

[81] H. Lee, D. Kang, J. Ryu, D. Won, H. Kim, and Y. Lee, "A three-factor anonymous user authentication scheme for Internet of Things environments," *J. Inf. Secur. Appl.*, vol. 52, Jun. 2020, Art. no. 102494.

[82] W. Li, H. Cheng, P. Wang, and K. Liang, "Practical threshold multi-factor authentication," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3573–3588, 2021.

[83] H.-Y. Lin, "Traceable anonymous authentication and key exchange protocol for privacy-aware cloud environments," *IEEE Syst. J.*, vol. 13, no. 2, pp. 1608–1617, Jun. 2019.

Liufu Zhu received the M.S. degree in information security from Fujian Normal University, Fuzhou, China, in June 2023. He is currently pursuing the Ph.D. degree with the College of Cyber Science, Nankai University, Tianjin, China. His research interests include public-key cryptography and password-based authentication.

Ding Wang received the Ph.D. degree in information security from Peking University in 2017. He was supported by the "Boya Postdoctoral Fellowship" with Peking University from 2017 to 2019. He is currently a Full Professor with Nankai University. As the first author (or the corresponding author), he has published more than 90 papers at venues, such as IEEE S&P, ACM CCS, NDSS, USENIX Security, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, and IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. His research has been reported by over 200 medias, such as Daily Mail, Science Daily, Forbes, IEEE Spectrum and Communications of the ACM, appeared in the Elsevier 2017" Article Selection Celebrating Computer Science Research in China," and resulted in the revision of the authentication guideline NIST SP800-63-2. His research interests include passwords, authentication, and provable security. He has been involved in the community as the PC Chair/TPC Member for over 60 international conferences, such as NDSS 2023–2025, ACM CCS 2022, IEEE EuroS&P 2025, PETS 2022–2024, ACSAC 2020–2024, RAID 2024/2023, ACM AsiaCCS 2025/2022/2021, FC 2025, IFIP SEC 2018–2021, ICICS 2018–2024, and SPNCE 2020–2022. He has received the ACM China Outstanding Doctoral Dissertation Award, the Best Paper Award from INSCRYPT 2018, the Outstanding Youth Award of China Association for Cryptologic Research, the Young Scientist Nomination Award for Powerful Nation, and the First Prize of Natural Science Award of Ministry of Education.

Robust Multi-Factor Authentication for WSNs with Dynamic Password Recovery (Appendix File)

Liufu Zhu, and Ding Wang

Abstract—This appendix file consists of four parts. Appendix A shows the formal security analysis of MFA-DPRU under the ROM model, Appendix B proves the security of the password recovery, Appendix C provides an informal security analysis, and Appendix D presents the security analysis of MFA-DPRU based on Tamarin.

Index Terms—Appendix file.

APPENDIX A: FORMAL ANALYSIS OF SECURITY UNDER THE ROM MODEL

Theorem 1. Assuming there exists an efficient adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ breaking the proposed scheme is no more than $\frac{q_{H_1}^2}{2^{L_1-2}} + \frac{(q_S+q_{Exe})^2}{N_0} + \frac{q_S}{2^{L_1-2}} + 2C^{'}q_S^{S^{'}} + \frac{2N\lambda_{Ans_i}}{G_{Ans_i}} + \frac{q_{H_2}^2}{2^{L_2-1}} + \frac{q_{H_1}^3}{2^{L_3-2}}$, where q_S, q_{H_i}, q_{Exe} represent the number of $Send(\Delta,\ M)$ $Hash(\cdot)$, and $Execute(\Delta)$ queries. L_i represents the length of hash function $H_i(\cdot)$, $C^{'}$, $S^{'}$, λ_{Ans_i} and G_{Ans_i} are four predefined constants related to password and private question.

Under the ROM model, MFA-DPRU is secure against known attacks. Let Ω_U^i denotes the user U_i, Ω_S^j denotes the sensor node SN_j, and Ω_{AGW}^k denotes the authentication gateway AGW. Suppose the existance of $\mathcal{A}$, then she will use $\Delta = \{\Omega_U^i,\ \Omega_S^j,\ \Omega_{AGW}^k\}$ to perform following queries.

(1) $Execute(\Delta)$. This oracle enables to examine all passive eavesdropping attacks performed by $\mathcal{A}$ and outputs session transcripts during the login and authentication phase from Ω_U^i, Ω_S^j, and Ω_{AGW}^k.

(2) $Send(\Omega_U^i,\ Start)$. This query is defined to be the initialization phase of the proposed scheme.

(3) $Send(\Delta, M)$. This oracle can capture active attacks. $\mathcal{A}$ can block and intercept messages transmitted in public and then forges M that is sent to Δ. Then, $\mathcal{A}$ could receive a response from Δ.

(4) $Corrupt(\Delta,\ b)$. According to the systematic evaluation criteria, $\mathcal{A}$ is allowed to obtain any two of the three authentication factors such as $(SC,\ PW_i)$, $(SC,\ Bio_i)$, and $(PW_i,\ Bio_i)$, except for all of them. There exist four cases in this oracle query:

- (a) $(\Omega_U^i,\ 1)$. It denotes that $\mathcal{A}$ achieves U_i's password PW_i and smart card SC_i.
- (b) $(\Omega_U^i,\ 2)$. It denotes that $\mathcal{A}$ obtains U_i's biometrics Bio_i and smart card SC_i.
- (c) $(\Omega_U^i,\ 3)$. It denotes that $\mathcal{A}$ captures U_i's password PW_i and biometrics Bio_i.
- (d) $(\Omega_{AGW}^k,\ 1)$. This denotes that the long-term secret key x of AGW is exposed to $\mathcal{A}$.

(5) $Hash(\cdot)$. This oracle will output a random value as a response when a string is input by $\mathcal{A}$.

(6) $Test(\Delta)$. If $\mathcal{A}$ performs this query, first of all, a coin b will be flipped. If $b = 0$, a session key $SK_s^{'}$ will be randomly chosen and then sent to $\mathcal{A}$. If $b = 1$, then the session key SK_s is returned to $\mathcal{A}$. It is computationally indistinguishable between the random $SK_s^{'}$ and the real SK_s for $\mathcal{A}$.

Definition 1. If adversary $\mathcal{A}$ can perform $Execute(\Delta)$, $Send(\Omega_U^i, Start)$, $Send(\Delta, M)$, $Corrupt(\Delta, b)$, $Hash(\cdot)$, and $Test(\Delta)$ queries in polynomial time under the ROM model, then she can break the session key security of the proposed protocol with advantage $Adv_{\mathcal{A}}^{Suc}$ as follows:

$$Adv_{\mathcal{A}}^{Suc} \leq \frac{q_{H_1}^2}{2^{L_1-2}} + \frac{(q_S+q_{Exe})^2}{N_0} + \frac{q_S}{2^{L_1-2}} + 2C^{'}q_S^{S^{'}} + \frac{2N\lambda_{Ans_i}}{G_{Ans_i}} + \frac{q_{H_2}^2}{2^{L_2-1}} + \frac{q_{H_1}^3}{2^{L_3-2}}.$$

Proof. A sequence of games $GAME_i$ is defined to prove the security of MFA-DPRU. $\mathcal{A}$ wins the games if and only if she guesses b in each game $GAME_i$ successfully. The proof is presented as follows.

$\boldsymbol{GAME_1}$: This game simulates the real word attacks without performing any queries under the ROM model. Therefore, the advantage of $\mathcal{A}$ breaking the scheme is $Adv_{\mathcal{A}}^{Suc}(t) = |2Pr[Suc_{\mathcal{A}}^{G_1}] - 1|$.

$\boldsymbol{GAME_2}$: This game is identical to $GAME_1$ except that it can perform $Execute(\Delta)$ queries. However, only messages $\{QR_i,\ CAK_i,\ M_1,\ M_2,\ T_1,\ M_3,\ M_4,\ M_5,\ MID_i,$

This work was supported by the Natural Science Foundation of Tianjin, China, under Grant 21JCZXJC00100 and by the National Natural Science Foundation of China under Grant 62222208, and in part by the Key Laboratory of Network Cryptography Technology under Grant LNCT2022-A02. (Corresponding author: Ding Wang.)

The authors are with the College of Cyber Science, Nankai University, Tianjin 300350, China, and also with Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450001, China, and also with Tianjin Key Laboratory of Network and Data Security Technology, Nankai University, Tianjin 300350, China. (Email: zhuliufu@mail.nankai.edu.cn; wangding@nankai.edu.cn;).

T_2, M_6, M_7, T_3, M_8, M_9, M_{10}, T_4} can be provided to $\mathcal{A}$. Notice that, due to the application of random values r_u, r_g, and r_s, it is difficult to guess b efficiently. The advantage of $\mathcal{A}$ to break the scheme is the same as $GAME_1$, that is, $Pr[Suc_{\mathcal{A}}^{G_1}] = Pr[Suc_{\mathcal{A}}^{G_2}]$.

$\boldsymbol{GAME_3}$: This game is identical to $GAME_2$ except that it will terminate immediately if a collision exists in the $Hash(\cdot)$ queries for SK_s or a collision exists in transcripts. The probability of collision in $Hash(\cdot)$ queries is at most $\frac{q_{H_1}^2}{2^{L_1+1}}$, and the probability of collision in transcripts is at most $\frac{(q_S+q_{Exe})^2}{2N_0}$. The advantage of $\mathcal{A}$ in this game is

$$Pr[Suc_{\mathcal{A}}^{G_3}] - Pr[Suc_{\mathcal{A}}^{G_2}] \leq \frac{q_{H_1}^2}{2^{L_1+1}} + \frac{(q_S+q_{Exe})^2}{2N_0}.$$

$\boldsymbol{GAME_4}$: It is the same as $GAME_3$, except that it will abort if $\mathcal{A}$ passes authentication without performing $Hash(\cdot)$ queries, where $Pr[Suc_{\mathcal{A}}^{G_4}] - Pr[Suc_{\mathcal{A}}^{G_3}] \leq \frac{q_S}{2^{L_1}}$.

$\boldsymbol{GAME_5}$: This game outlines the security measures of the session key SK_s, covering two specific cases. In the first case, $\mathcal{A}$ is allowed to conduct $Corrupt(\Omega_{AGW}^k, 1)$ queries to possess x of AGW and attempts to break the forward security. Note that, although in possession of x, it is infeasible for $\mathcal{A}$ to establish the session key SK_s because r_u, r_g, and r_s are confidential. In the second case, $\mathcal{A}$ has in possession of only one of the above three random values to check whether or not the protocol can resist the KSSTI attack. It is difficult to calculate any two of the three values from the other. As a consequence, the only way to break the security of the session key for $\mathcal{A}$ is to perform $Send(\Omega_U^i, Start)$, $Send(\Delta, M)$, and $Hash(\cdot)$ queries in both cases. The advantage of $\mathcal{A}$ to break the security of session key is $Pr[Suc_{\mathcal{A}}^{G_5}] - Pr[Suc_{\mathcal{A}}^{G_4}] \leq \frac{q_S}{2^{L_1}} + \frac{q_{H_1}^2}{2^{L_1+1}}$.

$\boldsymbol{GAME_6}$: This game is equal to $GAME_5$, except that $\mathcal{A}$ attempts to perform password guessing attacks and private questions guessing attacks. Note that, the premise for making private questions guesses is that the smart card is lost and $\mathcal{A}$ can extract the private data. Firstly, it is required that if $Corrupt(\Omega_U^i, 2)$ is queried, $Corrupt(\Omega_U^i, 1)$ and $Corrupt(\Omega_U^i, 3)$ are forbidden to query. Through q_s queries of $Send(\Omega_U^i, Start)$ and $Send(\Delta, M)$, the probability of guessing the correct password is no more than $C^{'} \cdot q_S^{S^{'}}$, where $C^{'}$ and $S^{'}$ are constants related to the size of password space. Secondly, it is required that only smart cards SC can be obtained by $\mathcal{A}$. The advantage of $\mathcal{A}$ successfully guessing private questions when the adversary performs q_H hash query is $\frac{N\lambda_{Ans_i}}{G_{Ans_i}}$, where λ_{Ans_i} and G_{Ans_i} are constant related to the guessing entropy of the private question. Accordingly, the advantage of $\mathcal{A}$ in this game is

$$Pr[Suc_{\mathcal{A}}^{G_6}] - Pr[Suc_{\mathcal{A}}^{G_5}] \leq C^{'} \cdot q_S^{S^{'}} + \frac{N\lambda_{Ans_i}}{G_{Ans_i}} + \frac{q_{H_1}^2}{2^{L_1+1}} + \frac{q_{H_2}^2}{2^{L_2+1}}.$$

$\boldsymbol{GAME_7}$: This game is the same as $GAME_6$. However, in order to check whether the protocol can prevent impersonation attacks, it will terminate when $\mathcal{A}$ makes $Hash(\cdot)$ queries. The advantage of $\mathcal{A}$ satisfies

$$Pr[Suc_{\mathcal{A}}^{G_7}] - Pr[Suc_{\mathcal{A}}^{G_6}] \leq \frac{q_{H_1}^2}{2^{L_1+1}} + \frac{q_{H_2}^2}{2^{L_2+1}} + \frac{q_{H_3}^2}{2^{L_3+1}}.$$

From the analysis of $GAME_1$ to $GAME_7$, the advantage of $\mathcal{A}$ to break the proposed protocol can be summarized as

$$Adv_{\mathcal{A}}^{Suc}(t) = |2Pr[Succ_{\mathcal{A}}^{G_1}] - 1|$$

$$= 2Pr[Succ_{\mathcal{A}}^{G_6}] - 1 + 2(Pr[Succ_{\mathcal{A}}^{G_1}] - Pr[Succ_{\mathcal{A}}^{G_6}])$$
$$\leq 2 \times \frac{1}{2} - 1 + 2\sum_{i=1}^{6}(Pr[Succ_{\mathcal{A}}^{G_{i+1}}] - Pr[Succ_{\mathcal{A}}^{G_i}])$$
$$= 2 \times [4\frac{q_{H_1}^2}{2^{L_1+1}} + \frac{(q_S+q_{Exe})^2}{2N_0} + 2\frac{q_S}{2^L} + 2\frac{q_{H_2}^2}{2^{L_2+1}}$$
$$+ C^{'} \cdot q_S^{S^{'}} + \frac{q_{H_3}^2}{2^{L_3+1}} + \frac{N\lambda_{Ans_i}}{G_{Ans_i}}]$$
$$= \frac{q_{H_1}^2}{2^{L_1-2}} + \frac{(q_S+q_{Exe})^2}{N_0} + \frac{q_S}{2^{L_1-2}} + 2C^{'} q_S^{S^{'}} + \frac{2N\lambda_{Ans_i}}{G_{Ans_i}}$$
$$+ \frac{q_{H_2}^2}{2^{L_2-1}} + \frac{q_{H_1}^3}{2^{L_3-2}}.$$

APPENDIX B: ANALYSIS OF PASSWORD RECOVERY SECURITY

Theorem 2. Assuming the existence of an efficient adversary $\mathcal{A}$ who can extract $\{Que_i\}_{1\leq N}$, PWC and $\{\delta_i\}_{1\leq i\leq N}$ from smart card SC_i, the advantage of $\mathcal{A}$ successfully recovering the password is no more than $Adv_{\mathcal{A}}^{AES^{256}} + \frac{1}{2^{Nq}} + \frac{N\lambda_{Ans_i}}{G_{Ans_i}}$, where $Adv_{\mathcal{A}}^{AES^{256}}$ denotes the advantage of $\mathcal{A}$ successfully breaking the AES^{256} algorithm and $\frac{\lambda_{Ans_i}}{G_{Ans_i}}$ denotes the advantage of guessing the private question.

Proof. According to the definition, $f(x)$ is a random polynomial defined in the finite field F_q. (x_i, β_i) and $\delta_i = \beta_i \oplus H_1(H_2(Ans_i)\|H_2(ID_i) \mod n_a)$ are randomly distributed in F_q. Let $F_q[x]$ be a set of all polynomials define F_q and let $S_{\delta_i}[x]$ be a set of polynomials related to δ_i, where $S_{\delta_i}[x] \subseteq F_q[x]$. When achieving δ_i, the maximum advantage of $\mathcal{A}$ recovering the password depends on $S_{\delta_i}[x]$. Define the maximum advantage as follows:

$$Pr_{\mathcal{A}}^{Max}[\delta] = \frac{Pr[Ans_i]}{\sum_{Ans_i^{'} \leftrightarrow S_{\delta_i}[x]} Pr[Ans_i^{'}]},$$

where $Pr[Ans_i]$ represents the probability of $H_1(H_2(Ans_i) \| H_2(ID_i) \mod n_a)$ being inferred from $S_{\delta_i}[x]$. When viewing $H_1(\cdot)$ as a random oracle, $Pr[Ans_i]$ is uniformally distributed and different from $Pr[Ans_j]$ if $i \neq j$. Therefore,

$$Pr_{\mathcal{A}}^{Max}[\delta] \approx 1/|S_{\delta_i}[x]| \approx 1/|F_q[x]|$$

, which denotes that $\mathcal{A}$ successfully reconstructs the polynomial $f(x)$. According to the scheme, $f(x) = a_0 + a_1x_1 + a_2x_2^2 + ... + a_{N-1}x_{N-1}^{N-1}$ mod p is defined in F_q, where $a_i \in [0, p-1]$. If $\mathcal{A}$ reconstructs $f(x)$ successfully, it infers that a_i has been obtained. Above all, $Pr_{\mathcal{A}}^{Max}[\delta] \approx 1/2^{pN}$.

In order to analyze the maximum and minimum bound of $Pr[Ans_i]$, we define Min_i and Max_i as follows,

$$Min_i = Min_{Ans_i^{'} \leftrightarrow S_{\delta_i}} Pr[Ans_i]$$

$$Max_i = Max_{Ans_i^{'} \leftrightarrow S_{\delta_i}} Pr[Ans_i].$$

Then we get the following inequality $\sum Pr[Ans_i] \geq \frac{1}{|S_{\delta_i}[x]|} Min_{Ans_i' \leftrightarrow S_{\delta_i}[x]} Pr[Ans_i] \approx \frac{1}{2^{pN}} \prod_{i=1}^{N} Min_i$. Due to the independence of each question, $Pr_{\mathcal{A}}^{Max}[\delta] \leq \frac{1}{2^{pn}} \prod_{i=1}^{N} \frac{Max_i}{Min_i}$. Additionally, it is infeasible for $\mathcal{A}$ to distinguish $H_1(H_2(Ans_i) \parallel H_2(ID_i) \bmod n_a)$ and $H_1(H_2(Ans_j) \parallel H_2(ID_i) \bmod n_a)$ under the random oracle, which infers $Min_i \approx Max_i$. As a result, when achieving δ_i from smartcard, the maximum advantage $Pr_{\mathcal{A}}^{Max}[\delta]$ in recovering the password is less than $\frac{1}{2^{pN}}$.

If $\mathcal{A}$ fails to achieve any useful information from δ_i, she can only obtain PWC from the smart card. If breaking the AES^{256} algorithm successfully, the advantage in recovering the password is nearly $Adv_{\mathcal{A}}^{AES^{256}}$.

Additionally, due to the assumption that $\mathcal{A}$ can extract $\{Que_i\}_{1 \leq N}$ from the smart card SC_i, then she attempts to perform private questions attacks to retrieve the correct answers. The advantage of $\mathcal{A}$ successfully guessing privacy questions is $\frac{N\lambda_{Ans_i}}{G_{Ans_i}}$.

Above all, the advantage of $\mathcal{A}$ successfully recovering the password is no more than $Adv_{\mathcal{A}}^{AES^{256}} + \frac{1}{2^{Nq}} + \frac{N\lambda_{Ans_i}}{G_{Ans_i}}$.

Appendix C: Informal Analysis of Security Properties

With the incorporation of the fuzzy verifier technique, honeywords method, and public key algorithm, the proposed scheme is secure against known attacks, such as password guessing attacks, node capture attacks, impersonation attacks, and privileged insider attacks. Furthermore, it ensures forward security, dynamic password recovery, key rotation, and anonymity.

(1) Resist Password Guessing Attacks. Password guessing attacks evolve significantly over time. In the early stage, dictionary attacks are commonly used [1], [2], which involve creating a dictionary from common words and applying heuristic transformation rules to generate password variations for conducting attacks. Furthermore, password guessing attacks shift towards incorporating probabilistic statistical methods [3]–[5], improving effectiveness and success rate. More recently, password guessing attacks have advanced with the integration of machine learning and deep learning techniques [6]–[9]. For consistency, we use the term "password guessing attacks" throughout the paper.

An adversary $\mathcal{A}$ obtains a smart card SC_i and enables to extract information $\{PID_i, C_1, C_1 \oplus a_i, C_2, P_i, \theta_i, Upd_i, Gen(\cdot), Rep(\cdot)\}$. Assuming that the biometrics Bio_i have been exposed to $\mathcal{A}$, which enables $\mathcal{A}$ to derive σ_i by calculating $Rep(Bio_i, \theta_i)$. Upon the calculation of σ_i, $\mathcal{A}$ attempts to guess PW_i and ID_i. Due to the fact that fuzzy verifier technique is applied to compute $C_1 = H_1(\theta_i) \oplus H_1(H_2(MID_i) \oplus H_2(HPW_i) \, mod \, n_p)$, it implies that there exist almost $\frac{|ID_i| \cdot |PW_i|}{N_0} \approx 2^{32}$ candidate pairs of (ID_i, PW_i) to thwart $\mathcal{A}$, where $N_0 \approx 2^8$, $|ID_i| \approx 10^6$ and $|PW_i| \approx 10^6$ represent the size of identity space and password space respectively. It is computationally infeasible to pinpoint the correct identity ID_i and password PW_i from the candidates without interacting with AGW. Fortunately, the introduction of the honeywords method reports malicious login behaviors with an accuracy of $1 - \frac{1}{2^{80}}$. It is difficult for $\mathcal{A}$ to launch online/offline password guessing attacks.

(2) Resist Node Capture Attacks. Notice that $K_{gs} = H_1(SID_j \parallel Y_j) \oplus H_1(H_3(x) \parallel Y_j)$, where Y_j is chosen randomly in each session. When dealing with node capture attacks, the first situation shows that the compromised sensor node will not weaken the security of other sensor nodes. The second one demonstrates that even if the preshared session key $K_{gs}^{current}$ is exposed, it is difficult for $\mathcal{A}$ to recover the previous K_{gs}^{old}. Following the construction of the proposed scheme, $K_{gs} = H_1(SID_j \parallel Y_j) \oplus H_1(H_3(x) \parallel Y_j)$, $K_{gs}^{new} = K_{gs} \oplus H_1(H_3(x) \parallel Y_j) \oplus H_1(H_3(x^{new}) \parallel Y_j)$. SID_j and Y_j are randomly chosen and further protected using long-term secret key x, leading to the infeasibility of extracting SID_j and Y_j. It is difficult for $\mathcal{A}$ to recover K_{gs}^{old} from $K_{gs}^{current}$ because x or x^{new} is secret.

We consider another scenario where $\mathcal{A}$ compromises SN_j and obtains SID_j, K_{gs}, Y_j, it's difficult to rebuild the session key $SK_s^{'}$ established before (suppose that SN_j doesn't record SK_s after each communication). $\mathcal{A}$ intercepts $\{M_3^{'}, M_4^{'}, T_2^{'}\}$ through public channel and proceeds to calculate $r_u^{'} = M_3^{'} \oplus H_2(SID_j \parallel MID_i^{'} \parallel K_{gs} \parallel T_2)$, $r_g^{'} = M_4^{'} \oplus H_2(SID_j \parallel MID_i^{'} \parallel K_{gs} \parallel r_u^{'} \parallel T_2)$. However, it is infeasible to extract $r_s^{'}$ since it is transmitted in ciphertext$M_6 = (H_1(SID_j) \parallel r_s)^2 \, mod \, n$. As a result, $\mathcal{A}$ can't retrieve the previous session key even if she compromises the SN_j.

(3) Resist Impersonation Attacks. Suppose that $\mathcal{A}$ eavesdrops, intercepts and further modifies the messages to impersonate the legitimate party. There exist three cases to consider: **(1).**impersonating U_i; **(2).** impersonating SN_j; **(3).** impersonating AGW. In the first case, $\mathcal{A}$ intercepts $\{QR_i, CAK_i, M_1, M_2, T_1\}$ and attempts to impersonate U_i. Observing apparently, it is difficult for $\mathcal{A}$ to obtain MID_i without knowing y, z. Additionally, despite passing the authentication using the original messages, it is computationally infeasible to rebuild the session key without knowing the preshared key K_{gu}. The same reason holds for the second case, without the knowledge of the preshared key K_{gs}, $\mathcal{A}$ cannot impersonate SN_j. The last case is impossible due to the fact that $\mathcal{A}$ doesn't know x, y, z.

(4) Resist Privileged Insider Attacks. Suppose that an insider $\mathcal{A}$ obtains U_i's registration request $\{HPW_i, MID_i, Y_i, \theta_i\}$ from AGW. However, $\mathcal{A}$ is unable to recover preshared key K_{gu} without the knowledge of x. Furthermore, $Gen(\cdot)$ and $Rep(\cdot)$ are unavailable for AGW to further compute σ_i^* from Bio_i. The application of the fuzzy verifier technique makes it computationally infeasible for $\mathcal{A}$ to guess the targeted identity and password (ID_i, PW_i).

(5) Resist Replay Attacks. Due to the deployment of timestamps, it can prevent replay attacks. When $|T_i^{'} - T_i| > \Delta T$, this session will be aborted immediately. In addition, the proposed protocol is mutually authenticated among three parties, where **(1).** AGM checks the equation $M_2^* = M_2^{'}$ to authenticate U_i; **(2).** SN_j verifies the equation $M_5^* = M_5^{'}$ to authenticate AGW; **(3).** AGW checks $M_7^* =$

$M_7^{'}$ to authenticate SN_j; **(4).** Finally, U_i verifies $M_{10}^{*} = M_{10}^{'}$ to authenticate AGW. Failures occurring in any of the above steps will lead to a termination.

(6) Resist Known Session-Specific Temporary Information (KSSTI) Attacks. According to the reference [10], KSSTI attacks describe that $\mathcal{A}$ attempts to calculate the session key using random numbers like r_u, r_g, r_s. It is required that only one random value can be obtained by $\mathcal{A}$, that is, $\mathcal{A}$ is restricted to be a legitimate user, authentication gateway, or sensor node during each KSSIT attack. According to the construction of the scheme, the session key is calculated as

$$SK_s = H_1(MID_i||SID_j||r_u||r_g||r_s),$$

showing that it is computationally infeasible for $\mathcal{A}$ to rebuild the session key in the case where only one of the random values is exposed.

(7) Resist Clone Card Attacks. Besides the message $\{PID_i, C_1, C_1 \oplus a_i, C_2, P_i, Upd_i, \theta_i, Gen(\cdot), Rep(\cdot)\}$, $\mathcal{A}$ is in possession of the real identityID_i of U_i. Assume that the biometrics Bio_i is also exposed to $\mathcal{A}$, then she attempts to log in to the system by submitting candidate passwords $PW_i^{'}$. Since C_1 is calculated as

$$C_1^{*} = H_1(\theta_i) \oplus H_1(H_2(MID_i) \oplus H_2(HPW_i) \mod n_p),$$

it is still frustrating for $\mathcal{A}$ to further guess the correct password because there exist $\frac{|D|}{N_0} \approx 2^{12}$ candidate passwords. It is difficult for $\mathcal{A}$ to guess the correct password.

(8) Resist KCI Attacks. Assuming the leakage of the long-term secret key x, $\mathcal{A}$ attempts to impersonate the legitimate user. Due to $\{HPW_i, MID_i, Y_i\}$ are transmitted secretly, it's difficult to compute

$$K_{gu} = H_1(MID_i||Y_i) \oplus H_1(H_3(x)||Y_i).$$

Additionally, because of the security of z_g and y_g, $\mathcal{A}$ fails to retrieve $\{H_1(MID_i^{'}), r_u^{'}, T_1\}$ from $QR_i^{'}$. Therefore, $\mathcal{A}$ cannot impersonate a user using a KCI attack.

(9) Perfect Forward Security. Notice that, the session key is computed as $SK_s = H_1(MID_i \,||\, SID_j \,||\, r_u \,||\, r_g \,||\, r_s)$, where r_u, r_g, r_s are chosen randomly and transmitted securely. Although x is available to $\mathcal{A}$, it is still computationally infeasible for $\mathcal{A}$ to compute the SK_s because she cannot decrypt the ciphertext to obtain r_u, r_s without the knowledge of y, z. Moreover, when AGW performs the key rotation to achieve a new long-term secret key x^{new}, the previous key becomes invalid. Despite $x^{current}$ is exposed, there is no way to rebuild the previous session key.

(10) Dynamic Password Recovery and Update. When forgetting and leaking passwords, it enables users to recover passwords dynamically and then perform updating. Once injecting the smart card and extracting $\{Ques_i\}_{1\le i\le N}$, U_i answers these questinos and gets answers Ans_i, calculates $\beta_i = \delta_i \oplus H_1(H_2(Ans_i)\|H_2(ID_i) \mod n_a)$. $f(x)$ can be reconstructed as

$$f(x) = \sum_{i=1}^{N} \beta_i \prod_{1\le j\le N, j\ne i} (x - x_i)/(x_j - x_i) \mod p.$$

Then she computes $a_0 = f(0)$, and performs decryption algorithm $AES^{256}.Dec_{a_0}(PWC)$ to recover PW_i. After recovering the password, U_i can further perform updating. Note that the password can be recovered by correctly answering N private questions, and can be updated to enhance the security to avoid exposure risk. These N private questions $\{Ques_i\}_{1\le i\le N}$ can also be dynamically changed, $\{Ques_i\}_{1\le i\le N} \to \{Ques_i\}_{1\le i\le N}^{new}$ to resist private question attacks. Additionally, U_i can selects a new polynomial to reset a_0^{new} as follows

$$f(x)^{new} = a_0^{new} + a_1^{new}x_1 + ... + a_{N-1}^{new}x_{N-1}^{N-1} \mod p.$$

(11) Anonymity and Untraceability. MFA-DPRU provides anonymity for U_i and sensor node SN_j, respectively. It generates pseudo-identity MID_i for U_i, where $MID_i = H_1(ID_i \,||\, r_i)$ and then encrypts MID_i as $QR_i = (H_1(MID_i)||r_u||T_1)^2 \mod n$. It proceeds to compute

$$M_1 = SID_j \oplus H_1(K_{gu}^{*}||r_u||PID_i||T_1)$$

which can be sent in public.

(12) Key rotation. When the long-term secret key x is required to be rotated to enhance the security, AGW can choose a new secret key x^{new} randomly and computes

$$K_{gu}^{new} = K_{gu} \oplus H_1(H_3(x)||Y_i) \oplus H_1(H_3(x^{new})||Y_i),$$
$$K_{gs}^{new} = K_{gs} \oplus H_1(H_3(x)||Y_j) \oplus H_1(H_3(x^{new})||Y_j).$$

Due to the randomness of x^{new}, K_{gu}^{new} and K_{gs}^{new} are uniformly distributed and keep secret.

APPENDIX D: ANALYSIS BASED ON TAMARIN

```
rule (modulo AC) user_login:
   [
   Regist( <~PID, C1, C1a, C2, P, ~x2> ), Fr( ~ru ),
   !Own_User( ID, PW, BIO, ~IDs ), Fr( ~T1 )
   ]
  --[ Init_once( '3' ) ]->
   [
   Out( <
         mul(<h(XOR(XOR(P, h(~PID)), h(XOR(h(BIO), h(ID))))), ~ru, ~T1
             >,
             <h(XOR(XOR(P, h(~PID)), h(XOR(h(BIO), h(ID))))), ~ru, ~T1>),
         XOR(<
              XOR(C1a,
                  XOR(h(BIO),
                      h(XOR(h(XOR(XOR(P, h(~PID)), h(XOR(h(BIO), h(ID))))),
                            h(h(<PW, rep(BIO, ~x2)>))))),
              XOR(XOR(C2, h(~PID)),
                  h(XOR(h(XOR(XOR(P, h(~PID)), h(XOR(h(BIO), h(ID))))),
                        h(h(<PW, rep(BIO, ~x2)>))))
             >,
             h(<~ru, ~T1>)),
         XOR(~IDs,
             h(<
                XOR(XOR(C2, h(~PID)),
                    h(XOR(h(XOR(XOR(P, h(~PID)), h(XOR(h(BIO), h(ID))))),
                          h(h(<PW, rep(BIO, ~x2)>))))),
                ~ru, ~PID, ~T1>)),
         h(<~PID, XOR(XOR(P, h(~PID)), h(XOR(h(BIO), h(ID)))), ~IDs,
            ~ru, ~T1>),
         ~T1>
   ),
   Login( <~PID, C1, C1a, C2, P, ~x2>, ~ru, ~T1,
          XOR(XOR(P, h(~PID)), h(XOR(h(BIO), h(ID)))),
          XOR(XOR(C2, h(~PID)),
              h(XOR(h(XOR(XOR(P, h(~PID)), h(XOR(h(BIO), h(ID))))),
                    h(h(<PW, rep(BIO, ~x2)>)))))
   ),
   QR_get( h(XOR(XOR(P, h(~PID)), h(XOR(h(BIO), h(ID))))), ~ru, ~T1 )
   ]
```

Fig. 1: User login and authentication.

In this section, we model MFA-DPRU using Tamarin and then conduct an automated analysis.

```
rule (modulo AC) SN_verification:
   [
   In( <M3, M4,
        h(<~IDs, MID, Kgs, XOR(M3, h(<~IDs, MID, Kgs, ~T2>)),
           XOR(M4, h(<~IDs, MID, Kgs, ~T2>)), ~T2>),
        MID, ~T2>
   ),
   !Own_Server( ~x, ~yj, ~IDs, Kgs ), Fr( ~rs ), Fr( ~T3 )
   ]
  -->
   [
   Out( <mul(<h(~IDs), ~rs>, <h(~IDs), ~rs>),
         h(<MID, ~IDs, XOR(M3, h(<~IDs, MID, Kgs, ~T2>)),
            XOR(M4, h(<~IDs, MID, Kgs, ~T2>)), ~rs,
            h(<MID, ~IDs, XOR(M3, h(<~IDs, MID, Kgs, ~T2>)),
               XOR(M4, h(<~IDs, MID, Kgs, ~T2>)), ~rs>),
            ~T3>),
         ~T3>
   ),
   M6_get( h(~IDs), ~rs )
   ]
```

Fig. 2: Sensor node authentication.

```
rule (modulo AC) AGW_verification:
   [
   Regist( <~PID, C1, C1a, C2, P, ~x2> ),
   In( <QR, CAK, M1, h(<~PID, MID, ~IDs, ~ru, ~T1>), ~T1>
   ),
   !Own_AWG( ~x, ~ID, HPW, ~Yi, MID, ~IDs ),
   QR_get( h(MID), ~ru, ~T1 ), Fr( ~rg ), Fr( ~yj ), Fr( ~T2 )
   ]
  -->
   [
   Out( <
         XOR(~ru,
             h(<~IDs, MID, XOR(h(<~IDs, ~yj>), h(<h(~x), ~yj>)),
               ~T2>)),
         XOR(~rg,
             h(<~IDs, MID, XOR(h(<~IDs, ~yj>), h(<h(~x), ~yj>)),
               ~T2>)),
         h(<~IDs, MID, XOR(h(<~IDs, ~yj>), h(<h(~x), ~yj>)),
            ~ru, ~rg, ~T2>),
         MID, ~T2>
   ),
   AGW_verification( ~ru, ~rg,
                     XOR(h(<MID, ~Yi>), h(<h(~x), ~Yi>))
   )
   ]
```

Fig. 3: Gateway authentication.

(1) Tamarin modeling of MFA-DPRU. The main flows are modeled based on Tamarin 1.8.0 and presented in this part, including server authentication, user login and authentication, and gateway authentication.

Fig. 1 presents the user login and authentication process. The user inserts the smart card SC, inputs the password PW_i, identity ID_i and biometrics BIO_i, generates the authentication credential M_2, and sends it to the authentication gateway. Only providing correct authentication factors, she can log in and achieve SK_s.

Fig. 2 shows the authentication process of the sensor node. Only the sensor node with the correct key K_{gs} can generate a valid authentication credential M_7 and obtain the correct session key SK_s.

Fig. 3 shows the verification of the authentication gateway. Only the authentication gateway with the correct long-term key x can calculate the sensor node key K_{gs} and the user key K_{gu} and generate a valid credential M_5.

(2) Security analysis of MFA-DPRU. Based on Ubuntu 22. 04. 4, we conduct an automated analysis of MFA-DPRU using Tamarin on a microcomputer with Intel(R) Core(TM) i5-12500H 2.50 GHz processor. We focus on analyzing the forward security, session key security of the proposed scheme, and check whether the proposed scheme can resist password guessing attacks. The results shown in Fig. 4

Running TAMARIN 1.8.0

Proof scripts

```
theory Zhu begin

Message theory

Multiset rewriting rules and restrictions (12)

Tactic(s)

Raw sources (17 cases, deconstructions complete)

Refined sources (17 cases, deconstructions complete)

lemma security:
  all-traces
  "∀ SK #i.
         (SessKeyU( SK ) @ #i) ⇒
         ((¬(∃ #j. K( SK ) @ #j)) ∨ (∃ #r. Pw_reveal( ) @ #r))"
by sorry

lemma forward_security:
  all-traces
  "∀ SK #i.
         (SessKeyU( SK ) @ #i) ⇒
         ((¬(∃ #j. K( SK ) @ #j)) ∨
          (∃ #r. (Pw_reveal( ) @ #r) ∧ (#r < #i)))"
by sorry

lemma unique_key:
  all-traces
  "∀ SK #i #j.
         ((SessKeyU( SK ) @ #i) ∧ (SessKeyU( SK ) @ #j)) ⇒ (#i = #j)"
by sorry

lemma multi_factor_reveal_pw:
  all-traces
  "∀ SC #i #j.
         ((Create_SmartCard( SC ) @ #i) ∧ (K( SC ) @ #j)) ⇒
         (∃ #t. Reveal_SC( SC ) @ #t)"
by sorry

end
```

Fig. 4: Security analysis on Tamarin.

demonstrate that MFA-DPRU is secure against password guessing attacks, and further keeps session key security and forward security.

REFERENCES

[1] B. Pinkas and T. Sander, "Securing passwords against dictionary attacks," in *Proc. CCS 2002*, pp. 161–170.

[2] R. Canetti, S. Halevi, and M. Steiner, "Mitigating dictionary attacks on password-protected local storage," in *Proc. CRYPTO 2006*, pp. 160–179.

[3] A. Narayanan and V. Shmatikov, "Fast dictionary attacks on passwords using time-space tradeoff," in *Proc. CCS 2005*, pp. 364–372.

[4] J. Ma, W. Yang, M. Luo, and N. Li, "A study of probabilistic password models," in *Proc. S&P 2014*, pp. 689–704.

[5] R. Veras, C. Collins, and J. Thorpe, "On semantic patterns of passwords and their security impact." in *Proc. NDSS 2014*.

[6] D. Wang, Z. Zhang, P. Wang, J. Yan, and X. Huang, "Targeted online password guessing: An underestimated threat," in *Proc. CCS 2016*, pp. 1242–1254.

[7] B. Hitaj, P. Gasti, G. Ateniese, and F. Perez-Cruz, "Passgan: A deep learning approach for password guessing," in *Proc. ACNS 2019*, pp. 217–237.

[8] D. Wang, Y. Zou, Y.-A. Xiao, S. Ma, and X. Chen, "Pass2edit: A multi-step generative model for guessing edited passwords," in *Proc. USENIX 2023*, pp. 983–1000.

[9] D. Wang, Y. Zou, Z. Zhang, and K. Xiu, "Password guessing using random forest," in *Proc. USENIX 2023*, pp. 965–982.

[10] D. Abbasinezhad-Mood and M. Nikooghadam, "Design and hardware implementation of a security-enhanced elliptic curve cryptography based lightweight authentication scheme for smart grid communications," *Futur. Gener. Comp. Syst.*, vol. 84, pp. 47–57, 2018.