Understanding Ephemeral Secret Leakage Attacks in Password-Based Multi-Factor Authentication for Mobile Devices

Ding Wang, Meijia Xu, and Qingxuan Wang

***Abstract*—As the first defense for system security, multi-factor authentication has been deployed in various security-critical applications with mobile devices (e.g., smart grid, e-health, and Industrial Internet of Things). After three decades of intensive research, the question of how to design a secure multi-factor authentication protocol is still unsettled. In recent years, the ephemeral secret leakage (ESL) attack, originally used to cryptanalyze the authenticated key exchange (AKE) schemes, has been introduced into the field of multi-factor authentication. Considerable efforts have been made to resist the ESL attack, and it has also been listed as one of the common attacks that a secure multi-factor authentication scheme should resist. As one of the capabilities of an ESL attacker, she can obtain the ephemeral secret of public-key techniques adopted by multi-factor authentication schemes.**

However, public-key techniques have been proven indispensable for password-based protocols to resist offline password guessing attacks. Now a question arises: *Is it possible to build a secure password-based multi-factor authentication protocol resistant to ESL attacks and offline password guessing attacks?* This paper aims to answer this fundamental question. More specifically, we first revisit 161 password-based multi-factor authentication schemes involving ESL attacks, and present a comprehensive cryptanalysis of three representative protocols. Then, we reveal the relationship between ESL attacks and the failure of each representative protocol. Finally, we conduct a large-scale comparative measurement of 41 representative multi-factor authentication schemes. Comparison results show that *all* these authentication schemes considering ESL attacks do not perform better than those not. The above comprehensive approach leads to the key insight: ESL attacks are *unsuitable/unrealistic* for evaluating password-based multi-factor authentication schemes, because the leaked ephemeral secrets will unavoidably lead to offline password guessing attacks, i.e., the security of all these 161 schemes would be compromised. We further conclude that employing hardware-protected devices (e.g., smart cards) as possession-based authentication factors can resist ESL attacks, because the premise for ESL attackers to obtain ephemeral secrets no longer exists. We believe that our findings are general and also provide valuable guidance for defending against ESL attacks against multi-factor authentication protocols for non-mobile device environments as well.

***Index Terms*—Ephemeral secret leakage attacks, mobile devices, multi-factor authentication, password guessing attacks.**

I. Introduction

BACK in 1991, Chang and Wu [1] proposed a method in which a remote user logs into a server over a public network using both a password and a smart card, and agrees on a cryptographically secure session key with the server to protect their subsequent communications (see Fig. 1). Their work, which is regarded as a seminal work of multi-factor authentication (MFA), has been highly successful in enhancing authentication security, as it ensures that only the user who possesses all the required authentication factors can be authenticated. Nowadays, MFA has been the most widely used method for security-critical applications with mobile devices (e.g., smart grid [2], e-health [3], and Industrial Internet of Things [4]). However, constructing a secure multi-factor authentication scheme remains non-trivial. Its ultimate goal is to enforce true multi-factor security, whereby authentication is granted only if all authentication factors are satisfied [5]. Since Chang-Wu [1] proposed the first two-factor authentication scheme, hundreds of relevant schemes have been proposed (e.g., [6], [7], [8]). However, most MFA schemes fail to achieve that fundamental security goals and are soon broken. As a result, these schemes constitute a vicious circle of "break-fix-break-fix" [9]. Generally, the failed schemes may lack important security attributes, such as user anonymity [10], [11], [12] and forward secrecy [13], [14], [15], or cannot resist known attacks, such as offline password guessing [16], [17], [18], smart card loss [19], [20], [21], and node capture attacks [22], [23], [24].

These security attributes and resistance to known attacks constitute a series of fundamental security requirements. Researchers devote considerable effort to designing new schemes to fulfill these requirements. Specifically, Wang and Wang [25] prove that public-key techniques are indispensable for constructing anonymous multi-factor authentication schemes; Ma et al. [26] argue that at least two exponentiation (e.g., elliptic curve point multiplication) operations conducted on the server are necessary for achieving forward secrecy. Moreover, they propose

Received 7 March 2024; revised 21 November 2025; accepted 10 December 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62222208, in part by Tianjin Natural Science Foundation Project under Grant 25JCJQJC00230 and in part by the China Postdoctoral Science Foundation under Grant 2025M781477. *(Corresponding author: Qingxuan Wang.)*

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

Digital Object Identifier 10.1109/TDSC.2025.3646469

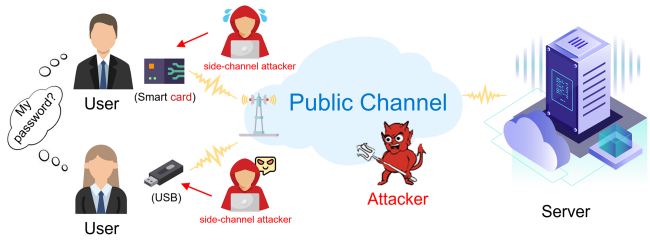

Fig. 1. Multi-factor authentication for mobile devices.

three general principles for designing secure password-based multi-factor authentication schemes, highlighting that public-key techniques are indispensable for resisting offline password guessing attacks; Wang et al. [27] systematize eight smart card loss attacks and provide in-depth understanding of how to avoid potential threats; For node capture attacks, Wang et al. [28] investigate their root causes, classify them into ten categories, and provide corresponding countermeasures for each.

In 2011, Islam and Biswas [29] pointed out that Chen et al.'s scheme [30] could not resist ephemeral secret leakage (ESL) attacks (also called session-specific information leakage attacks), *which originate from a widely accepted adversary model for public key infrastructure (PKI)-based authentication key exchange (AKE) protocols [31]*. This is the first time ESL attacks were used to *cryptanalyze a password-based MFA scheme*. Subsequently, dozens of studies on multi-factor authentication either list this attack as a common attack that should be resisted (see [13], [16], [19], [32], [33]), or use it to cryptanalyze other schemes [34], [35], [36], [37], [38]. However, unlike the aforementioned attacks, *an in-depth understanding of ESL attacks within the context of password-based multi-factor authentication is still lacking.*

A. Related Work

As early as 1993, the authenticated key exchange for distributed environments had already been central to computing practice, but it lacked satisfactory formal foundations. To formally define the significant authentication goals and prove that the candidate protocols meet these goals under standard cryptographic assumptions, Bellare and Rogaway [39] provided the first formal security definition for authenticated key exchange schemes. After that, Bellare et al. [40] pointed out that the previous work regarded key exchange as an independent problem, which was unsuitable for complicated authentication scenarios. In 2001, Canetti and Krawczyk [31] simplified the formal proof process and formed the famous CK model. In order to define the security of the protocols, the CK model provides a generic adversary model. Specifically, it first considers an adversary who fully controls the communication link and all message transmissions. In addition to the basic adversarial capabilities, it allows the attacker to obtain secret information from protocol participants through explicit attacks.

According to the type of information obtained by the adversary, the CK model classifies explicit attacks into three categories and simulates them as oracle queries, i.e., session-state reveal, session-key query, and party corruption. In a session-state reveal query, the adversary input the name of a party and a session identifier of an incomplete session. Then, the oracle outputs the internal state of the corresponding session. The session-state reveal query models the ESL attack, which is the first time this attack has been proposed. At that time, the ESL attack was more like a formalization, but Krawczyk [41] shows the realistic scenario that this attack is based on.

Specifically, to improve the performance of the protocol, many applications pre-compute the ephemeral secrets, such as the DH exponent x used to compute the public key g^x or the random bits used for a probabilistic encryption scheme. Compared with the long-term secret, this kind of information is more vulnerable to leakage because the former may be stored in a hardware-protected area while the latter is typically stored on disk (i.e., general memory devices) [41]. Furthermore, many similar incidents have occurred in recent years, such as the Heartbleed attack [42], the Meltdown attack [43], and the Spectre attack [44]. Accordingly, the concept of the ESL attack has been formed. However, the CK model does not provide a clear definition of what portion of the ephemeral secrets is allowed to be revealed, leaving this to be specified by the design of each protocol. LaMacchia et al. [45] pointed out that this unclear definition makes it difficult to determine the level of security that the protocol provides. As a result, they proposed an expanded CK (eCK) model and eliminated the ambiguity by defining the ephemeral secret key as all session-specific secret information.

ESL attacks are originally used to cryptanalyze AKE protocols [46], which are generally PKI-based. Many AKE protocols have been pointed out to be unable to resist this type of attacks [47], [48], [49]. As mentioned earlier, since Islam and Biswas [29] first introduced ESL attacks to cryptanalyze a password-based multi-factor authentication scheme in 2011, considerable efforts ([19], [21], [32], [38], [50]) have been invested in this research area. In these works, researchers exploit ESL attacks to analyse previous schemes [6], [51] or propose a new scheme, and claim that it can resist ESL attacks [13], [21], [52]. Generally, we divide the existing multi-factor authentication protocols involving ESL attacks into four categories: (1) Do not claim to be able to resist ESL attacks, and cannot resist ESL attacks because of poor design, e.g., only using the ephemeral secret to construct the session key [53]. (2) Claim to resist ESL attacks, but cannot achieve this goal (even under an attacker model without the ESL capability), due to ineffective ESL defensive measures (see examples [7], [54], [55]); (3) Claim to resist ESL attacks, and the defensive measure is effective, but suffer other known attacks under the adversary model that considers ESL attacks [8], [50]; (4) Claim to resist ESL attacks, but can only achieve this goal when an incomplete adversary model (e.g., the CK model) is adopted [6], [56].

B. Motivations

Unlike AKE protocols, the long-term secret in password authentication protocols is a user-generated password, essentially a short key chosen from a small space and easily guessed by an adversary [57], [58]. Therefore, resistance against offline password guessing is the primary security requirement of password-based

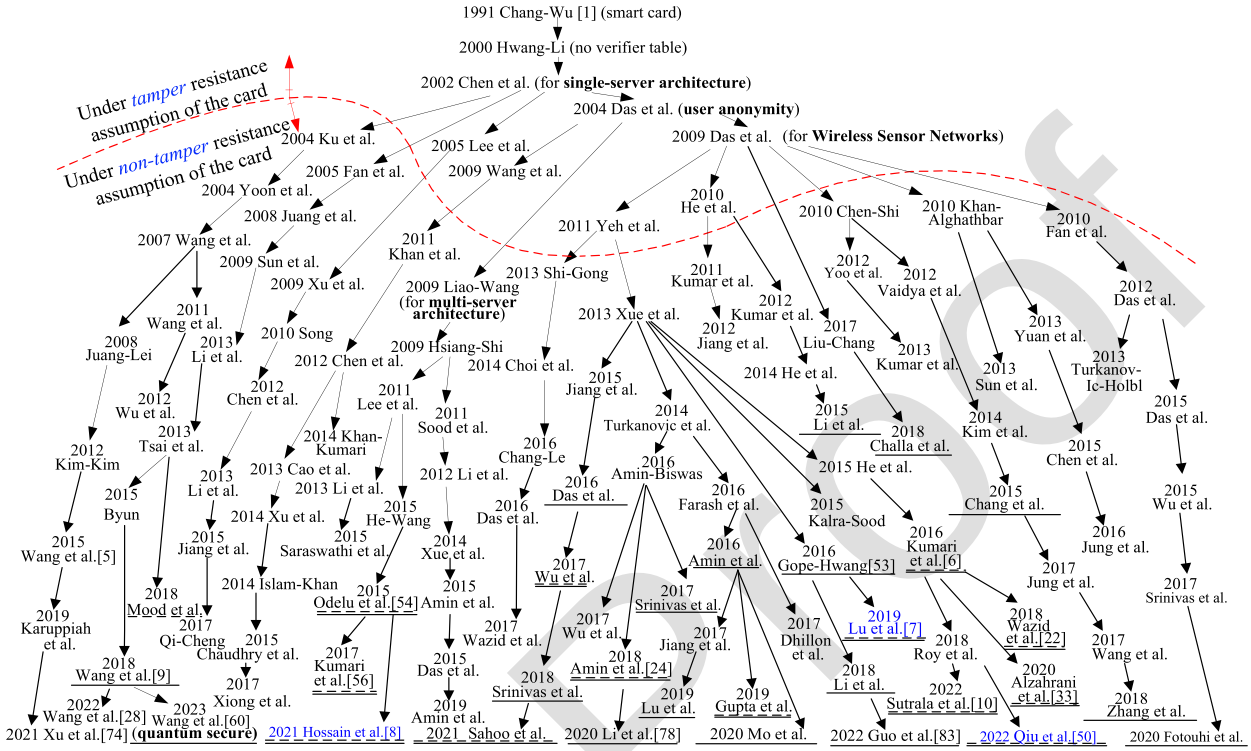

Fig. 2. A brief history of multi-factor authentication schemes. Each child node claims to be an improved scheme of its parent node but is found insecure by its child nodes. According to the analysis in the following text, all the listed schemes cannot resist ESL attacks, and schemes with a dashed line are claimed to be able to resist ephemeral secret leakage (ESL) attacks. Three schemes in blue are cryptanalyzed/exemplified in this work. There are 120 multi-factor authentication schemes listed in this figure, and for full references, see https://bit.ly/4p0mQHg.

protocols. Halevi and Krawczyk [59] studied the relationship between public-key cryptography and password protocols, and they proved that public-key techniques, serving as a one-way trapdoor function, are crucial for password protocols to resist offline guessing attacks.

Ma et al.'s public-key technique principle [26] shows that, under the non-tamper resistance assumption of smart cards, no smart card-based password authentication scheme can resist offline password guessing attacks without public-key techniques. The non-tamper resistance assumption of smart cards [9] (also known as the conditional non-tamper resistance assumption, see details in Sec. II-A) reveals the fact that the stored secrets will only be extracted by an adversary through side-channel attacks when the adversary somehow obtains the smart card for a relatively long time (e.g., a few hours). This assumption describes the realistic threats faced by possession-based authentication factors, so it is one of the basic security requirements for providing truly multi-factor security. As a result, this assumption is widely accepted in multi-factor authentication, and many related works have adopted it [28], [50], [60].

Besides, Wang-Wang [25] proved that public-key techniques are necessary to achieve user anonymity. These protocol design principles capture the intuition that in a password-based authentication scheme, public-key techniques (essentially the ephemeral secrets) are used to protect the user passwords and provide security properties for the protocol. However, as one of the capabilities of an ESL attacker, she can obtain ephemeral secrets, which eliminates the trapdoor protection provided by public-key cryptography, thereby making password protocols vulnerable to offline password guessing attacks. In this situation, a question naturally arises: *Is it possible to build a secure password-based multi-factor authentication protocol resistant to ESL attacks and offline password guessing attacks?*

Passwords and smart cards are the most commonly used authenticators for designing multi-factor authentication schemes. Velásquez et al. [61] investigate 442 multi-factor authentication protocols and find that 338 of them use passwords and smart cards as authenticators (including 270 for "password + smart card" and 68 for "password + smart card + biometric"). The protocols shown in Fig. 2 are all based on passwords and smart cards. Our taxonomy of multi-factor authentication schemes involving ESL attacks shows that such protocols are not secure when considering ESL adversaries. In this case, if the answer to the above question is negative, *it is necessary to prevent widespread misunderstandings caused by introducing unreasonable/unsuitable adversary models* in the security evaluation.

C. Contributions

In this work, our primary goal is to give a definite answer to the above question and clarify the potential long-standing misunderstanding between ESL attacks and password-based multi-factor authentication. Answering this question will facilitate the design of secure multi-factor authentication schemes. In summary, we make the following key contributions:

1) *Comprehensive cryptanalysis for representative protocols:* We revisit 161 (i.e., 120 schemes in Fig. 2 and 41 extra schemes in Table II) multi-factor authentication schemes involving ESL attacks and classify them into four categories. Among them, three categories of protocols claim to be able to resist ESL attacks. We select a representative protocol for each category and then conduct comprehensive cryptanalysis. The analysis shows that all these protocols fail to achieve the claimed security.
2) *A large-scale comparative evaluation:* To identify the security differences caused by considering ESL attacks or not in multi-factor authentication schemes, we employ two adversary models for these two situations and provide the formal security upper bound under each adversary model. Then, based on the widely accepted protocol evaluation criteria proposed by Wang and Wang [9], we conduct a large-scale comparative evaluation of 41 multi-factor authentication schemes. The comparison results show that when considering ESL attacks, even those protocols that are considered well-designed will become insecure.
3) *In-depth understanding of ESL attacks in multi-factor authentication schemes:* We investigate the relationship between ESL attacks and the failure of each representative protocol. Then, we point out the conflict between the realistic scenario on which ESL attacks are based and the conditional non-tamper resistance assumption of smart cards. Further, based on the original definition of ESL attacks in the CK model [31], and the definition of "strong AKE security" in the eCK model [45], we reveal that ESL attacks are unsuitable for evaluating the password-based multi-factor authentication.
4) *Some insights:* ESL attacks are unsuitable for eva- luating password-based multi-factor authentication schemes. In password-based MFA schemes, the leakage of ephemeral secrets will lead to the leakage of long-term secrets (i.e., passwords). In other words, an ESL attacker against password-based authentication schemes can lead to offline password guessing attacks. With the victims' long-term and ephemeral secrets, the adversary can completely compromise the security of any password-based MFA scheme. Moreover, the conditional non-tamper resistance smart card assumption reveals the fact that only when the smart card is held by an adversary for a relatively long time (e.g., a few hours) [9], the stored secrets will be obtained by her, which makes the premise of the ESL attack invalid. As a result, employing hardware-protected devices (e.g., smart cards) as possession-based authentication factors can naturally resist ESL attacks.

II. Preliminaries

In this section, we first elaborate on the conditional non-tamper resistance assumption of smart cards, then give two adversary models, i.e., the widely accepted adversary model for multi-factor authentication and the adversary model that considers ESL attacks. Finally, we introduce Wang and Wang's evaluation criteria [9].

A. Conditional Non-Tamper Resistance Assumption

Before 2008, most smart card-based MFA schemes were based on the assumption that adversaries cannot obtain the security parameters stored in smart cards (e.g., [62], [63]), that is, the smart card is tamper-resistant. However, with the development of side-channel attacks, the secret parameters stored in smart cards can be extracted by adversaries (e.g., [64], [65]). Therefore, to capture the adversary's realistic capabilities, the smart cards are no longer unconditionally secure. However, this does not mean that smart cards provide the same level of security as general memory devices in practice. Wang and Wang [66] point out the difference between smart cards and general memory devices.

The smart card is an integrated circuit card equipped with a processor that can execute cryptographic algorithms and a memory for storing data. In contrast, the general memory device is only a data storage device equipped with flash memory. Therefore, the smart card can be used in completely hostile environments, while the general memory device can only be used in trusted terminals. To reflect the security difference when building multi-factor authentication protocols based on smart cards and general storage devices, Wang and Wang [66] presented adversary models targeting these two different devices, respectively. When using the smart card, the adversary can get access to the smart card and compromise it by launching side-channel attacks, but this is *only* typically feasible when the smart card is in the adversary's hands for *a relatively long period of time* [9], [67]. When using general memory devices, the adversary can compromise them with malware in a few seconds. Furthermore, when using smart cards, the adversary cannot extract the secrets stored in them by using a malicious card reader because the user is on the scene, and she does not have enough time to launch side-channel attacks [66], [67]. This is the conditional non-tamper resistance assumption of smart cards.

B. Adversary Model

A realistic and concrete adversary model is necessary to assess the security of a cryptographic scheme. We present two concrete adversary models. The first model is widely used in multi-factor authentication, which follows the existing work [9], [28], and considers multi-factor security and the conditional non-tamper resistance assumption of smart cards. The second model further considers ESL attacks that are commonly considered in security models for evaluating AKE security (e.g., eCK [45].)

First, the capabilities of an adversary without ESL attacks (denoted as **A-I**) are described as follows.

A1. $\mathcal{A}$ can offline enumerate all items in the Cartesian product of identity and password space $\mathcal{D}_{id} \times \mathcal{D}_{pw}$ within polynomial time, and only when evaluating the scheme's security, can $\mathcal{A}$ get the user's identity.
A2. $\mathcal{A}$ has full control of the public channel, i.e., $\mathcal{A}$ can eavesdrop, intercept, and redirect messages transmitted among the communication participants.
A3. $\mathcal{A}$ may learn the password of the user via a malicious card reader, extract the secret data in the lost smart card by side-channel attacks, or attain a victim's biometric information using malicious devices. But the above cases

cannot be achieved at the same time, i.e., for an n-factor authentication scheme, the adversary can compromise $n-1$ factors at most, otherwise, it is a trivial case.

A4. $\mathcal{A}$ can learn server's long-term secret when assessing the system's forward security.

A5. $\mathcal{A}$ can register to be a legitimate user of the system or an administrator of the Server.

A6. $\mathcal{A}$ can learn the previous session keys.

In reality, passwords are human-chosen short keys [68], and identities are static texts with predefined structures, which enjoy little cryptographic strength [5]. Besides, both the identity space D_{id} and the password space D_{pw} are limited ($|D_{id}| \leq |D_{pw}| \approx 10^6$ [69]), therefore, the capability A1 is reasonable. The capability A2 shows the widely accepted Dolev-Yao model [70]. The capability A3 represents the design goal of "truly multi-factor security", which follows the harshest adversary model in [9]. According to the conditional non-tamper resistance assumption of smart cards, only when the smart card is under the adversary's control for a few hours can she obtain the stored parameters. Thus, $\mathcal{A}$ does not have enough time to extract any information from the smart card during the whole process, from the user entering the password to completing the authentication session with the server. The capability A5 indicates that $\mathcal{A}$ could be an insider attacker or a legitimate user.

Remark: Nowadays, NIST requires that users' passwords should be at least 15 characters in length, and all printing ASCII characters (95 printable characters in total) should be acceptable [71]. Therefore, intuitively, if the users' passwords follow a uniform distribution, the password space could be 95^8. However, previous research finds that user-generated passwords follow a heavily skewed Zipf distribution [57]. In general, 10^6 guesses are sufficient to successfully recover over 50% of a password dataset [69]. As a result, the effective space of the password is $|\mathcal{D}_{pw}| \approx 10^6$. Despite this, the password space size has little relevance to the protocol's security. A secure password authentication protocol should be able to resist offline password guessing attacks for any (small) size of password space [59]. Particularly, even if only two candidate passwords (i.e., $|\mathcal{D}_{pw}| = 2$) are given, the adversary should be unable to guess the correct password with a significantly higher probability than 1/2.

Besides capabilities A1∼A6 as described for the model **A-I**, the second model (denoted as **A-II**) further considers:

A7. $\mathcal{A}$ can obtain any subset of the four secret [45]: Party A's long-term secret, Party A's ephemeral secret, Party B's long-term secret and Party B's ephemeral secret, with the only restriction that $\mathcal{A}$ cannot obtain both the long-term and ephemeral secrets of *one party* at the same time.

After more than 30 years of continuous research in multi-factor authentication, resisting the **A-I** adversary has become a fundamental requirement in MFA protocol design. Therefore, when considering new attacks such as ESL attacks, we should maintain all the capabilities of the adversary in the **A-I** model. The capability A7 reflects ESL attacks, in AKE schemes, two parties exchange information and compute the shared session key by using the listed four kinds of secrets, where the ephemeral secrets contain all session-specific secret information of a party, namely ephemeral secret keys and random bits [45].

C. Formal Security Model

To formally capture the capabilities of the above two adversary models, we employ several queries to model the interactions between the adversary and honest parties. Specifically, the adversary **A-I** (i.e., capabilities A1∼A6) is considered as the basic requirement of MFA protocol design, which follows the BPR2000 security model [72] and Wang-Wang's modifications [9]. Besides, capability A7, which follows the eCK model [45], reflects ESL attacks. Our models are as follows.

Players: There are a number of participants in a multi-factor authentication protocol $\mathcal{P}$, namely, a user $U \in \mathcal{U}$ and a server $S \in \mathcal{S}$. Each of them has several instances called oracles. The user and server instances are denoted by U^i and S^j $(i, j \in \mathbb{Z})$, respectively, and the notation I is represented as any kind of such instances (i.e., $I \in \mathcal{U} \cup \mathcal{S}$).

Passwords and Long-term keys: Each user $U \in \mathcal{U}$ with identity ID_u owns a password PW_u which is considered to be drawn from a fixed, non-empty password dictionary $\mathcal{D}$ whose size is $|\mathcal{D}|$. As shown by Wang et al. [57], the password dictionary obeys Zipf's distribution. Besides, the server S stores some non-sensitive data and a few necessary public parameters as well as a long-term private key PRK.

Queries: The following queries model the adversary's capabilities in real-world attack scenarios.

-*Send*(I, m)*:* We use this query to model active attacks. When $\mathcal{A}$ calls this query, a message m will be sent to the instance I, which will process the message m according to the protocol $\mathcal{P}$. Then the instance I updates its state and gives the output to $\mathcal{A}$.

-*Execute*(U^i, S^j)*:* We use this query to model passive attacks. When the adversary $\mathcal{A}$ calls this query the protocol $\mathcal{P}$ will be executed completely and the messages that exchanged between U^i and S^j will be output.

-*Corrupt*(I, c)*:* When $\mathcal{A}$ calls this query, one of the three authentication factors of users will be output to $\mathcal{A}$.

- If $I = U, c = 1$, outputs all the security parameters that are stored in the smart card.
- If $I = U, c = 2$, outputs the password PW_U of U.
- If $I = U, c = 3$, outputs the biometric factors of U.

-*Reveal*(I, s)*:* We use this query to model ESL attacks. When the adversary $\mathcal{A}$ calls this query, one or two kinds of the secret will be returned to $\mathcal{A}$.

- If $I = U, s = 1$, outputs the long-term secret of U, i.e., the password PW_U and long-term nonces.
- If $I = U, s = 2$, outputs the ephemeral secret of U, i.e., the DH exponent x used to compute the public key g^x and all session-specific nonces.
- If $I = U, s = 3$, outputs the ephemeral secret of U, and the long-term secret of S.
- If $I = S, s = 1$, outputs the long-term secret of S, i.e., PRK and the data stored in S's database.
- If $I = S, s = 2$, outputs the ephemeral secret of S, i.e., the DH exponent y used to compute the public key g^y and all session-specific nonces.
- If $I = S, s = 3$, outputs the ephemeral secret of S, and the long-term secret of U.

-Test(I): This query is used to define session key's semantic security and can be called only once during the game. When

the adversary $\mathcal{A}$ calls this query a bit b will be flipped. If $b = 1$, then the actual session key sk of the instance I is returned to $\mathcal{A}$; otherwise, a random key of the same size is returned.

D. Evaluation Criteria

The evaluation criteria describe security requirements and ideal attributes that a well-designed multi-factor protocol should meet. As mentioned before, Wang and Wang proposed an evaluation set for two-factor authentication schemes [9], which is superior to previous works. Therefore, we employ their criteria as a building block. The evaluation criteria are listed below.

- C1. *No password verifier table:* The server doesn't maintain a database for storing the registered users' passwords or their relevant value.
- C2. *Password friendly:* The users could choose their password freely and change it at any time.
- C3. *No password exposure:* Even the administrator the of server cannot extract the users' passwords.
- C4. *No smart card loss attack:* The scheme is free from smart card loss attacks, i.e., if an attacker captures the user's lost smart card and extracts the secrets stored in the smart card, she cannot recover the user's password or impersonate the user by launching offline/online password guessing attacks.
- C5. *Resistance to known attacks:* The scheme can resist various attacks, such as replay attacks, offline password guessing attacks and de-synchronization attacks.
- C6. *Sound repairability:* Allowing the user to revoke her smart card without changing her identity.
- C7. *Provision of key agreement:* When completing the authentication phase, the user and the server share a common session key for subsequent communication.
- C8. *No clock synchronization:* The proposed scheme should avoid clock synchronization.
- C9. *Timely typo detection:* To reduce unnecessary communication cost, the user can be timely notified when she inputs a wrong password.
- C10. *Mutual authentication:* The user side and server side can authenticate each other.
- C11. *User anonymity:* The scheme should protect the user's identity and user's activities cannot be traced.
- C12. *Forward secrecy:* The leakage of long-term keys cannot affect the security of previous sessions.

III. Review of Lu et al.'s Scheme

This section briefly reviews the MFA scheme proposed by Lu et al. [7]. This scheme consists of three phases: registration, log-in and authentication, and update the password phase. To save space, we omit the update the password phase, which has little connection with our work. For ease of presentation, we employ some intuitive abbreviations listed in Table I.

TABLE I
Notations and Abbreviations

Symbol	Description	Symbol	Description
U_i	the i^{th} user	GWN	gateway node
HA	home agent	SC_i	U_i's smart card/device
$\mathcal{A}$	the adversary	CS_j	the j^{th} remote server
ID_i	identity of U_i	PW_i	password of U_i
TTP	trusted third party	SK	session key
SN	sensor node	$\mathsf{PK}_{\mathsf{TTP}}$	the public key of TTP
FA	foreign agent	$\mathsf{ID}_{\mathsf{TTP}}$	the identity of TTP
$\oplus$	bitwise XOR operation	$\|\|$	concatenation operation

A. Registration

HA selects a cyclic additive group $\mathcal{G}$ with order q on an elliptic curve $E_p(a, b) : y^2 = x^3 + ax + b \ (mod\ p)$ over a finite field $\mathbb{F}_p$, where $a, b \in \mathbb{F}_p$ and $4a^3 + 27b^2 \neq 0 \bmod p$, together with a point $\mathcal{O}$ at infinity on an elliptic curve and the generator P of $\mathcal{G}$. Then, HA selects a secret key $x \in \mathbb{Z}_q^*$, computes $\text{pub} = xP$. After that, HA chooses two one-way hash functions $h(\cdot)$ and $H(\cdot)$, a message authentication code function $MAC_{key}(\cdot)$, a symmetric encryption/decryption algorithm $Enc_{key}(\cdot)/Dec_{key}(\cdot)$. Finally, HA publishes $\{q, h(\cdot), H(\cdot), pub, \mathbb{F}_p, P, Enc_{key}(\cdot)/Dec_{key}, MAC_{key}(\cdot)\}$ and keeps x as secret.

R1: U_i selects her identity ID_i, password PW_i, and generates random numbers r_i. Then, U_i computes $PSW_i = h(PW_i, r_i)$, and submits the registration request $\{ID_i, PSW_i\}$ to HA through a private channel.

R2: On receiving the request, the home agent (HA) generates two random numbers x_1 and x_2. The HA records a set of data that contains $\{ID_i, Num_{i_m}\}$, where Num_{i_m} is a sequence number that represents mth registration for the user U_i. Next, HA computes $PID_i = h(h(ID_i, x_1), x_2)$, $K_{uh} = h(PID_i, x)$, $\alpha_h = Enc_{PSW_i}K_{uh}$, and $\beta_h = h(h(ID_i), PSW_i)$. Finally, HA issues a smart card, which includes parameters $\{\alpha_h, \beta_h, PID_i\}$, and transmits it to the user U_i via a secure channel. Also, HA stores K_{uh} into its database.

R3: After receiving the smart card, U_i enters the random number r_i into the smart card. Now, the smart card contains the parameters $\{\alpha_h, \beta_h, PID_i, r_i, h(\cdot), H(\cdot), P, q, \mathbb{F}_p, Enc_{key}/Dec_{key}, MAC_{key}\}$.

B. Login and Authentication

U_i logs into the system through the following steps:

L1: U_i inserts her smart card into a card reader and inputs her identity ID_i and password PW_i. After that, the smart card computes $PSW_i = h(PW_i, r_i)$ and checks if the equation $h(h(ID_i), h(PW_i, r_i)) = \beta_h$ holds. If the equation holds, the smart card generates a random nonce N_i and computes $K_{uh} = Dec_{PSW_i}(\alpha_h)$, $A_i = N_iP + H(K_{uh}, ID_i, ID_h)P$, $B_i = Enc_{K_{uh}}\ (ID_i, Ts_1, PID_i)$, and $C_i = MAC_{K_{uh}}(N_iP, ID_i, Ts_1)$. Then, MU sends the message $Ms_1 : \{A_i, B_i, C_i, PID_i, Ts_1\}$ to a foreign agent FA, where Ts_1 is the current timestamp at the U_i side.

L2: Upon receiving message from U_i, FA generates a present timestamp Ts_2 and checks if the equation $|Ts_2 - Ts_1| < \Delta T$ holds. If the equation holds, FA generates a random number N_f and a timestamp Ts_2, and then computes $A_f = N_fP + H(K_{fh}, ID_f, Ts_2)P$ and $B_f = MAC_{(N_fP)}(Ts_1, ID_h)$. After that, FA sends $Ms_2 : \{Ms_1, A_f, B_f, Ts_2\}$ to HA.

L3: Once receiving message from FA, HA verifies the freshness of received Ts_1 and Ts_2, and generates an immediate timestamp Ts_3. Then, HA decrypts B_i to derive U_i's identity ID_i using $K_{uh} = h(PID_i, x)$. Next, HA reads U_i's identity ID'_i from its database, and checks if the equation $ID'_M = ID_M$ holds. If it holds, HA continues to compute $N'_iP = A_i - H(K_{uh}ID_i, ID_h)P$ and verifies the integrity of C_i with K_{uh}. If it holds, HA computes $N'_fP = A_f - H(K_{fh}, ID_f, Ts_2)P$ and checks if the equation $MAC_{(N'_fP)_x}(ID_h, Ts_1) = B_f$ holds. If the equation holds, HA updates HA updates $K^*_{fh} = K_{fh} \oplus h(ID_f, N_fP, Ts_3)$ and $K^*_{uh} = K_{uh} \oplus h(ID_i, N_iP, Ts_3)$, and computes $A_h = N_iP + H(ID_i)P + H(K_{fh}, ID_h, N_fP)P$, $B_h = MAC_{K_{fh}}(N_fP, N_mP + H(ID_i)P, Ts_3)$, $C_h = N_fP + H(K_{uh}, ID_h, N_mP)P$, and $D_h = MAC_{K_{uh}}(ID_f, N_fP, Ts_3, PID_i)$. Finally, HA sends the message $Ms_3 : \{A_h, B_h, C_h, D_h, Ts_3\}$ to FA and updates the value Num_{i_m} to $Num_{i_{m+1}}$.

L4: When receiving the response from HA, FA first checks the freshness of Ts_3, and generates the current timestamp Ts_4. Next, FA computes $N_iP + H(ID_i)P = A_h - H(K_{fh}, ID_h, N_fP)P$ and checks the integrity of B_h. Then, FA computes $C_f = MAC_{(N_iP+H(ID_i)P)}(ID_f, N_fP, Ts_1, Ts_4, C_i)$ and the session key $SK = h(N_f(N_mP + H(ID_i)P))$. Finally, FA sends the message $Ms_4 : \{C_f, C_h, D_h, Ts_3, Ts_4\}$ to U_i.

L5: After receiving Ms_4, U_i checks if both the equations $|Ts_5 - Ts_3| < \Delta T$ and $|Ts_5 - Ts_4| < \Delta T$ hold. If the equations are hold, U_i computes $N_fP = C_h - H(K_{uh}, ID_h, N_iP)P$ and checks the integrity of D_h, and C_f. After that, U_i computes the session key $SK = h((N_i + H(ID_i))N_fP)$, $D_i = MAC_{((N_i+H(ID_i))P)}(C_f, N_fP)$ and sends the message $Ms_5 : \{D_i, Ts_5\}$ to the foreign agent FA through a public channel.

L6: FA examines whether the received timestamp Ts_5 is within a tolerable time interval. Next, FA authenticates U_i by verifying D_i. If so, the handshake between U_i and FA is finished with the help of HA. Finally, the established shared session key $SK = h(N_f(N_iP + H(ID_i)P)$ can be used to protect communication between U_i and FA.

IV. Analysis of Lu et al.'s Scheme

Lu et al. [7] propose this scheme aiming to improve the security of Gope and Hwang's scheme [53]. They point out that Gope and Hwang's scheme cannot resist ESL attacks and denial-of-service attacks. However, based on our analysis, we find that two kinds of offline password guessing attacks exist in Lu et al.'s scheme, and we point out that their countermeasures against ESL attacks are ineffective.

A. Offline Password Guessing Attack I

Based on the capability A3 of the adversary, an attacker $\mathcal{A}$ can obtain the security parameters stored in the smart card by side-channel attacks. Then, the offline password guessing attack can be launched as follows:

Step 1: $\mathcal{A}$ guesses U_i's identity ID^*_i and password PW^*_i from the dictionary space D_{id} and D_{pw}.

Step 2: $\mathcal{A}$ computes $PSW^*_i = h(PW^*_i, r_i)$, $h(ID^*_i)$, and $\beta^*_h = h(h(ID^*_i), PSW^*_i)$, where r_i is obtained from the smart cards. Then, $\mathcal{A}$ validates the correctness of (ID^*_i, PW^*_i) by comparing the calculated β^*_h and the extracted β_h.

Step 3: $\mathcal{A}$ will repeat the steps 1∼2 until she finds the correct pair of (ID^*_i, PW^*_i).

The time complexity of this attack is $\mathcal{O}(|D_{pw}| \times |D_{id}| \times 3T_h)$, where $|D_{pw}|$ and D_{id} are the size of password space D_{pw} and the size of identity space, respectively. T_h denotes the running time for hash operations. Further, according to capability A1, the user's identity enjoys little cryptographic strength. Thus, it can be directly obtained by a determined adversary. Therefore, the time complexity could be reduced to $\mathcal{O}(|D_{pw}| \times 3T_h)$, which is linear to $|D_{pw}|$. As mentioned earlier, the size of D_{pw} is limited (i.e., $|D_{pw}| \approx 10^6$ [69]) and the hash function is a lightweight operation. As a result, $\mathcal{A}$ could identify the correct password in polynomial time.

B. Offline Password Guessing Attack II

To change the password locally and avoid unnecessary communication costs incurred by entering wrong passwords, the HA stores the parameter β_h into the user's smart card, which leads to the above password guessing attack. This attack can be easily avoided by removing this parameter from the smart card or employing the "fuzzy-verifier" technique proposed in [9]. However, the next attack shows that it is possible for $\mathcal{A}$ to determine the victim's password without the parameter β_h.

Step 1: $\mathcal{A}$ guesses U_i's identity ID^*_i and password PW^*_i from the dictionary space D_{id} and D_{pw}.

Step 2: $\mathcal{A}$ computes $PSW^*_i = h(PW^*_i, r_i)$, where r_i is extracted from the smart card.

Step 3: $\mathcal{A}$ computes $K^*_{uh} = Dec_{PSW^*_i}(\alpha_h)$, where α_h is extracted from the smart card.

Step 4: $\mathcal{A}$ computes $B^*_i = Enc_{K^*_{uh}}(ID^*_i, Ts_1, PID_i)$, where Ts_1 and PID_i are obtained from previously intercepted transcripts. After that, $\mathcal{A}$ validates the correctness of (ID^*_i, PW^*_i) by comparing the calculated B^*_i and the B_i obtained from previously intercepted transcripts.

Step 5: $\mathcal{A}$ will repeat the steps 1∼4 until she finds the correct pair of (ID^*_i, PW^*_i).

The time complexity of this attack is $\mathcal{O}(|D_{pw}| \times |D_{id}| \times (T_h + 2T_S))$, where T_S denotes the running time for symmetric encryption operations.

C. No User Anonymity

The notion of user anonymity for authentication has two meanings [25]. Basically, anonymity means user identity protection, which protects the user's identity from being found out from protocol transcripts. Further, anonymity means user untraceability, that is, the adversary $\mathcal{A}$ cannot tell apart whether two sessions are initiated by the same user. Although Lu et al.'s scheme considers the basic level of

anonymity (i.e., use $PID_i = h(h(ID_i, x_1), x_2)$ to protect the user's identity), the adversary can still trace the user's behavior and threaten the user privacy. Specifically, the calculation of PID_i is fixed and directly related to the user U_i. As a result, the adversary $\mathcal{A}$ can track this fixed parameter PID_i to determine the interaction between U_i and HA. Therefore, Lu et al.'s scheme does not achieve user anonymity.

D. Discussion on ESL Attacks

Lu et al. claim that their scheme can resist ESL attacks. Specifically, they bind the random nonce N_i with the user's identity ID_i and compute the session key $SK = h((N_i + H(ID_i))N_f P)$. They argue that even both of the ephemeral secrets N_i and N_f are leaked to $\mathcal{A}$, without knowing the user's identity ID_i, $\mathcal{A}$ cannot compute the session key SK. However, based on our cryptanalysis of Lu et al.'s scheme, there are two ways for $\mathcal{A}$ to obtain the correct parameter ID_i. Thus, if $\mathcal{A}$ can obtain the ephemeral secrets N_i and N_f, she can compute the session key correctly.

As mentioned above, there are four categories of multi-factor authentication schemes involving ESL attacks. We classify Lu et al.'s scheme as the second category, i.e., "Claim to resist ESL attacks, but cannot achieve this goal (even under an attacker model without the ESL capability), due to ineffective ESL defensive measures. Actually, the design flaws of Lu et al.'s scheme are not fresh, and many works have pointed out similar failures [73], [74], [75]. Particularly, the design flaws in Lu et al.'s scheme, such as not resisting the offline password guessing attack II and no user anonymity, are essentially due to the absence of public-key cryptographic primitives. Therefore, the common fixed strategy is to use public key materials to construct the log-in message.

For example, suppose that the FA's public/private key pair of the Diffie-Hellman (DH) scheme is $(Pk_f = g^a, a)$, and when the user U_i registers, the FA's public key Pk_f has been stored into the U_i's smart card. After that, when U_i logs in, she first generates a random DH exponent b. Then, she computes the public key $Pk_i = g^b$ and the shared secret $ss = Pk_f^b$. Next, U_i computes $B_i = Enc_{k_{uh}}(ID_i, Ts_1, PID_i, ss)$, and sends B_i, Pk_i, and other parameters to FA. In this case, the adversary $\mathcal{A}$ cannot guess U_i's password because $\mathcal{A}$ does not know the shared secret ss. However, according to the capability A7 in the adversary model **A-II**, the DH exponents are considered ephemeral secrets that could be leaked to the adversary. Thus, countermeasures to prevent the offline password guessing attack II are ineffective for ESL attackers. $\mathcal{A}$ can still guess the password of the U_i and finally impersonate her identity to FA.

V. Review of Qiu et al.'s Scheme

This section briefly reviews the multi-factor authentication scheme proposed by Qiu et al. [50]. This scheme consists of five phases: system setup, registration, login and mutual authentication, password or biometrics update, and revocation phase. To save space, we omit the last two phases, which have little connection with our work.

A. System Setup Phase

Before system initialization, the server S selects a random number $s \in \mathbb{Z}_p^*$ and two one-way hash functions $H(\cdot)$ and $H_0(\cdot)$. After that, S computes the public key $T_S(x)$, publicizes the parameters $\{T_S(x), x, H(\cdot), H_0(\cdot)\}$, and finally keeps s as a long-term secret.

B. Registration Phase

In this phase, a user U_i gets registered with the S in a secure channel. The detailed steps are as follows:

R1: U_i selects her identity ID_i and sends it to S.

R2: On receiving the request, S selects the random numbers $e_i, r_i \in \mathbb{Z}_p^*$ and computes the parameters $CID_i = H(ID_i||e_i)$ and $B_0 = H(ID_i||s||r_i||CID_i)$. Then, S stores $\{ID_i, r_i, Honey\_list = NULL\}$ in its database. After that S inputs the parameters $\{T_s(x), x, CID_i\}$ to a new smart card SC_i. Finally, S sends $\{SC_i, B_0\}$ to U_i.

R3: After receiving the smart card SC_i, U_i inputs her password PW_i and biometrics BIO_i into SC_i. Then, SC_i generates a random number $2^4 \leq n_0 \leq 2^8$ and computes $GEN(BIO_i) = (\sigma_i, \tau_i)$, $VPW_i = H(PW_i||ID_i||CID_i||\sigma_i)$, $W_i = H((H(ID_i) \oplus VPW_i) \mod n_0)$, and $B_1 = B_0 \oplus VPW_i \oplus \sigma_i$. Finally, the smart card SC_i contains the following parameters: $\{CID_i, B_1, W_i, \tau_i, T_s(x), x, n_0\}$ and the functions $\{H(\cdot), H_0(\cdot), GEN(\cdot), REP(\cdot)\}$.

C. Login and Mutual Authentication Phase

After the user U_i finishes the registration phase, she logs into the system and establishes a shared session key with the server S through the following steps:

L1: U_i inserts her smart card SC_i into the card reader, and then inputs the identity ID_i, password PW_i, and biometrics BIO_i^*. After that, SC_i computes $\sigma_i = REP\ (BIO_i^*)$, $VPW_i = H(PW_i||ID_i||CID_i||\sigma_i)$ and checks if the equation $W_i = H((H(ID_i) \oplus VPW_i) \mathrm{mod} n_0)$ holds. If not, SC_i rejects the log-in request. Otherwise, SC_i computes $B_0 = B_1 \oplus VPW_i \sigma_i$. Next, SC_i selects a random number $u \in \mathbb{Z}_p^*$ and computes $C_1 = T_u(x)$, $C_2 = T_u(T_s(x))$, $C_3 = (ID_i||B_0) \oplus H_0(C_2)$, $C_4 = H(ID_i||CID_i||B_0||C_2||C_3)$. Finally, SC_i sends the message $\{CID_i, C_1, C_3, C_4\}$ to S.

L2: After receiving the message $\{CID_i, C_1, C_3, C_4\}$, S computes the parameters $C_2 = T_s(C_1)$ and $(ID_i'||B_0') = C_3 \oplus H_0(C_2)$. Then, S searches for ID_i' in its database, if S cannot find the ID_i' from the stored item $\{ID_i,\ r_i, Honey\_list\}$, S will terminate the session. Otherwise, S computes $B_0 = H(ID_i||s||r_i||CID_i)$ and checks whether the equation $C_4 = H(ID_i||CID_i||B_0||C_2||C_3)$ holds. If it holds, S continues to check whether the received B_0' equals the computed B_0. If $B_0' \neq B_0$, S inserts the B_0' into $Honey\_List$ and reject the log-in request. Moreover, if $|Honey\_List| \geq N_0$, where N_0 is a threshold value (e.g., $N_0 = 5$), S will suspend the U_i's smart card SC_i until she re-registers.

L3: If the above verification holds, S generates a random number $e_i^{new} \in \mathbb{Z}_p^*$ and computes $CID_i^{new} = H(ID_i||e_i^{new})$, $B_o^{new} = H(ID_i||s||r_i||CID_i^{new})$. Then, S updates $\{ID_i,$

$Honey\_List = Null\}$. After that, S selects a random number $v \in \mathbb{Z}_p^*$ and computes $C_5 = T_v(x)$, $C_6 = T_v(C_1)$, $C_7 = (CID_i^{new}||B_0^{new}) \oplus H(C_6||B_0)$, $SK_s = H(ID_i||CID_i||B_0||B_0^{new}||C_2||C_6)$, and $C_8 = H(ID_i||CID_i||B_0||C_2||C_5||SK_s)$. S sends $\{C_5, C_7, C_8\}$ to U_i over a public channel.

L4: After receiving the message $\{C_5, C_7, C_8\}$, SC_i first computes the following secure parameters $C_6 = T_u(C_5)$, $(CID_i^{new}||B_0^{new}) = C_7 \oplus H(C_6||B_0)$, and $SK_u = H\ (ID_i||CID_i||B_0||B_0^{new}||C_2||C_6)$. Then, U_i checks if the equation $C_8 = H(ID_i||CID_i||B_0||C_2||C_5||SK_u)$ holds. If not, U_i aborts this session. Otherwise, SC_i accepts the shared session key $SK_u = SK_s$ and uses it for future secure communications. After that, SC_i generates a new random number n_0^{new} $(2^4 \leq n_0^{new} \leq 2^8)$ and computes $VPW_i^{new} = H(PW_i||ID_i||CID_i^{new}||\sigma_i)$, $W_i^{new} = H\ ((H(ID_i) \oplus VPW_i^{new}) \text{mod} n_0^{new})$, and $B_1^{new} = B_0^{new} \oplus VPW_i^{new} \oplus \sigma_i$. Finally, the smart card SC_i replace $\{CID_i, W_i, B_1, n_0\}$ with $\{CID_i^{new}, W_i^{new}, B_1^{new}, n_0^{new}\}$.

VI. Analysis of Qiu et al.'s Scheme

Qiu et al. [50] propose this authentication scheme to protect the communication between a resource-constrained user device and the remote server. Qiu et al. state that their scheme can resist the session-specific ephemeral information leakage attacks (i.e., ESL attacks). However, under the adversary model **A-II**, we find that their scheme cannot resist the offline password guessing attack and the user impersonation attack, and we point out that their countermeasures against ESL attacks are ineffective.

A. Offline Password Guessing Attack

Based on the capability A3 of the adversary, an attacker $\mathcal{A}$ can obtain the victim's biometrics BIO_i^* by using malicious devices and the security parameters stored in the SC_i by side-channel attacks. Besides, based on the capability A7 of the adversary, $\mathcal{A}$ can also obtain the ephemeral secrets u, v, e_i^{new}. Then, the offline password guessing attack can be launched as follows:

- *Step 1:* $\mathcal{A}$ computes $C_2 = T_u(T_s(x))$, where $T_s(x)$ is extracted from the user U_i's smart card SC_i. Then, $\mathcal{A}$ computes $(ID_i||B_0) = C_3 \oplus H_0(C_2)$, where C_3 is obtained from the transmitted message $\{CID_i, C_1, C_3, C_4\}$.
- *Step 2:* $\mathcal{A}$ computes the parameters $C_6 = T_v(C_1)$ and $(CID_i^{new}||B_0^{new}) = C_7 \oplus H(C_6||B_0)$, where C_7 is obtained from the message $\{C_5, C_7, C_8\}$.
- *Step 3:* $\mathcal{A}$ computes the parameters $\sigma_i = REP(BIO_i^*, \tau_i)$ and $VPW_i^{new} = B_1^{new} \oplus B_0^{new} \oplus \sigma_i$, where τ_i and B_1^{new} are extracted from SC_i.
- *Step 4:* $\mathcal{A}$ guesses U_i's identity ID_i^* and password PW_i^* from the dictionary space D_{id} and D_{pw}, and then, $\mathcal{A}$ computes the verification parameter $VPW_i^{new*} = H(PW_i^*||ID_i^*||CID_i^{new}||\sigma_i)$.
- *Step 5:* $\mathcal{A}$ validates the correctness of the guessed pair (ID_i^*, PW_i^*) by checking whether the equation $VPW_i^{new*} = VPW_i^{new}$ holds.
- *Step 6:* $\mathcal{A}$ will repeat the steps 4∼5 until she finds the correct pair of (ID_i^*, PW_i^*).

The time complexity of this attack is $\mathcal{O}(|D_{pw}| \times |D_{id}| \times T_h)$. Although this attack consists of six steps, the calculations in step 1 to step 3 only need to be performed once. Therefore, their computational cost can be omitted.

B. User Impersonation Attack

Based on the capability A3 of the adversary, an attacker $\mathcal{A}$ can obtain the victim's biometrics BIO_i^* by using malicious devices. Also, based on the capability A7 of the adversary, $\mathcal{A}$ can also obtain the ephemeral secrets u, v, e_i^{new}. Then, $\mathcal{A}$ can impersonate U_i as follows:

- *Step 1:* $\mathcal{A}$ computes $C_2 = T_u(T_s(x))$, where $T_s(x)$ is the server S's public parameter. Then, $\mathcal{A}$ computes $(ID_i||B_0) = C_3 \oplus H_0(C_2)$, where C_3 is obtained from the message $\{CID_i, C_1, C_3, C_4\}$.
- *Step 2:* $\mathcal{A}$ computes the parameters $C_6 = T_v(C_1)$ and $(CID_i^{new}||B_0^{new}) = C_7 \oplus H(C_6||B_0)$, where C_7 is obtained from the message $\{C_5, C_7, C_8\}$.
- *Step 3:* $\mathcal{A}$ selects a random number u' and computes the parameters $C_1' = T_{u'}(x)$, $C_2' = T_{u'}(T_s(x))$, $C_3' = (ID_i||B_0^{new}) \oplus H_0(C_2')$ and the verification parameter $C_4' = H(ID_i||CID_i^{new}||B_0^{new}||C_2'||C_3')$.
- *Step 4:* $\mathcal{A}$ sends the forged log-in message $\{CID_i^{new}, C_1', C_3', C_4'\}$ to the server S.

In this attack, an adversary $\mathcal{A}$ can impersonate the user U_i even without extracting the secret parameters from the victim's smart card SC_i. Since the verification parameter B_0^{new} is computed from C_7 sent by S, $\mathcal{A}$ can pass the verification of S.

C. Discussion on ESL Attacks

Qiu et al.'s scheme employs the public key technique chaotic-map to construct the log-in message and it will update the stored parameters $\{CID_i^{new}, W_i^{new}, B_1^{new}, n_0^{new}\}$ after completing each authentication session. Thus, under the adversary model **A-I**, Qiu et al.'s scheme is well-designed and can resist common known attacks. However, Qiu et al. claim that their scheme can resist ESL attacks. Specifically, they first assume that all of the ephemeral secrets $\{u, v, e_i\}$ are leaked to the adversary $\mathcal{A}$. Then, they argue that it is difficult for $\mathcal{A}$ to compute the session key $SK = H(ID_i||CID_i||B_0||B_0^{new}||C_2||C_6)$ without correct secrets B_0 and B_0^{new}. However, according to our cryptanalysis of Qiu et al.'s scheme, $\mathcal{A}$ can obtain B_0 and B_0^{new}. As a result, Qiu et al.'s scheme resist ESL attacks.

As mentioned above, there are four categories of multi-factor authentication schemes involving ESL attacks. We classify Qiu et al.'s scheme as the third category, i.e., "Claim to resist ESL attacks, and the defensive measure is effective, but suffer other known attacks under the adversary model that considers ESL attacks". Such protocols imply a paradox: After introducing ESL attacks, a protocol suffers from known attacks that would otherwise be resistant. Besides, Kumari et al. [6] point out the countermeasure adopted by those schemes claiming to resist ESL attacks. Specifically, the session keys should rely on fixed parameters (long-term secrets) and ephemeral secrets. However, in a password-based multi-factor authentication scheme, the user's long-term secret is an easily enumerated password (See details in [57], [76], [77]) protected by ephemeral secrets. Thus, this countermeasure is ineffective for password-based multi-factor authentication schemes.

VII. Review of Hossain et al.'s Scheme

In this section, we briefly review the multi-factor authentication scheme for remote servers proposed by Hossain et al. [8], which consists of four phases: setup, registration, authentication, and renewing phase. To save space, we only show the registration and authentication phase.

A. Registration Phase

The registration phase includes the user registration phase and the server registration phase. First, the detailed steps of the user registration phase are as follows:

UR1: U_i chooses her identity ID_i, password PW_i and a random number b_i. Then, the user computes $B_i = H(g_i)$, where g_i is the user's biometrics. After that, U_i computes $BFU_i = h(b_i||PW_i||B_i)$ and sends the message $\{ID_i, BFU_i\}$ to TTP via a secure channel.

UR2: On receiving $\{ID_i, BFU_i\}$ at time T, TTP first checks whether the identity exists in its database. If it does not exist, TTP chooses a random number α_i and computes $A_i = h(h(ID_i) \oplus BFU_i) \bmod p_0$ where p_0 is the medium integer $2^4 \leq p_0 \leq 2^8$ and stores $\{ID_i, T_{\text{reg}_i}, T$, HoneyList $=$ Null $\}$ on its database. Otherwise, the trusted third party only updates $T_{reg_i} = T, \alpha_i = \alpha_i^{\text{new}}$, and HoneyList$=$ Null. Finally, TTP computes $N_i = BFU_i \oplus h(ID_i\|S_{TTP}\|T_{reg_i})$, stores $ID_i, h(ID_i\|S_{TTP}\|\ T_{reg_i}), \alpha_i$ on the database and sends $\{A_i, N_i, A_i \oplus \alpha_i, p_0\}$ to $\mathcal{U}_i$.

UR3: On receiving $\{A_i, N_i, A_i \oplus \alpha_i, p_0\}$, the user device securely stores $\{A_i, N_i, A_i \oplus \alpha_i, b_i, p_0\}$.

The detailed steps of server registration are as follows:

SR1: A remote server sends ID_{CS_j} to TTP for a registration request via a secure channel.

SR2: TTP computes $D_j = h(ID_{CS_j}||S_{TTP})$, stores $ID_{CS_j}, D_j\}$, and sends D_j to CS_j via a secure channel.

SR3: On receiving D_j, CS_j stores it on its storage.

B. Authentication Phase

Through this phase, with the help of TTP, CS_j authenticates the U_i's identity and establishes a shared session key between the user U_i and the remote server CS_j.

V1: U_i inputs her ID_i^*, PW_i^* and biometric $B_i^* = H(g_i)$. Next, the user device computes $BFU_i^* = h(b_i||PW_i^*||B_i^*), A_i^* = h(h(ID_i^*) \oplus BFU_i^*) \bmod p_0$ and checks whether $A_i = A_i^*$ holds to verify the validity of ID^*, PW^*, and B_i^*. If it holds, the user device computes $U_{SK_i} = h(ID_i||S_{TTP}||T_{reg_i}) = N_i \oplus BFU_i$. Then U_i generates a random number a, and computes $E_{uil} = aP$, $\alpha_i^* = A_i \oplus \alpha_i \oplus A_i^*$. After that, U_i computes $d_i = h_1(E_{uil}, ID_i, ID_{TTP}, ID_{CS_j})$, $\overline{E_{uil}} = (U_{SK_i} + ad_i)P$ and $PS_{uil} = (U_{SK_i} + ad_i)PK_{TTP}$, where PK_{TTP} is the public key of TTP. Finally, U_i computes the short term secret key pair $(k_1, k_2) = h_2(PS_{uil}, \overline{E_{uil}}\|ID_{CS_j}\|\ ID_{TTP})$. Then, U_i encrypts the secret parameters using the generated key k_1, $C_{ULT} = Enc_{k_1}(ID_i||E_{uil}||U_{SK_i}||\ \alpha_i^*)$, and sends the message $M_1 = \{\overline{E_{uil}}, C_{ULT}\}$ to CS_j.

V2: The remote server selects a random number b, computes $E_{slt} = bP$, $d_j = h_1(E_{slt}\|ID_{CS_j}\|ID_{TTP})$, $\overline{E_{slt}} = (D_j + bd_j)P$, and $PS_{slt} = (D_j + bd_j)PK_{TTP}$. Then, CS_j computes the a pair of keys $(k_3, k_4) = h_2(PS_{slt}, \overline{E_{slt}}\|ID_{TTP}\|\overline{E_{uil}}\|C_{ULT})$, uses the generated key to encrypt the credentials $C_{SLT} = Enc_{k_3}(ID_{CS_j}\|E_{slt})$, and sends $M_2 = \{\overline{E_{slt}}, C_{SLT}, \overline{E_{uil}}, C_{ULT}\}$ to TTP.

V3: On receiving M_2, TTP decrypts $PS_{slt} = S_{TTP}\overline{E_{slt}}$ by using the the secret key S_{TTP}. Then, TTP computes the short term key $(k_3, k_4) = h_2(PS_{slt}, \overline{E_{slt}}\|ID_{TTP}\|\overline{E_{uil}}\|C_{ULT})$. Next, TTP computes $(ID_{CS_j}\|E_{slt}) = Dec_{k_3}(C_{SLT})$, $d_j = h_1(E_{slt}, ID_{CS_j}, ID_{TTP})$, $\overline{E_{slt}^*} = D_jP + d_jE_{slt}$, $D_j = h(ID_{CS_j}||S_{TTP})$ and checks whether $\overline{E_{slt}^*} = \overline{E_{slt}}$. If it holds, TTP authenticates the CS_j. Otherwise, TTP ends the session. To verify the user, TTP computes $PS_{uil} = S_{TTP}\overline{E_{uil}}$, $(k_1, k_2) = h_2(PS_{uil}, \overline{E_{uil}}\|ID_{CS_j}\|ID_{TTP})$, and decrypts $(ID_i\|E_{uil}\|U_{SK_i}\|\ \alpha_i) = Dec_{k_1}(C_{ULT})$. TTP computes the parameters $d_i = h_1(E_{uil}, ID_i, ID_{TTP}, ID_{CS_j})$, $U_{SK_i}^* = h(S_{TTP}\|\ ID_i\|T_{reg_i})$, and checks if the equations $U_{SK_i}^* = U_{SK_i}$ and $\alpha_i^* = \alpha_i$ hold. If they hold, TTP computes $C_{TLS} = Enc_{k_4}(E_{uil}\|E_{slt}\|ID_i\|ID_{CS_j})$, $C_{TLU} = Enc_{k_2}(E_{slt}\|\ E_{uil}\|ID_i\|ID_{CS_j})$ and sends CS_j: $M_3 = \{C_{TLS}, C_{TLU}\}$.

V4: On receiving M_3, CS_j computes $(E_{uil}\|E_{slt}\|\ ID_{CS_j}\|ID_i) = Dec_{k_4}(C_{TLS})$ and checks if the ID_{CS_j} and E_{slt} hold. If they do not hold, CS_j rejects the session. Then CS_j computes the session key $K_{ji} = h(bE_{uil}\|ID_{CS_j}\|ID_i)$, ciphertext $C_{ji} = Enc_{K_{ji}}(ID_{TTP}\|\ E_{uil}\|ID_{CS_j}\|ID_i)$ and sends $M_4 = \{C_{ji}, C_{TLU}\}$ to U_i.

V5: U_i decrypts $(E_{slt}\|E_{uil}\|ID_i\|ID_{CS_j}) = Dec_{k_2}(C_{TLU})$ and checks whether the parameters ID_i and E_{uil} are valid. Then U_i computes $K_{ij} = h(aE_{slt}\|\ ID_{CS_j}\|ID_i)$, $(ID_{TTP}\|E_{uil}\|ID_{CS_j}\|ID_i) = Dec_{K_{ij}}\ (C_{ji})$ and checks whether the parameters ID_{TTP}, E_{uil}, and ID_i are valid. Finally, $\mathcal{U}_i$ computes $C_{ij} = Enc_{K_{ij}}\ (E_{uil} + E_{slt}\|ID_{CS_j}\|ID_i)$ and sends $M_5 = \{C_{ij}\}$ to CS_j.

V6: CS_j uses K_{ji} to decrypt $((E_{uil} + E_{slt})^*\|ID_{CS_j}\ \|ID_i) = Dec_{K_{ji}}(C_{ij})$. Then, CS_j checks whether $(E_{uil} + E_{slt})^* = E_{uil} + E_{slt}$ holds. If it holds, CS_j confirms that E_{uil} is fresh and agrees with the session key $\{K_{ij}\}$ (or K_{ji}).

VIII. Analysis of Hossain et al.'s Scheme

Hossain et al. [8] propose this authentication scheme to protect user security and privacy in remote server environments. They claim that their scheme is secure against various known attacks, such as man-in-the-middle and leakage-of-randomness attacks (i.e., ESL attacks). However, their scheme cannot resist man-in-the-middle attacks, replay attacks, and offline password guessing attacks. In addition, when considering the security of their scheme under the adversary **A-II**, it is also vulnerable to password-guessing attacks. It cannot resist ESL attacks.

A. Man-in-The-Middle Attack

Suppose an adversary $\mathcal{A}$'s public/private key pair is $(P_A = AP, A)$. When the user U_i tries to send the log-in request to the target remote server CS_j, $\mathcal{A}$ can claim that she is the TTP and deceives U_i to use her public key P_A to construct the log-in message. The attack can be launched as follows:

Step 1: U_i inputs her identity ID_i^*, password PW_i^*, and biometrics B_i^* into her device. After the verification, U_i generates a random number a, and computes $E_{uil} = aP, \alpha_i^* = A_i \oplus \alpha_i \oplus A_i^*$. Next, U_i computes $d_i = h_1(E_{uil}, ID_i, ID_{TTP}, ID_{CS_j})$, $\overline{E_{uil}} = (U_{SK_i} + ad_i)P$ and $PS'_{uil} = (U_{SK_i} + ad_i)P_A$, where P_A is the public key of $\mathcal{A}$. Finally, U_i computes the short-term secret key pair $(k'_1, k'_2) = h_2(PS'_{uil}, \overline{E'_{uil}} \| ID_{CS_j} \| ID_{TTP})$. Then, U_i computes $C_{ULT} = Enc_{k'_1}(ID_i || E_{uil} || U_{SK_i} || \alpha_i^*)$, and sends the message $M'_1 = \{\overline{E'_{uil}}, C'_{ULT}\}$ to CS_j

Step 2: $\mathcal{A}$ intercepts the message M_1 and computes $PS'_{uil} = A\overline{E'_{uil}}$, where A is the private key of $\mathcal{A}$. After that, $\mathcal{A}$ computes the key pair $(k'_1, k'_2) = h_2(PS'_{uil}, \overline{E'_{uil}} \| ID_{CS_j} \| ID_{TTP})$.

Step 3: $\mathcal{A}$ decrypts the U_i's ciphertext C'_{ULT} as $(ID_i \| E_{uil} \| U_{SK_i} \| \alpha_i^*) = Dec_{k'_1}(C'_{ULT})$.

Step 4: $\mathcal{A}$ can impersonate the U_i after obtaining the parameters U_{SK_i} and α_i^*.

The root cause of this attack is that U_i does not authenticate the identity of TTP before building the log-in message with TTP's public key. Since U_i does not save the TTP's public key in her device during the registration phase, the adversary $\mathcal{A}$ may easily deceive U_i by setting up phishing websites. Therefore, Hossain et al.'s scheme cannot resist the man-in-the-middle attack.

B. Replay Attack

According to the capability A5 of the adversary, suppose $\mathcal{A}$ is an administrator of the server CS_j, and she has the parameter D_j stored in CS_j's database. Then, $\mathcal{A}$ can obtain the U_i's critical parameter E_{uil} as follows:

Step 1: $\mathcal{A}$ eavesdrops on the public channel and obtains a victim U_v's message $M_1 = \{\overline{E_{uil}}, C_{ULT-v}\}$.

Step 2: $\mathcal{A}$ generate a random number b, computes $E_{slt} = bP, d_j = h_1(E_{slt} \| ID_{CS_j} \| ID_{TTP})$, $\overline{E_{slt}} = (D_j + bd_j)P$, and $PS_{slt} = (D_j + bd_j)PK_{TTP}$. Then, CS_j computes the a pair of keys $(k_3, k_4) = h_2(PS_{slt}, \overline{E_{slt}} \| ID_{TTP} \| \overline{E_{uil}} \| C_{ULT})$, uses the generated key to encrypt the credentials $C_{SLT} = Enc_{k_3}(ID_{CS_j} \| E_{slt})$, and sends $M_2 = \{\overline{E_{slt}}, C_{SLT}, \overline{E_{uil}}, C_{ULT-v}\}$ to TTP.

Step 3: Since every parameter in M_2 comes from a legal participant, it can pass the verification launched by TTP. After that, TTP will output $M_3 = \{C_{TLS}, C_{TLU-v}\}$ and send it to CS_j ($\mathcal{A}$).

Step 4: $\mathcal{A}$ decrypts the ciphertext C_{TLS} by computing $(E_{uil} \| E_{slt} \| ID_{CS_j} \| ID_i) = Dec_{k_4}(C_{TLS})$.

E_{uil} is an essential parameter in Hossain et al.'s scheme. According to the capability A3 of the adversary, when getting the U_v's device, the parameter E_{uil} could be used to guess U_v's password. The detailed steps of the attack are as follows.

C. Offline Password Guessing Attack

Based on the capability A3 of the adversary, $\mathcal{A}$ can obtain the parameters that are stored in the user's device and the U_i's biometrics g_i. Besides, Hossain et al. declare that their scheme can resist ESL attacks. Thus, based on the capability A7 of the adversary model **A-II**, $\mathcal{A}$ can obtain the U_i's ephemeral secret a. Then, $\mathcal{A}$ can guess the victim's password as follows:

Step 1: $\mathcal{A}$ first extracts $\overline{E_{uil}}$ from the transmitted message M_1. Then, $\mathcal{A}$ computes the parameters $E_{uil} = aP$, $\alpha_i = A_i \oplus a \oplus A_i$, and $B_i = H(g_i)$.

Step 2: $\mathcal{A}$ guesses U_i's identity ID_i^* and password PW_i^* from the dictionary space D_{id} and D_{pw}.

Step 3: $\mathcal{A}$ computes $BFU_i^* = h(b_i || PW_i^* || B_i)$, where b_i is extracted from the U_i's device. After that, $\mathcal{A}$ computes $U_{SK_i}^* = N_i \oplus BFU_i^*$ and $d_i^* = h_1(E_{uil}, ID_i^*, ID_{TTP}, ID_{CS_j})$.

Step 4: $\mathcal{A}$ computes $\overline{E_{uil}^*} = (U_{SK_i}^* + ad_i^*)P$, and then, validates the correctness of (ID_i^*, PW_i^*) by comparing the calculated $\overline{E_{uil}^*}$ and the $\overline{E_{uil}}$ obtained from previously intercepted transcripts.

Step 5: $\mathcal{A}$ will repeat the step 2∼4 until she finds the correct pair of (ID_i^*, PW_i^*).

The time complexity of this attack is $\mathcal{O}(|D_{pw}| \times |D_{id}| \times (2T_h + T_P))$, where T_P denotes the operation time for elliptic curve point multiplication. The compute cost of the $\oplus$ operation can be omitted. It should be noted that, $\overline{E_{uil}} = (U_{SK_i} + ad_i)P = U_{SK_i} \cdot P + d_i \cdot E_{uil}$, according to the Step 4, the parameter $\overline{E_{uil}}$ is essential to verify the guessed pair of (ID_i^*, PW_i^*). For an adversary who extracts the victim's device and her biometrics, only if she has the parameter E_{uil} can she launch the above offline password attack. This illustrates the harmfulness of the replay attack shown in Section VIII-B.

D. Discussion on ESL Attacks

Hossain et al. claim that their scheme can resist the leakage of randomness attack (essentially the ESL attack). Specifically, they first assume that, at most, one randomness of $\{a, b\}$ is leaked to the adversary $\mathcal{A}$. Then, they argue that it is difficult for $\mathcal{A}$ to forge a valid ciphertext C_{ULT} without the secret value U_{SK_i}. However, according to our cryptanalysis of Hossain et al.'s scheme, there are three ways for $\mathcal{A}$ to obtain the secret parameter U_{SK_i}. As a result, Hossain et al.'s scheme cannot achieve the claimed resistance to ESL attacks.

As mentioned above, there are four categories of multi-factor authentication schemes involving ESL attacks. We classify Hossain et al.'s scheme as the last category, i.e., "Claim to resist ESL attacks, but can only achieve this goal when an incomplete adversary model is adopted". Actually, the eCK model [45] defines that an ESL attacker can obtain any subset of the four kinds of secret (i.e., both the long-term and short-term secrets of two participants). But, Hossain et al. only allow the adversary to obtain at most one random nonce of $\{a, b\}$, which adopts an incomplete adversary model. Besides, the concept of the ESL attack is proposed to measure the design goals of "Session-Key Security" [31]. In other words, even if both the ephemeral secrets of two participants are leaked, the session key should still be secure. However, Hossain et al.'s scheme also fails to achieve this important design goal. Specifically, the session key established between the user U_i and the remote

server CS_j is calculated as $K_{ij} = h(a \cdot E_{slt}||ID_{CS_j}||ID_i) = h(a \cdot b \cdot P||ID_{CS_j}||ID_i) = K_{ji}$, according to the capability A1 of the adversary, ID_i and ID_{CS_j} can be obtained directly by the adversary $\mathcal{A}$. When holding ephemeral secrets a and b, ESL attacker can compute the session key.

IX. Comparative Evaluations

In the above sections, we have revealed the relationship between ESL attacks and the failure of each representative protocol. Since no representative protocol can resist offline password guessing attacks when ESL attackers are considered, our key conclusion, i.e., ESL attacks are unsuitable for evaluating password-based ~~multi-factor authentication~~ protocols, is empirically supported. To provide further evidence and to more clearly identify the security differences resulting from whether ESL attack are considered in MFA schemes, we conduct a large-scale comprehensive assessment of 41 MFA schemes listed in Table II, including both schemes that explicitly claim resistance to ESL attacks and those that do not. Our comparison is based on the two adversary models **A-I** (i.e., the adversary model widely accepted in MFA) and **A-II** (i.e., the adversary model that considers ESL attacks), as defined in Section II-B, as well as ~~and~~ Wang and Wang's 12 ~~multi-factor authentication~~ protocol design criteria [9] shown in Section II-D.

The selected protocols are recently proposed and representative typical schemes that have attracted significant attention, and some of them are considered well-designed (e.g., [9], [50], [78]). Resistance to offline password guessing attacks is the primary concern in password-based multi-factor authentication schemes. In this work, the inability to resist such attacks is also identified as the main reason for the failure of many schemes under ESL attackers. Therefore, we separate the offline password guessing attacks from C5: Resistance to known attacks and form an independent item $*$: Resistance to offline password guessing attacks. It can be seen from Table II that, under the adversary model **A-II**, no protocol can satisfy all of the evaluation criteria, and no protocol can resist offline password guessing attacks. More seriously, as expected, some protocols that perform well under the adversary model **A-I** are no longer secure when measured with the adversary model **A-II**. As a result, considering ESL attacks in multi-factor authentication not only cannot improve the security of each protocol but also affects the accuracy of the protocol evaluation results to a large extent.

X. Discussion

This section will provide a negative answer to the question: Is it possible to build a secure password-based MFA protocol resistant to ESL and offline password guessing attacks? By unveiling the contradiction between ESL attacks and the conditional non-tamper resistance assumption of smart cards, we show that the premise of ESL attacks becomes invalid when hardware-protected devices are employed as possession-based authentication factors.

A. Inapplicability of ESL Attacks to Password-Based MFA

As mentioned earlier, based on our review of a large number of multi-factor authentication schemes involving ESL attacks, we divide them into four categories. (1) Do not claim to be able to resist ESL attacks, and cannot resist ESL attacks because of poor design (e.g., only using the ephemeral secret to construct the session key); (2) Claim to resist ESL attacks, but cannot achieve this goal (even under an attacker model without the ESL capability), due to ineffective ESL defensive measures; (3) Claim to resist ESL attacks, and the defensive measure is effective, but suffer other known attacks under the adversary model that considers ESL attacks; (4) Claim to resist ESL attacks, but can only achieve this goal when an incomplete adversary model is adopted. According to the above classification, the existing multi-factor authentication protocols involving ESL attacks can be further classified into two types: (1) Those that do not claim to resist ESL attacks; (2) Those that claim to be resistant to ESL attacks but fail. In conclusion, these schemes are not secure when facing an ESL attacker.

To demonstrate this, we revisit three typical protocols and conduct comprehensive cryptanalysis under two adversary models (i.e., **A-I** and **A-II**), respectively. Specifically, Lu et al.'s scheme [7] represents the second category that "Claim to be able to resist ESL attacks, but cannot achieve this goal (even under an attacker model without the ESL capability), due to ineffective ESL defensive measures. According to our cryptanalysis, Lu et al.'s scheme cannot resist two kinds of offline password guessing attacks under the adversary model **A-I**. With the guessed password, the adversary can easily impersonate the victim. We give some countermeasures to prevent these attacks in Section IV-D, but they are ineffective under the adversary model **A-II**. The protocols represented by Lu et al. [7] show that it is already challenging to design a multi-factor authentication scheme that can satisfy basic security requirements, and resisting ESL attacks in such protocols would only be harder.

Qiu et al.'s scheme [50] represents the third category that "Claim to be able to resist ESL attacks, and the defensive measure is effective, but suffer other known attacks under the adversary model that considers ESL attacks. According to our cryptanalysis, Qiu et al.'s scheme is well-designed under the adversary model **A-I**. However, under the adversary model **A-II**, it cannot resist the offline password guessing attack and the user impersonation attack. The protocols represented by Qiu et al. indicate a potential contradiction between the basic security requirements of MFA and the resistance to ESL attacks.

Hossain et al.'s scheme [8] represents the last category that "Claim to be able to resist ESL attacks, but can only achieve this goal when an incomplete adversary model is adopted". According to our cryptanalysis, under the adversary model **A-I**, Hossain et al.'s scheme cannot resist the man-in-the-middle attack and the replay attack. While under the adversary model **A-II**, it cannot resist the offline password guessing attack. Besides, when measuring the ESL resistance, Hossain et al. only allow the adversary to obtain at most one ephemeral secret, which does not conform to the definition of ESL attacks in the eCK model. The protocols represented by Hossain et al. show that current

TABLE II
SECURITY AND EFFICIENCY COMPARISON AMONG RELEVANT USER AUTHENTICATION SCHEMES UNDER TWO ADVERSARY MODELS

Protocol	Year	Ref.	Adversary Model	Evaluation Criteria[2]												Computational Cost[1]		
				C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	*	User	GWN/Server	Sensor Node
Xu et al.	2025	[81]	A-I	√	√	√	√	√	√	√	√	√	√	√	√	√	$2T_P + 9T_H$	$T_P + 6T_H$	$2T_P + 4T_H$
			A-II	√	√	√	×	×	√	√	√	√	√	×	×	×			
Zhao et al.	2025	[82]	A-I	√	√	√	√	√	√	√	√	√	√	√	√	√	$T_{Ev} + T_{Pu} + 7T_H$	-	$2T_{Ev} + T_{Pu} + 5T_H$
			A-II	√	√	√	×	×	√	√	√	√	√	×	×	×			
Gui et al.†	2024	[32]	A-I	√	√	√	√	√	√	√	√	√	√	√	√	√	$3T_C + T_B + 7T_H$	$T_C + 8T_H$	$2T_C + 5T_H$
			A-II	√	√	√	×	×	√	√	√	√	√	×	×	×			
Wazid et al.†	2023	[13]	A-I	√	√	√	×	×	√	√	×	√	√	√	×	×	$T_B + 18T_H$	$7T_H$	$7T_H$
			A-II	√	√	√	×	×	√	√	×	√	√	√	×	×			
Qiu et al.†	2022	[50]	A-I	√	√	√	√	√	√	√	√	√	√	√	√	√	$10T_H + T_B + 3T_C$	$8T_H + 3T_C$	-
			A-II	√	√	√	×	×	√	√	√	√	√	×	×	×			
Sutrala et al.†	2022	[10]	A-I	√	√	√	×	×	√	√	×	√	√	×	√	×	$16T_H + T_B + 5T_P$	$9T_H + 3T_P$	$8T_H + 4T_P$
			A-II	√	√	√	×	×	√	√	×	√	√	×	×	×			
Shamshad et al.†	2022	[14]	A-I	√	√	√	×	×	√	√	√	√	√	√	×	×	$6T_H$	–	$3T_H$
			A-II	√	√	√	×	×	√	√	√	√	√	√	×	×			
Guo et al.†	2022	[11]	A-I	√	√	√	×	×	√	√	×	√	√	×	×	×	$28T_H$	$46T_H$	$22T_H$
			A-II	√	√	√	×	×	√	√	×	√	√	×	×	×			
Xia et al.†	2022	[16]	A-I	√	√	√	×	×	√	√	×	√	√	×	×	×	$T_B + 11T_H + T_S$	$10T_H + 4T_S$	$T_B + 2T_S + 6T_H$
			A-II	√	√	√	×	×	√	√	×	√	√	×	×	×			
Guo et al.	2022	[83]	A-I	√	√	√	×	×	√	√	×	√	√	√	×	×	$21T_H$	$18T_H$	$8T_H$
			A-II	√	√	√	×	×	√	√	×	√	√	×	×	×			
Hossain et al.†	2021	[8]	A-I	√	√	√	×	√	√	×	×	×	×	×	×	×	$7T_H + 3T_P + 4T_S$	$3T_H + 3T_P + 4T_S$	$5T_H + 4T_P + 4T_S$
			A-II	√	√	√	×	√	√	√	√	×	×	×	×	×			
Srinivas et al.	2021	[17]	A-I	√	√	√	×	×	√	√	×	√	×	×	×	×	$T_B + 17T_H + 6T_P$	$12T_H + 3T_P$	$8T_H + 4T_P$
			A-II	√	√	√	×	×	√	√	×	√	×	×	×	×			
Alzahrani et al.†	2021	[18]	A-I	√	√	√	×	×	√	√	×	√	√	√	√	×	$2T_P + T_S + 7T_H$	$T_P + 8T_H + 2T_S$	$5T_H + T_S + T_P$
			A-II	√	√	√	×	×	√	√	×	√	√	√	√	×			
Kumar et al.†	2021	[15]	A-I	√	√	√	×	×	√	√	√	√	√	√	×	×	$12T_H$	$9T_H$	$3T_H$
			A-II	√	√	√	×	×	√	√	√	√	√	√	×	×			
Zhang et al.†	2021	[19]	A-I	√	√	√	×	×	√	√	×	√	√	√	√	×	$9T_H + 3T_P + T_S$	$10T_H + 2T_P + T_S$	$3T_P + 7T_H$
			A-II	√	√	√	×	×	√	√	×	√	√	√	×	×			
Kumar et al.†	2021	[84]	A-I	√	√	√	√	×	√	√	×	√	√	√	×	√	$7T_H$	$7T_H$	$4T_H$
			A-II	√	√	√	×	×	√	√	×	√	√	√	×	×			
Chen et al.†	2021	[12]	A-I	√	√	√	×	×	√	√	×	√	√	×	√	×	$7T_H + 2T_P + T_B$	$10T_H + 3_P$	$5T_H + 4T_P$
			A-II	√	√	√	×	×	√	√	×	√	√	×	√	×			
Wu et al.	2021	[20]	A-I	√	√	√	×	×	×	√	√	√	√	×	×	×	$11T_H + T_B$	$14T_H$	$6T_H$
			A-II	√	√	√	×	×	×	√	√	√	√	×	×	×			
Lee et al.†	2021	[21]	A-I	√	√	√	×	×	√	√	×	√	√	√	×	×	$11T_H$	$13T_H$	$6T_H$
			A-II	√	√	√	×	×	√	√	×	√	√	√	×	×			
Qi et al.†	2021	[85]	A-I	√	√	√	×	×	√	√	×	√	√	×	×	-	$10T_H + T_S + T_B + T_C$	$11T_H + 2T_S$	$7T_H + T_S + 2T_C$
			A-II	√	√	√	×	×	√	√	×	√	√	×	×	-			
Wu et al.†	2021	[86]	A-I	√	√	√	×	×	√	√	√	√	√	×	×	×	$7T_H + T_S$	$6T_H + T_S$	$12T_H + 2T_S$
			A-II	√	√	√	×	×	√	√	√	√	√	×	×	×			
Li et al.	2020	[78]	A-I	√	√	√	√	√	√	√	√	√	√	√	√	√	$3T_P + 10T_H + T_B$	$T_P + 8T_H$	$2T_P + 4T_H$
			A-II	√	√	√	×	×	√	√	√	√	√	×	×	×			
Wazid et al.†	2020	[87]	A-I	√	√	√	×	×	√	√	×	√	√	√	×	×	$T_B + 13T_H$	$8T_H$	$8T_H$
			A-II	√	√	√	×	×	√	√	×	√	√	√	×	×			
Alzahrani et al.†	2020	[33]	A-I	√	√	√	×	×	×	√	×	√	√	√	√	×	$7T_H + 5T_P + 2T_M$	$5T_H + 6T_P + 2T_M$	$6T_H + 6T_P + 4T_M$
			A-II	√	√	√	×	×	×	√	×	√	√	×	×	×			
Ayub et al.†	2020	[34]	A-I	√	√	√	×	×	×	√	√	√	√	√	√	×	$7T_H + 2T_P + T_S$	$T_H + 2T_P + T_S$	-
			A-II	√	√	√	×	×	×	√	√	√	√	×	×	×			
Bera et al.†	2020	[88]	A-I	√	√	√	×	×	√	√	×	√	√	√	×	×	$13T_H + T_S + T_B$	$6T_H$	$6T_H$
			A-II	√	√	√	×	×	√	√	×	√	√	√	×	×			
Nikooghadam et al.†	2020	[55]	A-I	√	√	√	×	×	√	√	×	√	√	√	×	×	$T_H + 4T_P + 2T_S$	$2T_H + T_P + 3T_S$	-
			A-II	√	√	√	×	×	√	√	×	√	√	×	×	×			
Mo et al.	2020	[89]	A-I	√	√	√	√	×	√	√	×	√	√	√	×	√	$11T_H + T_E$	$13T_H + T_E$	$6T_H$
			A-II	√	√	√	×	×	√	√	×	√	√	×	×	×			
Hajian et al.	2020	[90]	A-I	√	√	√	×	×	√	√	×	√	√	√	×	×	$13T_H$	$7T_H$	$9T_H$
			A-II	√	√	√	×	×	√	√	×	√	√	√	×	×			
Lee et al.	2020	[91]	A-I	√	√	√	×	×	√	√	×	√	√	×	×	×	$9T_H$	$7T_H$	$7T_H + 2T_S$
			A-II	√	√	√	×	×	√	√	×	√	√	×	×	×			
Srinivas et al.	2020	[4]	A-I	√	√	√	×	×	√	√	×	√	√	×	√	×	$2T_C + 14T_H + T_B$	$10T_H$	$2T_C + 6T_H$
			A-II	√	√	√	×	×	√	√	×	√	√	×	×	×			
Lu et al.†	2019	[7]	A-I	√	√	√	×	×	√	√	×	√	√	×	√	×	$6T_H + 2T_M + 2T_P$	$3T_H + 5T_P + 2T_M$	$7T_H + 3T_M + 7T_P$
			A-II	√	√	√	×	×	√	√	×	√	√	×	×	×			
Wazid et al.†	2018	[22]	A-I	√	√	√	×	×	√	√	×	√	√	×	×	×	$2T_S + T_B + 13T_H$	$4T_S + 5T_H$	$2T_S + 4T_H$
			A-II	√	√	√	×	×	√	√	×	√	√	×	×	×			
Wang et al.	2018	[9]	A-I	√	√	√	√	√	√	√	√	√	√	√	√	√	$5T_H + 2T_S$	$4T_H + 2T_S$	-
			A-II	√	√	√	×	×	√	√	√	√	√	×	×	×			
Ali et al.	2018	[23]	A-I	√	√	√	×	×	√	√	×	√	√	×	×	×	$T_B + 7T_H + 2T_S$	$11T_H + 5T_S$	$4T_H + T_S$
			A-II	√	√	√	×	×	√	√	×	√	√	×	×	×			
Amin et al.	2018	[24]	A-I	√	√	√	×	×	√	√	×	√	√	√	×	×	$T_B + 14T_H$	$16T_H$	$4T_H$
			A-II	√	√	√	×	×	√	√	×	√	√	×	×	×			
Qiu et al.†	2018	[92]	A-I	√	√	√	×	×	√	√	√	√	√	√	√	×	$7T_H + 3T_P$	$4T_H + 2T_P$	-
			A-II	√	√	√	×	×	√	√	√	√	√	×	×	×			
Kumari et al.†	2017	[51]	A-I	√	√	√	√	√	√	√	√	×	√	√	×	√	$5T_H + 3T_P$	$4T_H + 2T_P$	-
			A-II	√	√	√	×	×	√	√	√	×	√	×	×	×			
Kumari et al.†	2017	[56]	A-I	√	√	√	×	×	√	√	√	√	√	√	√	×	$7T_H + 2T_P$	$5T_H + 2T_P$	$6T_H + 2T_P$
			A-II	√	√	√	×	×	√	√	√	√	√	×	×	×			
Kumari et al.†	2016	[6]	A-I	√	√	√	×	×	√	√	√	√	×	×	√	×	$5T_H + 2T_C + T_S$	$6T_H + T_S$	$3T_H + 2T_C$
			A-II	√	√	√	×	×	√	√	√	×	×	×	×	×			
Odelu et al.†	2015	[54]	A-I	√	√	√	×	×	√	√	√	√	√	√	√	×	$7T_H + T_B + 3T_P$	$3T_P + 6T_H$	–
			A-II	√	√	√	×	×	√	√	√	√	√	√	√	×			

[1]: T_E, T_P, T_C, T_B, T_H, T_S, T_M, T_{Ev}, T_{Pu} denote the operation time for modular exponentiation, elliptic curve point multiplication, chebysev chaotic-map, fuzzy extractor, hash, symmetric encryption, message authentication code, the eval and punct functions in PPRF, respectively. Some lightweight operations like XOR and || are omitted.

[2]: $*$ measures whether the scheme can resist offline password guessing attacks; "√" denotes the scheme can provide the corresponding attribute. "×" in the white area denotes that the scheme cannot provide the corresponding attribute under the normal adversary model **A-I**; the gray area shows the evaluation results under the enhanced adversary model **A-II**, which considers ESL attacks; "-" means the item is not applied to the scheme.

†: The authentication scheme claims to be able to resist ESL attacks.

ESL attacks in the field of multi-factor authentication protocols have not been systematically studied.

Not surprisingly, a common feature of these protocols is that they cannot resist offline password guessing attacks under the adversary model **A-II**, which is consistent with our initial intuition (i.e., ESL attackers will essentially removes the one-way trapdoor protection provided by public key cryptography against offline password guessing attacks). To further provide conclusive evidence, we first review the definition of "strong AKE security" in the eCK model [45]. Specifically, in an AKE scheme, two participants exchange information and compute a shared session key with four kinds of secrets, i.e., one party's long-term (static) and ephemeral (random) secret keys and the other party's long-term and ephemeral secret keys. The eCK model allows an adversary to obtain any subset of the above four kinds of information but excludes compromising the same party's long-term and ephemeral secret keys simultaneously. Otherwise, the eCK model regards this case as a trivial attack. However, in a password-based multi-factor authentication scheme, the user's long-term secret is a low-entropy password. Compared with the long-term secret of AKE schemes (a completely random long secret key), a password (a short secret with highly skewed distribution [57]) can be enumerated [77].

As a result, in a password-based multi-factor authentication scheme, *the session-specific ephemeral secret is used to protect the user's password* (long-term secret). More specifically, employing the public technique to build the log-in message (the public-key technique principle, see Section 6.1 of [26]) is indispensable. If an adversary gets the ephemeral secret, she can use it to guess the user's password, which explains why the three representative protocols revisited in this paper cannot resist offline password guessing attacks under the adversary model **A-II**. Therefore, launching ESL attacks in a password-based multi-factor authentication protocol means that the victim's long-term and ephemeral secrets are obtained by the adversary $\mathcal{A}$ at the same time, which violates the definition of "strong AKE security" in the eCK model.

Formal Security Upper Bound for Password-based Multi-factor Authentication Schemes: Semantic security is the major concern of an AKE scheme, which requires that an external attacker cannot distinguish the real session key from a random string of the same length in polynomial time. During the execution of the protocol $\mathcal{P}$, the adversary $\mathcal{A}$ can ask a polynomial number of queries shown in the Section II-C, i.e., Send-query, Execute-query, Corrupt-query, and Reveal-query. At the end of the game, $\mathcal{A}$ can ask a single Test-query and outputs a guess bit b' for the bit b involved in the Test-query. If $b' = b$, $\mathcal{A}$ wins the game. We use $Succ$ to denote this event. The advantage of $\mathcal{A}$ in breaking the semantic security of $\mathcal{P}$ is

$$Adv_{\mathcal{P}}^{AKE}(\mathcal{A}) = 2Pr[Succ(\mathcal{A})] - 1 = 2Pr[b' = b] - 1$$

In a secure password-based multi-factor authentication scheme, an adversary's advantage in attacking the protocol $\mathcal{P}$ should not be greater than launching online password guessing attacks [9], [79], [80]. In other words, online guessing is the best strategy for adversaries to attack a password authentication scheme. Therefore, with a negligible function ϵ, the formal security upper bound for password-based multi-factor authentication schemes under the adversary model **A-I** and **A-II** should be

$$Adv_{\mathcal{P},\mathcal{D}}^{AKE}(\mathcal{A}) \leq C' \cdot q_{send}^{s'} + \epsilon,$$

where $\mathcal{D}$ is the password space that follows Zipf's distribution [57], C' and s' are Zipf parameters.

However, based on our analysis, under the adversary model **A-II**, no password-based MFA schemes can resist offline password guessing attacks. The best strategy to attack these protocols (e.g., [9], [50], [81], [82]) is no longer online guessing. After successfully guessing the victim's password, the adversary now obtains the victim's both long-term and ephemeral secrets, making the claimed ESL resistance in vain. As a result, there is no secure upper bound for these schemes. In summary, ESL attacks are not suitable for analyzing password-based multi-factor authentication protocols since it is meaningless to consider an attack that cannot be defended.

B. Smart Cards Can Make Things Better

In the case of the key exchange scheme (i.e., the Diffie-Hellman scheme), a prime concern is that many applications will compute the ephemeral DH pairs (x, $X = g^x$) in advance to improve the performance of the protocol [41]. In this case, compared with the long-term secrets stored in a hardware-protected area, those ephemeral pairs will be typically stored in insecure memory (e.g., general memory devices) and more vulnerable to leakage [45]. These realistic scenarios motivate ESL attacks. Although none of the password-based multi-factor authentication scheme can be secure when ephemeral secrets are leaked, it does not mean that we are helpless in the face of these attacks. Let's review the conditional non-tamper resistance assumption of smart cards [9], which reveals the fact that the secrets in smart cards can only be extracted by an adversary through side-channel attacks when she somehow obtains the smart card for a relative long time (e.g., a few hours). Otherwise, the smart card can be seen as a hardware-protected area [66] (i.e., the stored secrets cannot be leaked).

Besides, we review the definition of "Session-state reveal" in the CK model [31], which is also the formal description of ESL attacks. Specifically, the adversary provides the name of a party and a session identifier of a *yet incomplete* session at that party and receives the internal state of that session. It should be noted that the premise of this definition is to select a yet incomplete session. The realistic scenario that the ESL attacks rely on can explain this premise, i.e., to improve the efficiency of the key exchange protocol, many applications will calculate the ephemeral secrets in advance and store them in an insecure environment. However, for a smart card-based multi-factor authentication scheme, all the security parameters (including both the long-term and ephemeral secrets) are stored in the smart card. An adversary cannot obtain them before completing the whole authentication session (Conditional non-tamper resistance assumption of smart cards, see Section II-A for more details).

This results in a contradiction between the definition of ESL attacks and the conditional non-tamper resistance smart card

assumption, which makes the premise of an ESL attack invalid, i.e., using smart cards as a possession-based authentication factor can naturally resist ESL attacks, and protocol designers can directly ignore these attacks. Velásquez et al.'s investigation [61] shows that passwords and smart cards are the most commonly used authentication factors. In this situation, most multi-factor authentication schemes (e.g., [7], [21], [50]) do not need to consider ESL attacks but are still listed as targets of ESL attacks. Since ESL attacks were introduced into the multi-factor authentication field by Islam and Biswas in 2011 [29], this has become a long-standing misunderstanding.

Our main idea is to highlight the long-standing misunderstanding between ESL attacks and password-based MFA, i.e., there is no secure password-based multi-factor authentication scheme resilient to ESL attacks. However, if a scheme employs the smart card as an authentication factor and does not pre-compute any ephemeral secret. In that case, it is inherently immune to ESL attacks. Note that if ephemeral secrets are calculated in advance and stored in smart cards, adversaries could still obtain them. However, such attacks fall under smart card loss attacks [27] rather than ESL attacks. For authentication scenarios that do not involve smart cards, some secure computing environments similar to smart cards, such as the trusted execution environment (TEE), may have the same effect, but this is beyond our scope of work.

Krawczyk points out that "a good security system is not one that denies the possibility of failures but rather one designed to confine the adverse effects of such failures to the possible minimum" [41]. This fundamental security principle requires continuously expanding the scope of known attacks. In other words, introducing possible/newly emerging attacks to challenge the protocol's security would stimulate new designs to deal with potential threats. We believe this is also the original intention of previous works that introduce ESL attacks into multi-factor authentication.

However, protocol designers "tend to assert the superiority of their scheme while (maybe unconsciously) ignoring the features that their scheme fails to support" [5]. Our work provides clear evidence for this insight. The countermeasures to ESL attacks proposed in previous works, that is, using both ephemeral secrets and long-term secrets to construct session keys, can resist ESL attacks intuitively, but designers have ignored the latent offline password guessing attacks when considering an ESL attacker, which ultimately makes the efforts for preventing ESL attacks against password-based MFA protocols in vain. These failed efforts (see [19], [21], [32], [38], [50]) highlight the importance of employing devices with trusted computing environments (e.g., smart cards) to make ESL impractical in subsequent password-based multi-factor authentication protocol designs (e.g., [9], [60], [82]).

XI. Conclusion

In this paper, we have taken a substantial first step towards systematically exploring the relationship between ESL attacks and MFA schemes. We first categorize existing multi-factor authentication schemes involving ESL attacks into four categories, and cryptanalyze a representative protocol for each ESL category. After that, we investigate the root causes for the failure of these representative protocols, and reveal that ESL attacks are unsuitable for analyzing the password-based MFA schemes. Besides, we conduct a large-scale comparative measurement of 41 representative MFA schemes and show comparison results under two adversary models (i.e., with and without ESL). Finally, we point out the subtleties between realistic scenarios on which the ESL attacks are based and the conditional non-tamper resistance assumption of smart cards, and conclude that using smart cards as an authentication factor can naturally resist ESL attacks. We believe this work provides a better understanding of ESL attacks in password-based multi-factor authentication for mobile devices and also has strong guiding significance for defending against ESL attacks in non-mobile device environments.

References

[1] C. Chang and T. Wu, "Remote password authentication with smart cards," *IEE Comput. Digit. Tech.*, vol. 138, no. 3, pp. 165–168, 1991.

[2] X. Huang et al., "Robust multi-factor authentication for fragile communications," *IEEE Trans. Dependable Secur. Comput.*, vol. 11, no. 6, pp. 568–581, Nov./Dec. 2014.

[3] J. Li et al., "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.

[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 Secur. Comput.*, vol. 17, no. 6, pp. 1133–1146, Nov./Dec. 2020.

[5] D. Wang, D. He, P. Wang, and C. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Trans. Dependable Secur. Comput.*, vol. 12, no. 4, pp. 428–442, Jul./Aug. 2015.

[6] S. Kumari et al., "A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps," *Future Gener. Comput. Sys.*, vol. 63, pp. 56–75, 2016.

[7] Y. Lu, G. Xu, L. Li, and Y. Yang, "Robust privacy-preserving mutual authenticated key agreement scheme in roaming service for global mobility networks," *IEEE Syst. J.*, vol. 13, no. 2, pp. 1454–1465, Jun. 2019.

[8] M. J. Hossain et al., "ICAS: Two-factor identity-concealed authentication scheme for remote-servers," *J. Syst. Archit.*, vol. 117, 2021, Art. no. 102077.

[9] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE Trans. Dependable Secur. Comput.*, vol. 15, no. 4, pp. 708–722, Jul./Aug. 2018.

[10] A. K. Sutrala et al., "Authenticated key agree- ment 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.

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

[12] C. Chen et al., "A secure authenticated and key exchange scheme for fog computing," *Enterp. Inf. Syst.*, vol. 15, no. 9, pp. 1200–1215, 2021.

[13] M. Wazid et al., "Design and testbed experiments of user authentication and key establishment mechanism for smart healthcare cyber physical systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 2697–2709, Sep./Oct. 2023.

[14] S. Shamshad et al., "An efficient privacy-preserving authenticated key establishment protocol for health monitoring in industrial cyber-physical systems," *IEEE Internet Things J.*, vol. 9, no. 7, pp. 5142–5149, Apr. 2022.

[15] D. Kumar et al., "An efficient anonymous user authentication and key agreement protocol for wireless sensor networks," *Int. J. Commun. Syst.*, vol. 34, no. 5, pp. e4724, 2021.

[16] Y. Xia et al., "PUF-assisted lightweight group authentication and key agreement protocol in smart home," *Wireless Commun. Mob. Comput.*, pp. 1–15, 2022.

[17] J. Srinivas et al., "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.

[18] B. A. Alzahrani, A. Barnawi, and S. A. Chaudhry, "A resource-friendly authentication protocol for UAV-based massive crowd management systems," *Secur. Commun. Netw.*, vol. 2021, 2021, Art. no. 3437373.

[19] J. Zhang et al., "SMAKA: Secure many-to-many authentication and key agreement scheme for vehicular networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 1810–1824, 2021.

[20] T. Wu et al., "A provably secure three-factor authentication protocol for wireless sensor networks," *Wireless Commun. Mob. Comput.*, vol. 2021, 2021, Art. no. 5537018.

[21] J. Lee et al., "Secure and efficient honey list-based authentication protocol for vehicular ad hoc networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2412–2425, Jul.–Sep. 2021.

[22] M. Wazid et al., "Design of secure user authenticated key management protocol for generic IoT networks," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 269–282, Feb. 2018.

[23] R. Ali et al., "A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring," *Future Gener. Comput. Syst.*, vol. 84, pp. 200–215, 2018.

[24] R. Amin et al., "An untraceable and anonymous password authentication protocol for heterogeneous wireless sensor networks," *J Netw. Comput. Appl.*, vol. 104, pp. 133–144, 2018.

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

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

[27] D. Wang, Q. Gu, H. Cheng, and P. Wang, "The request for better measurement: A comparative evaluation of two-factor authentication schemes," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur.*, 2016, pp. 475–486.

[28] 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. Dependable Secur. Comput.*, vol. 19, no. 1, pp. 507–523, Jan./Feb. 2022.

[29] S. H. Islam and G. P. Biswas, "A more efficient and secure ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *J. Syst. Softw.*, vol. 84, no. 11, pp. 1892–1898, 2011.

[30] T. Chen, Y. Chen, and W. Shih, "An advanced ECC ID-based remote mutual authentication scheme for mobile devices," in *Proc. 7th Int. Conf. Ubiquitous Intell.& Comput. 7th Int. Conf. Autonomic& Trusted Comput.,*2010, pp. 116–120.

[31] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Proc. EUROCRYPT*2001, pp. 453–474.

[32] S. Gui, M. Zheng, and Y. Liu, "One cycle with N messages: A secure and lightweight key agreement scheme for WSNs," in *Proc. 21st Annu. IEEE Int. Conf. Sens., Commun., Netw.*, 2024, pp. 1–9.

[33] B. A. Alzahrani et al., "A privacy preserving authentication scheme for roaming in IoT-based wireless mobile networks," *Symmetry*, vol. 12, no. 2, 2020, Art. no. 287.

[34] M. F. Ayub et al., "A provably secure two-factor authentication scheme for USB storage devices," *IEEE Trans. Consum. Electron.*, vol. 66, no. 4, pp. 396–405, Nov. 2020.

[35] C. Dai and Z. Xu, "A secure three-factor authentication scheme for multi-gateway wireless sensor networks based on elliptic curve cryptography," *Ad Hoc Netw.*, vol. 127, 2022, Art. no. 102768.

[36] Y. Han, H. Guo, J. Liu, B. B. Ehui, Y. Wu, and S. Li, "An enhanced multifactor authentication and key agreement protocol in industrial Internet of Things," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 16243–16254, May 2024.

[37] 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.

[38] X. Ai, A. Badshah, S. Tu, M. Waqas, and I. Ahmad, "An improved ultra-lightweight anonymous authenticated key agreement protocol for wearable devices," *IEEE Trans. Mob. Comput.*, vol. 24, no. 5, pp. 4543–4557, May 2025.

[39] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Proc. CRYPTO 1993*, pp. 232–249.

[40] M. Bellare, R. Canetti, and H. Krawczyk, "A modular approach to the design and analysis of authentication and key exchange protocols," in *Proc. 30th Annu. ACM Symp. Theory Comput., 1998*, pp. 419–428.

[41] H. Krawczyk, "HMQV: A high-performance secure Diffie-Hellman protocol," in *Proc. CRYPTO 2005*, pp. 546–566.

[42] W. Wang, "Heartbleed–openssl zero-day bug leaves millions of websites vulnerable," Apr. 2014, [Online]. Available: https://thehackernews.com/2014/04/heartbleed-openssl-zero-day-bug-leaves.html?utm_source=tuicool&utm_medium=referral

[43] M. Lipp et al., "Meltdown: Reading kernel memory from user space," in *Proc. USENIX SEC*, pp. 1–16, 2018.

[44] P. Kocher et al., "Spectre attacks: Exploiting speculative execution," in *Proc. IEEE Symp. Secur. Privacy*, 2019, pp. 1–19.

[45] B. A. LaMacchia, K. E. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *Proc. ProvSec*2007.

[46] Y. Tseng et al., "A novel ID-based authentication and key exchange protocol resistant to ephemeral-secret-leakage attacks for mobile devices," *Int. J. Distrib. Sens. Netw.*, vol. 11, no. 5, 2015, Art. no. 898716.

[47] T. Wu and Y. Tseng, "An efficient user authentication and key exchange protocol for mobile client-server environment," *Comput. Netw.*, vol. 54, no. 9, pp. 1520–1530, 2010.

[48] D. He et al., "Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography," *IET Commun.*, vol. 10, no. 14, pp. 1795–1802, 2016.

[49] J. Tsai and N. Lo, "Secure anonymous key distribution scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 906–914, Mar. 2016.

[50] S. Qiu, D. Wang, G. Xu, and S. Kumari, "Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices," *IEEE Trans. Dependable Secur. Comput.*, vol. 19, no. 2, pp. 1338–1351, Mar./Apr. 2022.

[51] S. Kumari et al., "An improved smart card based authentication scheme for session initiation protocol," *Peer-to-Peer Netw. Appl.*, vol. 10, no. 1, pp. 92–105, 2017.

[52] Y. Lu et al., "Anonymous three-factor authenticated key agreement for wireless sensor networks," *Wireless Netw.*, vol. 25, no. 4, pp. 1461–1475, 2019.

[53] P. Gope and T. Hwang, "Lightweight and energy-efficient mutual authentication and key agreement scheme with user anonymity for secure communication in global mobility networks," *IEEE Syst. J.*, vol. 10, no. 4, pp. 1370–1379, Dec. 2016.

[54] V. Odelu et al., "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.

[55] M. Nikooghadam and H. Amintoosi, "A secure and robust elliptic curve cryptography-based mutual authentication scheme for session initiation protocol," *Secur. Priv.*, vol. 3, no. 1, 2020, Art. no. e92.

[56] S. Kumari et al., "Design of a provably secure biometrics-based multi-cloud-server authentication scheme," *Future Gener. Comput. Syst.*, vol. 68, pp. 320–330, 2017.

[57] D. Wang et al., "Zipf's law in passwords," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 11, pp. 2776–2791, Nov. 2017.

[58] B. Pal et al., "Beyond credential stuffing: Password similarity models using neural networks," in *Proc. IEEE Symp. Secur. Privacy 2019*, pp. 417–434.

[59] S. Halevi and H. Krawczyk, "Public-key cryptography and password protocols," *ACM Trans. Inf. Syst. Secur.*, vol. 2, no. 3, pp. 230–268, 1999.

[60] Q. Wang et al., "Quantum2FA: Efficient quantum-resistant two-factor authentication scheme for mobile devices," *IEEE Trans. Dependable Secur. Comput.*, vol. 20, no. 1, pp. 193–208, Jan./Feb. 2023.

[61] I. Velásquez, A. Caro, and A. Rodrıguez, "Authentication schemes and methods: A systematic literature review," *Inf. Softw. Technol.*, vol. 94, pp. 30–37, 2018.

[62] I. Liao, C. Lee, and M. Hwang, "A password authentication scheme over insecure networks," *J. Comput. Syst. Sci.*, vol. 72, no. 4, pp. 727–740, 2006.

[63] M. Hwang and L. Li, "A new remote user authentication scheme using smart cards," *IEEE Trans. Consum. Electron.*, vol. 46, no. 1, pp. 28–30, Feb. 2000.

[64] M. Carbone et al., "Deep learning to evaluate secure RSA implementations," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2019, no. 2, pp. 132–161, 2019.

[65] T. Roche et al., "A side journey to titan," in *Proc. USENIX SEC, 2021*, pp. 231–248.

[66] D. Wang and P. Wang, "Offline dictionary attack on password authentication schemes using smart cards," in *Proc. ISC*, pp. 221–37, 2013.

[67] Y. Wang, "Password protected smart card and memory stick authentication against off-line dictionary attacks," in *Proc. SEC, 2012*, pp. 489–500.
[68] J. Bonneau, "Guessing human-chosen secrets," Ph.D. dissertation, Univ. of Cambridge, Cambridge, U.K., 2012.
[69] J. Bonneau, "The science of guessing: Analyzing an anonymi- zed corpus of 70 million passwords," in *Proc. IEEE Symp. Secur. Privacy*, 2012, pp. 538–552.
[70] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–207, Mar. 1983.
[71] D. Temoshok et al., "NIST 800-63B-4 IPD digital identity guidelines: Authentication and lifecycle management," McLean, VA, Tech. Rep. NIST SP 800-63B-4, Dec. 2022.
[72] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Proc. EUROCRYPT, 2000*, pp. 139–155.
[73] 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. Inform.*, vol. 14, no. 9, pp. 4081–4092, Sep. 2018.
[74] M. Xu, D. Wang, Q. Wang, and Q. Jia, "Understanding security failures of anonymous authentication schemes for cloud environments," *J. Syst. Archit.*, vol. 118, 2021, Art. no. 102206.
[75] D. Wang, S. Hong, and Q. Wang, "Revisiting a multifactor authentication scheme in industrial IoT," *Secur. Commun. Netw.*, vol. 2021, 2021, Art. no. 9995832.
[76] J. Ma, W. Yang, M. Luo, and N. Li, "A study of probabilistic password models," in *Proc. IEEE Symp. Secur. Privacy*, 2014, pp. 689–704.
[77] D. Wang et al., "Targeted online password guessing: An underestimated threat," in *Proc. ACM CCS*, pp. 1242–1254, 2016.
[78] X. Li et al., "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.
[79] E. Bresson et al., "Security proofs for an efficient password-based key exchange," in *Proc. ACM CCS*, pp. 241–250, 2003.
[80] P. MacKenzie, "The pak suite: Protocols for password-authenticated key exchange," *Contributions IEEE P*, vol. 1363, no. 2, 2002.
[81] M. Xu and D. Wang, "Practical two-factor authentication protocol for real-time data access in WSNs," *IEEE Trans. Dependable Secur. Comput.*, vol. 22, no. 5, pp. 5215–5230, Sep./Oct. 2025.
[82] X. Zhao, Q. Jiang, X. Gong, M. Li, X. Ma, and J. Ma, "Three birds with one arrow: Symmetric two-factor authentication protocol based on puncturable Pseudorandom function," *IEEE Trans. Inf. Forensics Secur.*, vol. 20, pp. 10849–10863, 2025.
[83] Y. Guo, Z. Zhang, and Y. Guo, "Anonymous authenticated key agreement and group proof protocol for wearable computing," *IEEE Trans. Mob. Comput.*, vol. 21, no. 8, pp. 2718–2731, Aug. 2022.
[84] D. Kumar, "A secure and efficient user authentication protocol for wireless sensor network," *Multim. Tools Appl.*, vol. 80, no. 18, pp. 27131–27154, 2021.
[85] R. Qi et al., "Security preservation in industrial medical CPS using chebyshev map: An AI approach," *Future Gener. Comput. Syst.*, vol. 122, pp. 52–62, 2021.
[86] T. Wu et al., "Fog-driven secure authentication and key exchange scheme for wearable health monitoring system," *Secur. Commun. Netw.*, vol. 2021, pp. 8368646:1–8368646:14, 2021.
[87] M. Wazid et al., "LAM-CIoT: Lightweight authentication mechanism in cloud-based IOT environment," *J. Netw. Comput. Appl.*, vol. 150, 2020, Art. no. 102496.
[88] B. Bera et al., "On the design of biometric-based user authentication protocol in smart city environment," *Pattern Recognit. Lett.*, vol. 138, pp. 439–446, 2020.
[89] J. Mo, W. Shen, and W. Pan, "An improved anonymous authentication protocol for wearable health monitoring systems," *Wireless Commun. Mob. Comput.*, vol. 2020, 2020, Art. no. 5686498.
[90] R. Hajian et al., "SHAPARAK: Scalable healthcare authentication protocol with attack-resilience and anonymous key-agreement," *Comput. Netw.*, vol. 183, 2020, Art. no. 107567.
[91] H. Lee et al., "A three-factor anonymous user authentication scheme for Internet of Things environments," *J. Inf. Secur. Appl.*, vol. 52, 2020, Art. no. 102494.
[92] S. Qiu et al., "An enhanced password authentication scheme for session initiation protocol with perfect forward secrecy," *PLoS one*, vol. 13, no. 3, 2018, Art. no. e0194072.

Ding Wang received the PhD degree in information security from Peking University, in 2017, and was supported by the "Boya Postdoctoral Fellowship" in Peking University from 2017 to 2019. Currently, he is a full professor with Nankai University. As the first author (or corresponding author), he has published more than 100 papers with venues like 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 more than 200 media like The Wall Street Journal, Daily Mail, Forbes, IEEE Spectrum, and Communications of the ACM, and resulted in the revision of the authentication guideline NIST SP800-63-2. He has been involved in the community as a PC chair/TPC member for more than 60 international conferences such as NDSS 2023–2025, ACM CCS 2022, PETS 2022–2024, ACSAC 2020–2024, RAID 2024/2023, IEEE EuroS&P 2025, FC 2026/2025, ACM AsiaCCS 2025/2022/2021, ICICS 2018–2025, SPNCE 2020–2022. He has received the "ACM China Outstanding Doctoral Dissertation Award", the Best Paper Award at INSCRYPT 2018, the First Prize of Cryptologic Innovation Award of the China Association for Cryptologic Research, and the First Prize of Natural Science Award of Ministry of Education. His main research interests focus on passwords, authentication, and provable security.

Meijia Xu received the MS degree from the College of Cryptology and Cyber Science, Nankai University, Tianjin, P.R. China, in June 2023. She is currently working toward the PhD degree with the College of Cryptology and Cyber Science, Nankai University, Tianjin, P.R. China. She has published papers with venues like *IEEE Transactions on Dependable and Secure Computing* and *Journal of Systems Architecture*. Her research interests include applied cryptography and password-based authentication.

Qingxuan Wang received the PhD degree in information security from Nankai University, in 2024. He is currently a postdoctoral researcher with Nankai University, Tianjin, P.R. China. He has published papers with venues like *IEEE Transactions on Dependable and Secure Computing*, *IEEE Transactions on Information Forensics and Security* and *IEEE Transactions on Services Computing*. He has been involved in the community as a Submission chair for ICICS 2023, and a Web chair for SPNCE2022. He has received the 2024 Excellent Paper Award in Information Countermeasure Technology and the Second Prize for Scientific and Technological Progress of Tianjin Municipality, and authored two "ESI highly cited papers". His research interests include applied cryptography and password-based authentication.