Accepted by IEEE Transactions on Information and Forensics & Security, 2022 DOI: 10.1109/TIFS.2022.3227753

Understanding Failures in Security Proofs of Multi-factor Authentication for Mobile Devices

Qingxuan Wang and Ding Wang

***Abstract*—Multi-factor authentication is a promising way to enhance the security of password-based authenticated key exchange (PAKE) schemes. It is widely deployed in various daily applications for mobile devices (e.g., e-Bank, smart home, and cloud services) to provide the first line of defense for system security. However, despite intensive research, how to design a secure and efficient multi-factor authentication scheme is still a challenging problem. Hundreds of new schemes have been successfully proposed, and many are even equipped with a formal security proof. However, most of them have been shortly found to be insecure and cannot achieve the claimed security goals. Now a paradox arises: *How can a multi-factor scheme that was "formally proven secure" later be found insecure*?**

To answer this seemingly contradicting question, this paper takes a substantial first step towards systematically exploring the security proof failures in multi-factor authentication schemes for mobile devices. We first investigate the root causes of the "provable security" failure in vulnerable multi-factor authentication schemes under the random oracle model, and classify them into eight different types in terms of the five steps of conducting a formal security proof. Then, we elaborate on *each* type of these eight proof failures by examining three typical vulnerable protocols, and suggest corresponding countermeasures. Finally, we conduct a large-scale comparative measurement of 70 representative multi-factor authentication schemes under our extended evaluation criteria. The schemes we select range from 2009 to 2022, and the comparison results suggest that understanding failures in formal security proofs is helpful to design more secure multi-factor authentication protocols for mobile devices.

***Index Terms*—Multi-factor authentication; Provable security; Mobile devices; Random oracle model.**

I. Introduction

The concept of "ubiquitous computing" [1] has opened the era of mobile Internet [2]. Nowadays, many popular remote services are based on mobile Internet, such as the Internet of Things (IoT), smart home, vehicular ad hoc networks (VANET), cloud services, e-commerce and e-health. The value proposition of mobile Internet has gradually evolved from simply extending or replacing wired networks to cloud-assisted smart object intelligence. However, whether in wireless sensor networks (WSNs), which is an indispensable technical basis of the IoT [3], single/multi-server architectures in distributed systems, or cloud-based networks, there is a problem that the sensitive data transmitted in them could be accessed by malicious adversaries [4]. Therefore, employing a well-designed authenticated key exchange scheme is a key solution for the above mentioned problem.

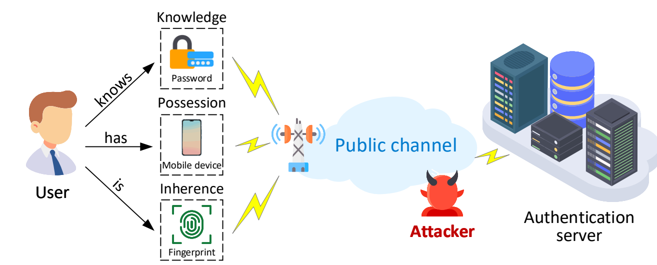


Fig. 1: Multi-factor authentication for mobile devices.

An authenticated key exchange scheme provides user authentication and establishes a shared session key for secure data transmission in public channels. To achieve user authentication, handreds of authentication methods [6]–[9] have been proposed, and they can be categorized as: 1) "knowledge": something the user knows (e.g., passwords); 2) "possession": something the user possesses (e.g., smart cards); 3) "inherence": something the user is (e.g., fingerprints). Among them, passwords are most widely and unlikely to be replaced in the near future [10]. Password-based authenticated key exchange (PAKE) converts a low-entropy password into a high-entropy shared session key, and has attracted intensive attention [11]–[13]. In PAKE schemes, the server needs to maintain a password verification table for verifying the identity of the user. This password table is an attractive attacking target. Recent years we have seen unending large-scale password leaks, e.g., the Yahoo 3 billion leak [14], the Rockyou 8.4 billion leak [15], and the FlexBooker 3.7 million leak [16].

Like passwords, each element that could be used for user authentication has inherent defects. For smart cards, both potential smart card loss and smart card theft pose serious threats to its security. Particularly, with the development of side-channel attacks, some emerging technologies such as energy analysis [17] and reverse engineering [18] make the security parameters stored in the smart card accessible to adversaries (non-tamper resistance assumption of the smart card [5]). What's worse, in 2019, Carbone et al. [19] proposed an attack for an RSA algorithm implementation on a processor equipped with common side-channel attack defense methods such as blind modulus and exponent. For biological factors, there have been serious and irrevocable threats that affected individuals (victims of biological information leakage) cannot update or replace the comprised biometric features [20].

As single-factor based authentication schemes have inherent security drawbacks, they are not suitable for security-critical applications. Therefore, it is natural to combine two or

This research is supported by the National Natural Science Foundation of China under Grant No. 62222208, and by Natural Science Foundation of Tianjin, China under Grant No. 21JCZXJC00100. The corresponding author is Ding Wang.

Qingxuan Wang and Ding Wang are with the College of Cyber Science, Nankai University, Tianjin 300350, China, and also with Tianjin Key Laboratory of Network and Data Security Technology, Nankai University, Tianjin 300350, China. (Email: wangqingxuan@mail.nankai.edu.cn; wangding@nankai.edu.cn;).

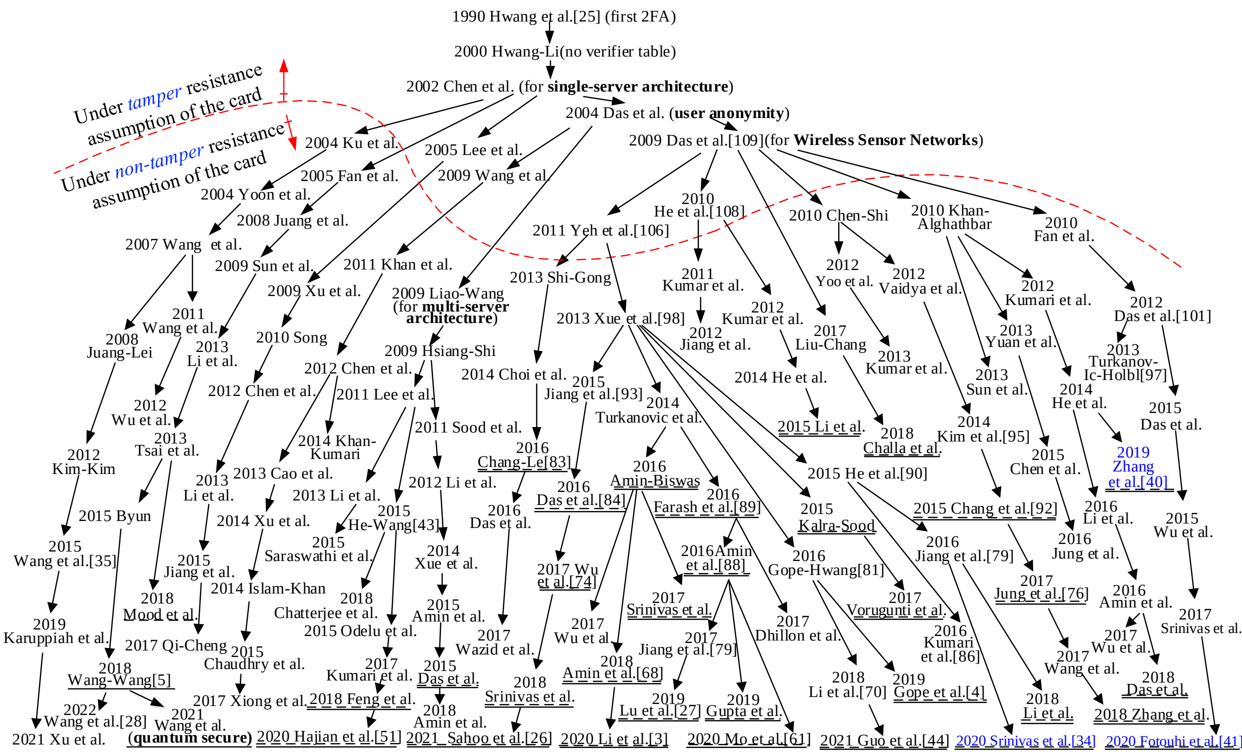


Fig. 2: A brief history of multi-factor user authentication. This figure is based on Fig.2 of [5]. Each child node claims to be an improved scheme of its parent node but is found insecure by its child nodes. Schemes underlined with a solid line are equipped with formal security proof, and those underlined with a dashed line have flaws that exist in their formal security proof. It can be found that only a few works conduct correct security proofs.

more authentication factors to build multi-factor authentication schemes that achieve higher security. However, a simple combination of multiple factors is likely to lead to a system that simply inherits the individual weakness from each factor [21], [22]. Thus, how to avoid the inherent weakness of each authentication factor while ensuring the entire system enjoy "truly multi-factor security" is a great challenge.

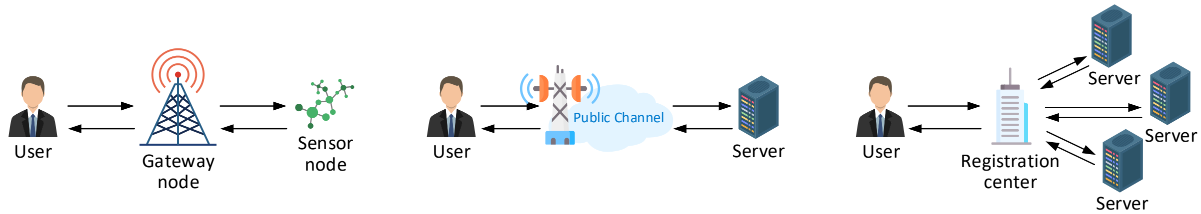
A. Motivations

The past 30 years have witnessed the tough development of multi-factor authentication [5], [23], [24]. Since Hwang et al. [25] designed the first two-factor authentication scheme in 1990, hundreds of relevant attempts have been proposed. However, behind the prosperity lurks some crisis. Most schemes are found to have various defects shortly after they were proposed. *How to design a secure multi-factor authentication scheme* remains a challenging question. Some protocols are vulnerable to known attacks, such as smart card loss attacks [26], [27] or node capture attacks in WSNs [28]; the others fail to achieve important security goals, such as forward security [29]–[31] or mutual authentication [32], [33]. To addressing these issues, new protocols are unceasingly proposed. Generally, the research history of this area falls into an unsatisfactory cycle of "break-fix-break-fix" [5]. As shown in Fig. 2, we provide a brief development history of this field.

Fortunately, some nice progress has been made. Many studies [5], [34]–[36] strive to prevent this vicious cycle by investigating the evaluation criteria. They manage to address the question of whether or not there are inherent limitations that prevent us from designing an ideal scheme that satisfies all the desired security goals and features. Specifically, based on the smart card non-tamper resistance assumption, Madhusudhan and Mittal [34] propose nine security requirements and ten desirable attributes. After that, Wang et al. [35] point out some redundancies and inherent conflicts in their evaluation sets. Accordingly, they give an evaluation set of 12 criteria in [5], which eliminates the redundancies and inherent conflicts in [34] and takes the harshest adversary model. Thus, it is suitable for evaluating multi-factor authentication protocols, and we employ it as a building block (See Section II-C).

Besides the evaluation criteria, the line of research strives to elevate the situation by resorting to provable security. In 1984, Goldwasser and Micali [37] proposed the concept of semantic security, which is the seminal work of provable security. After that, a series of influential works [38], [39] were proposed. These works advocate linking the security of the proposed protocol with well-known difficult problems (e.g., the discrete logarithm problem and the integer factorization problem). First, suppose that an adversary can attack the protocol with a non-negligible advantage. Then, she can be used to construct another adversary who can attack the difficult problem with the same advantage. Finally, the security of the protocol is proved by creating contradictions. As a result, provable security is indispensable in analyzing and evaluating the security of a cryptographic protocol. However, things are not going well, as shown in Fig. 2. Many protocols are still found to have serious defects [40]–[42], even equipped with formal proof.

Now, a question arises: *Why does a protocol that proves to be secure still has vulnerabilities?* This is a natural but rarely concerned question because when designing a new scheme, proponents usually pay more attention to what new techniques are employed and how to use these techniques to fix the previous protocol. With this fundamental question untouched, more new attempts will only become a part of the vicious "break-fix-break-fix" circle yet contribute little real progress.



(a) System models in WSNs (b) Single-server architecture (c) Multi-server architecture

Fig. 3: Three typical communication models for multi-factor authentication schemes for mobile devices.

B. Contributions

In this work, we aim to give a definite answer to the above question and further contribute to tackling the challenge of how to design a secure multi-factor authentication scheme. With the experience of analyzing more than 200 multi-factor authentication schemes, we manage to figure out the various causes of the failed security proof. Generally, the process of conducting the "provable security" under the random oracle model (ROM) could be divided into five steps [35]: (1) Define the adversary model; (2) Declare security goals; (3) State cryptographic assumptions; (4) Describe the protocol; (5) Reductionist proof. Based on these five steps, we classify the failures of "provable security" into eight types. Further, we explain each type of proof failure by pointing out the vulnerabilities and the corresponding flaws of security proof in typical schemes. The contributions of this paper are three-fold:

(1) We investigate the root causes of provable security failures in dozens of vulnerable multi-factor authentication schemes under the ROM model, and classify these causes into eight different types in terms of the five steps of conducting formal security proofs. As far as we know, we are the first to provide a concrete taxonomy of failures in formal security proofs.

(2) We elaborate on each type of security proof failure by first pointing out the vulnerabilities of a typical multi-factor authentication scheme. Then we show the flaws in their formal security proof. Finally, we give corresponding suggestions on conducting formal security proof.

(3) We combine our taxonomy of failures in security proofs with the protocol evaluation criteria proposed in [5] and form an improved evaluation set. Based on this new evaluation set, we conduct a large-scale comparative evaluation of 70 multi-factor authentication schemes. The comparison provides the neglected measurements and presents a better understanding of existing schemes.

II. System model, Adversary model, and Evaluation criteria

In this section, we give the system model, widely accepted adversary model, and Wang-Wang's evaluation criteria [5].

A. Applicable System Model

There are three typical system models when applying multi-factor authentication schemes (see Fig. 3). The first model, Model (a), is the standard for multi-factor authentication in WSNs and is recommended by Wang et al. [36]. They evaluate eight types of WSNs system architecture and conclude Model (a) is better than other models. The participants in this model are a set of user U, a gateway node GWN, and distributed sensor nodes SN. This kind of scheme comprises four basic phases, including registration, log-in and authentication, password change, and dynamic sensor node addition [36]. In the registration phase, U chooses a user name ID and password PW, then submits them to the GWN, and the GWN issues a smart card or a device to the user. The smart card/mobile device may contain some public and user-related security parameters, which could be used to verify the user's identity. After that, the user is able to access the GWN in the log-in and authentication phase. The password change phase and dynamic node addition phase are necessary to resist the password leakage [5] and sensor nodes compromise [36].

The remaining two models are single-server architecture [5] (Model (b) in Fig. 3) and multi-server architecture [43] (Model (c) in Fig. 3) in general environments. The single-server architecture participants contain a set of users and a remote server. Specifically, the authentication schemes in this architecture consist of three basic phases, i.e., registration, log-in and authentication, and password change. Compared with the single-server architecture, multi-server architecture involves an additional participant, i.e., registration center (RC). After registering with the RC, a user can obtain service from multiple servers. There are four basic phases in a multi-server architecture scheme: user registration, server registration, login-in and authentication, and password change. Remarkably, our findings in this paper have universal applicability to the above mentioned system models. The main reason is that the process of conducting provable security on which our findings rely can be directly applied to prove the multi-factor authentication schemes in these system models. More specifically, the underlying adversary models and security goals, which are the core steps of provable security, are similar in the above mentioned application scenarios.

B. Adversary Model

In order to assess the security of a cryptographic scheme, a realistic and concrete adversary model is necessary. Besides, defining the adversary model is the first step in conducting security proofs. According to our analysis, the incomplete definition of the adversary model is the main reason for the failure of many security proofs. Therefore, it is necessary to give a comprehensive adversary model. Following the existing work [5], the adversary's capabilities are as follows.

A1. $\mathcal{A}$ can offline enumerate all items in the Caresian product of identity and password space $\mathcal{D}_{id} \times \mathcal{D}_{pw}$ within polynomial time, or get user's identity only when evaluating the scheme's security.

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, such as, users, gateway nodes (GWNs), and sensor nodes (SNs).

A3. $\mathcal{A}$ may learn the user's password via a malicious card reader, extract the secret in the lost smart card by side-channel attacks, or attain a victim's biometrics using malicious devices. But the above cases cannot be

TABLE I: A Taxonomy of Failures in Formal Security Proofs of Multi-factor Authentication for Mobile Devices.

Type	Security proof step†	The main reason for the failure of the proof	Associated attacks⋆	Ref.	Cases
I	Define adversary models	Regard additional authentication factors as a secure black box*	Password guessing attacks	Sec. V-A	[42] [29]
II		The user password disclosure query is not defined	User impersonation attacks	Sec. VII-B	[41]
III		Do not consider a legitimate user might be an attacker	Insider attacks	Sec. V-C	[42] [27]
IV	Declare security goals	Declare a wrong security goal	Password guessing attacks	Sec. IX-A	[40] [29]
V		Adopt the wrong password distribution model	Password guessing attacks	Sec. III	[44] [30]
VI	State assumptions	Public key cryptography primitives are not applied	Password guessing attacks	Sec. III	[32] [31]
VII	Reductionist proof	Do not correctly reduce the adversary's advantage	Password guessing attacks	Sec. V-B	[42] [44]
VIII	Not suitable‡	Some attacks cannot be described by provable security	De-Synchronization attacks	Sec. VII-C	[41] [32]

†: Our taxonomy is based on the five steps of conducting the provable security [35]: i.e., (1) Define adversary models; (2) Declare security goals; (3) State cryptographic assumptions; (4) Describe the protocol; (5) Reductionist proof. Among these steps, the correct description of the protocol is the basic requirement of any protocol design. Thus, no work makes mistakes in step (4), and it is not considered in this taxonomy.

‡: Some complex attacks, such as de-synchronization attacks and man-in-the-middle attacks, are not suitable for provable security.

*: Additional authentication factors refer to any authentication factors other than passwords (e.g., biometrics and smart cards).

⋆: The known attacks that directly related to each type of security failure. Note that there may be multiple attacks corresponding to each security proof failure, and only one is listed in the table due to space constraints.

achieved simultaneously, 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 GWN/server's secret key(s) when assessing the system's forward security.

A5. $\mathcal{A}$ can compromise a limited number of SNs, i.e., extracting the sensitive data stored in SNs, and impersonating the compromised SNs to join the communication between the users and the GWN.

A6. $\mathcal{A}$ can register to be a legitimate user of the system or an administrator of the GWN/server.

The capability A1 is reasonable because both the identity space D_{id} and the password space D_{pw} are limited ($D_{id} \leq D_{pw} \approx 10^6$ [45]). Essentially, the passwords are human-chosen short keys with low entropy [46], and the identities are static texts with predefined structure, which is of little cryptographic strength and should not be considered as a secret [35]. The capability A2 is based on the wildly accepted Dolev-Yao model [47]. The capability A3 follows the harshest adversary model in [5] and represents the goal of "truly multi-factor security", that is, the security of an n-factor authentication scheme cannot be compromised by an adversary who even holds n-1 factors. The capability A4 is the common assumption when measuring forward security [35].

The capability A5 models node capture attacks in WSNs environments [28]. Since the sensor nodes are usually resource-constrained devices, complex cryptographic algorithms cannot be applied to them. Besides, they are deployed in unattended environments to collect data, and without physical protection, adversaries can capture sensor nodes easily. Under capability A6, $\mathcal{A}$ could be an insider attacker of the system.

C. Evaluation Criteria

The evaluation criteria are the touchstone of the multi-factor authentication protocol design. Wang and Wang proposed an evaluation set for two-factor authentication schemes [5] in the general environment. After that, they update it to the WSNs environment [36]. Combining these state-of-the-art evaluation frameworks, we show the 12 evaluation criteria employed.

C1. **No password verifier table**: The GWN/server and sensor nodes don't store the relevant value of the registered users' password.

C2. **Password friendly**: The users could choose the password by themselves and change it anytime.

C3. **No password exposure**: Even the administrator of GWN/server cannot extract the users' password.

C4. **No smart card/device loss attack**: The scheme is free from smart card/device loss attack, i.e., if an attacker captures the user's lost smart card or mobile device and extract the secure parameters stored in it, she cannot recover the password or even impersonate the user by using the offline/online password guessing attacks.

C5. **Resistance to known attacks**: The scheme can resist various kinds of attacks, such as replay attacks, node capture attacks, man-in-the-middle attacks and de-synchronization attacks.

C6. **Sound repairability**: Allowing the user to revoke her smart card without changing her identity. Besides, the scheme can support the dynamic addition of sensor nodes in the WSNs environment.

C7. **Provision of key agreement**: After authentication, a shared session key is established between the participants for subsequent secure communication.

C8. **No clock synchronization**: The proposed scheme should avoid clock synchronization.

C9. **Timely typo detection**: To reducing unnecessary communication cost, the user can be timely notified when she input a wrong password.

C10. **Mutual authentication**: The user side and server side can authenticate each other.

C11. **User anonymity**: The scheme should protect 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.

Remark: Both the evaluation criteria and the provable security are the answers to the question of how to design a secure multi-factor authentication scheme. The former has been extensively investigated and our taxonomy of the failures in security proofs (which will be shown in the next session) effectively supplements the current protocol evaluation criteria set. The reasons for the failures in the security proofs are divided into eight types. Combined with the existing 12 criteria, we propose more complex evaluation criteria[1]. However, we think this complexity is necessary and will make our criteria more comprehensive to be employed.

[1] In order to avoid repetition, we did not list them separately but used them directly in Section XI to evaluate the relevant protocols.

III. A TAXONOMY OF SECURITY PROOF FAILURES

Based on the analysis of more than 200 multi-factor authentication protocols, we investigate the causes and consequences of failures in security proofs, and classify them into eight different types (see Table I) in terms of the five steps of conducting the formal security proof.

As shown in Table I, we present the attacks directly related to each type of security proof failure. Note that each listed attack does not uniquely correspond to a type of failure in security proofs. On the contrary, a protocol with a certain type of failure in security proofs may not be vulnerable to the listed attacks. Essentially, various known attacks in multi-factor authentication schemes, such as smart card loss attacks, insider attacks, and user impersonation attacks, utilize various protocol design vulnerabilities. Exploiting different vulnerabilities would lead to different consequences and eventually be summarized into different attacks. Therefore, from this point of view, various known attacks are more like the explicit expression of the flaws in security proofs. To illustrate our taxonomy of failures in security proofs, we give the cryptanalysis of typical multi-factor authentication schemes [40]–[42] and point out their formal security proof flaws.

Type I∼Type III in Table I depict the security proof failures caused by the incorrect definition of the adversary models. A typical scheme of Type I is proposed by Srinivas et al. [42]. In this scheme, they treat biometrics as an absolutely secure authentication factor (i.e., a secure black box). However, biometrics can be obtained by adversaries in both reality and theory. In reality, malicious readers may extract the victims' fingerprint [48]; In theory, multi-factor security [28] requires that even if the adversary holds n-1 authentication factors at the same time, the security of the last factor cannot be affected. We show an offline password guessing attack directly related to this type of security proof failure in Section V-A.

A typical representative scheme of Type II is proposed by Fotouhi et al. [41]. In their scheme, Fotouhi et al. do not consider the security of their scheme under the case of user password disclosure. As a result, this scheme cannot resist the user impersonation attack given in Section VII-B. Type III represents a kind of insider attack. In these attacks, the server does not set separate authentication credentials for each user but uses a unified master key, which leads to a legitimate user can impersonate the server by calculating its master key. A typical representative scheme of Type III is also Srinivas et al.'s scheme [42], and we show this attack in Section V-C.

Type IV∼Type V in Table I occur at the stage of declaring security goals. There are four independent security goals, i.e., semantic security, authentication, key privacy, and password protection. Semantic security requires that an external attacker cannot distinguish the real session key from a random string of the same length in polynomial time; Authentication requires that the attacker cannot impersonate the real entity in the protocol; Key privacy requires that the session key established between the user and the SN is indistinguishable by the honest and curious server; Password protection refers to that the attacker cannot obtain any information of the user's password through protocol operation. These four goals are originally proposed by Abdalla et al. [49] for Gateway-oriented PAKE (GPAKE) scheme, but they are still suitable for password-based multi-factor authentication. Type IV represents proving the schemes' security under a wrong security goal. For example, Zhang et al. [40] prove their scheme under the security goal of semantic security, but their scheme cannot resist the password guessing attack shown in Section IX-A.

TABLE II: Notations and Abbreviations

Symbol	Description	Symbol	Description
U	user	SN	sensor node
U_i	the i^{th} user	SN_j	the j^{th} sensor node
GWN	gateway node	SC_i	U_i's smart card/device
$\mathcal{A}$	the adversary	ISD_j	the j^{th} IoT sensing device
ID_i	identity of U_i	PW_i	password of U_i
ID_{SN_j}	identity of SN_j	SK	session key
X_{GWN}	secret key of GWN	X_{SN_j}	secret key of SN_j
$\oplus$	bitwise XOR operation	$\|\|$	concatenation operation

Type V represents the security proofs that do not use an accurate password distribution model to describe the adversary's advantage. The distribution of passwords has been an open question for a long time. After declaring the security goals, the usual practice is to give the adversary's advantage function of the security parameter k. For ease of description, many protocols use the uniform distribution model to describe the password distribution [30], [32], [44]. However, Wang et al. [50] have uncovered Zipf's law in passwords and give the precise distribution function, but there are protocols [51]–[53] still use the imprecise password distribution model. Type V will not directly make the protocol unable to resist specific attacks but will lead to an imprecise security reduction result of the adversary's advantage, which is about 2 to 4 orders of magnitude lower than the actual distribution [50].

A typical representative scheme of Type VI is proposed by Li et al. [32]. In their scheme, Li et al. claim that the proposed scheme is AKA-secure if the hash function range and the password size are large. However, Ma et al. [54] have revealed three potential principles for multi-factor protocol design. They reveal that public key techniques are essential to resist offline password guessing attacks and provide user anonymity. In other words, the security assumptions based on the one-way hash function and the size of its output space cannot guarantee the protocol's security. As expected, Li et al.'s scheme cannot resist password guessing attacks.

Type VII describes that the attacker's advantage in password guessing is incorrectly calculated. A typical representative scheme of Type VII is still Srinivas et al.'s scheme [42], and we show this attack in Section V-B. In type VIII, some complex attacks cannot be described by provable security, such as man-in-the-middle attacks and de-synchronization attacks, which means that a scheme with correct security proof may still be unable to resist these attacks. We take Fotouhi et al.'s scheme [41] as an example and demonstrate the de-synchronization attacks in Section VII-C.

IV. REVIEW OF SRINIVAS ET AL.'S SCHEME

This section briefly reviews Srinivas et al.'s multi-factor authentication scheme [42]. This scheme comprises seven phases: pre-deployment, registration, log-in, authentication, password and biometric update, smart card revocation, and dynamic sensing device addition. Due to space constraints, we only show the log-in and the authentication phase.

A. Pre-deployment Phase

Before deploying the IoT sensing device ISD_j into the network, GWN uses its master secret key X_{GWN} to compute $S_{key_j} = h(SID_j \| X_{GWN})$. Then, GWN stores the credentials $\langle SID_j, S_{key_j} \rangle$ into ISD_j 's memory. After that, GWN calculates $Key_j = S_{key_j} \oplus X_{GWN}$ and stores the credentials $\langle SID_j, Key_j \rangle$ into its database.

B. Registration Phase

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

R1: U_i selects her identity ID_i, password PW_i, and generates random numbers b_i, m_{i1} and m_{i2}. Then, U_i computes $DID_i = h(ID_i \| b_i)$ and $DPW_i = h(ID_i \| \ PW_i)$, and submits the registration request $\langle DID_i \oplus m_{i1}, DPW_i \oplus m_{i2} \rangle$ secretly to the registered GWN.

R2: On receiving the request, the GWN checks the availability of DID_i in its user-list database. If DID_i is available, the GWN computes $C_i = (DID_i \oplus m_{i1}) \oplus (DPW_i \oplus m_{i2}) \oplus h(X_{GWN} \| h(X_{GWN-U_i}))$. The GWN issues a smartcard SC_i that contains $\{C_i, h(\cdot)\}$ to U_i.

R3: After receiving SC_i, U_i imprints her biometrics BIO_i, and computes $(\sigma_i, \tau_i) = \text{Gen}(BIO_i)$, $L_i = b_i \oplus h(\sigma_i \| PW_i)$, $RB_i = h(ID_i \| \sigma_i \| PW_i)$, $C_i' = (C_i \oplus m_{i1} \oplus m_{i2}) \oplus h(\sigma_i \| ID_i)$. Finally, U_i replaces C_i with C_i', and stores $RB_i, L_i, \text{Gen}(\cdot), \text{Rep}(\cdot)$, τ_i and t into SC_i to complete the registration process. As a result, $SC_i = \{L_i, RB_i, C_i', h(\cdot), \text{Gen}(\cdot), \text{Rep}(\cdot), \tau_i, t\}$.

C. Log-in Phase

U_i logs into the system through the following steps:

L1: U_i inserts her smartcard SC_i (SC_i stores the parameters $\{L_i, \ RB_i, C_i', h(\cdot), \text{Gen}(\cdot), \text{Rep}(\cdot), \tau_i, t\}$) into the card reader, then inputs the identity ID_i, password PW_i, and biometrics BIO_i'. After that, SC_i computes $DPW_i = h(ID_i \| PW_i)$, $\sigma_i^* = \text{Rep}\ (BIO_i', \tau_i)$ with the criteria that $\text{dis}(BIO_i, BIO_i') \leq t$, $b_i^* = L_i \oplus h(\sigma_i^* \| PW_i)$, and checks if the equation $RB_i = h(ID_i \| \sigma_i^* \| PW_i)$ holds.

L2: If the above equation holds, SC_i confirms that U_i 's entered credentials (ID_i, PW_i, BIO_i') are valid. Then, SC_i computes $C_i = C_i' \oplus h(\sigma_i^* \| ID_i)$, $DID_i = h(ID_i \| b_i^*)$, $J_i = C_i \oplus DID_i \oplus DPW_i$. After that, U_i chooses the identity SID_j of the IoT device she wishes to access.

L3: SC_i generates a random number r_i and current timestamp TS_1, and computes $E_i = h(J_i \| h(\sigma_i^* \| PW_i) \| TS_1)$, $A_g = T_{r_i}(DID_i \| SID_j \| E_i)$, $G_i = A_g \oplus h(DID_i \| J_i \| TS_1)$, $V_{GWN} = h(DID_i \| A_g \| G_i \| SID_j \| TS_1)$, $E_i' = E_i \oplus h(DID_i \| J_i \| TS_1)$, $DID_i' = DID_i \oplus h(E_i' \| J_i \| TS_1)$ and $SID_j' = SID_j \oplus h(DID_i \| TS_1)$, and sends the log-in message $MSG_1 = \{E_i', DID_i', V_{GWN}, G_i, SID_j', TS_1\}$ to the GWN over an open channel.

D. Authentication Phase

GWN authenticates the U_i, and a session key is established between U_i and the sensor node ISD_j. The following steps are essential to complete this phase:

A1: Upon receiving MSG_1 from U_i, GWN checks the freshness of the message by $|TS_1' - TS_1| < \Delta T$, where the maximum transmission delay is ΔT and the received time of the message is TS_1'. Then, GWN computes $M_i = h\ (X_{GWN} \| h(X_{GWN-U_i}))$, $DID_i = DID_i' \oplus h(E_i' || M_i \| TS_1)$, $A_g^* = G_i \oplus h(DID_i \| M_i \| TS_1)$, $SID_j = SID_j' \oplus h(DID_i \| TS_1)$, and checks if the equation $V_{GWN} = h(DID_i \| A_g^* \| G_i \| SID_j \| TS_1)$ holds.

A2: If the above equation holds, GWN continues to compute $E_i = E_i' \oplus h(M_i \| DID_i \| TS_1)$, fetches S_{key_j} and generates current timestamp TS_2, computes $SID_j'' = h(SID_j \| S_{\text{key}\ j} \| TS_2) \oplus DID_i$, $H_j = S_{key_j} \oplus A_g^*$, $V_{SN_j} = h(S_{key_j} \| SID_j \| A_g^* \| H_j \| TS_2)$, $E_i'' = E_i \oplus h(S_{key_j} \| TS_2)$ and then transmits the message $MSG_2 = \{H_j, V_{SN_j}, SID_j'', E_i'', TS_2\}$ to the target sensor node ISD_j.

A3: On receiving MSG_2, ISD_j checks the freshness of the message by $|TS_2' - TS_2| < \Delta T$, where TS_2' is the message reception time. After that, ISD_j computes $DID_i = h(SID_j \| S_{key_j} \| TS_2) \oplus SID_j''$, $E_i = E_i'' \oplus h(S_{key_j} \| TS_2)$, $A_g' = S_{key_j} \oplus H_j$ and checks the equation $V_{SN_j} = h(S_{key_j} \| SID_j \| A_g' \| H_j \| TS_2)$ holds or not. If the verification fails, ISD_j rejects MSG_2.

A4: ISD_j generates a random number r_j and current timestamp TS_3, and then computes $N_j = T_{r_j}(DID_i \| SID_j \| E_i)$, $SK_{ij} = h(T_{r_j}(A_g') \pmod p) \| DID_i \| TS_3)$ as the session key shared between U_i and ISD_j, $P_j = h\ (SK_{ij} \| N_j \| TS_3)$ and $N_j' = N_j \oplus h(DID_i \| SID_j \| TS_3)$. After that, ISD_j sends the message $MSG_3 = \{P_j, N_j', TS_3\}$ to U_i via open channel.

A5: U_i receives the MSG_3 and checks the freshness of the message by $|TS_3' - TS_3| < \Delta T$, where the reception time of the message is TS_3'. Then, SC_i computes $N_j = N_j' \oplus h(DID_i \| SID_j \| TS_3)$ and the session key $SK_{ij}^* = h(T_{r_i}(N_j) \pmod p) \| DID_i \| TS_3)$ shared with ISD_j to check if $P_j = h(SK_{ij}^* \| N_j \| TS_3)$ holds. If it holds, U_i authenticates ISD_j.

Finally, both U_i and ISD_j store the common session key $SK_{ij}^* = SK_{ij}$ for their future secure communication.

V. Analysis of Srinivas et al.'s Scheme

Srinivas et al. [42] propose this authentication scheme to protect the Industrial Internet of Things (IIoT) from being illegally accessed by an adversary. In order to illustrate the security of their scheme, they give formal security proof under the ROM model. However, based on our assumptions of the adversary, we will show that it fails to resist insider attacks and two kinds of smart card loss attacks. Further, we point out the flaws in Srinivas et al.'s formal proof.

A. Offline Password Guessing Attack I

Based on the capability A3 of the adversary, an attacker $\mathcal{A}$ can obtain the victim's biometrics by using malicious devices and the secrets stored in SC_i by side-channel attacks. The offline password guessing attack can be launched as follows:

Step 1. $\mathcal{A}$ computes $\sigma_i^* = \text{Rep}(BIO_i', \tau_i)$, where τ_i is extracted from the user's smart card SC_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 $RB_i^* = h(ID_i^*||\sigma_i^*||PW_i^*)$, and then, validates the correctness of (ID_i^*, PW_i^*) by comparing the calculated RB_i^* and the extracted RB_i.

Step 4. $\mathcal{A}$ will repeat the step 2∼3 until she finds the correct pair of (ID_i^*, PW_i^*).

The time complexity of this attack is $\mathcal{O}(|D_{pw}| \times |D_{id}| \times T_h)$, where $|D_{pw}|$ denotes the size of password space D_{pw}, $|D_{id}|$ denotes the size of identity space D_{id} and T_h denotes the running time for hash operation. Moreover, according to the capability A1, an determine adversary can obtain the victim's identity. Thus, the time complexity could be reduced to $\mathcal{O}(|D_{pw}| \times T_h)$, which is linear to $|D_{pw}|$. As mentioned above, the size of D_{pw} is limited and $D_{pw} \approx 10^6$ [50] in reality and hash function is also a lightweight operation. As a result, $\mathcal{A}$ could identify the correct password in polynomial time.

B. Offline Password Guessing Attack II

The reason for the above attack lies in the storage of the parameter RB_i. In this scheme, RB_i is used for preliminary detection of user input information to avoid the waste of computing resources and communication resources caused by the user's incorrect password input. This attack can be avoided by removing this parameter or employing the "fuzzy-verifier" [5] technique [36]. However, in the next attack, $\mathcal{A}$ can determine the victim's password without using RB_i.

Step 1. $\mathcal{A}$ computes $\sigma_i^* = \text{Rep}(BIO_i', \tau_i)$, where BIO_i' is the obtained victim's biometrics and τ_i is extracted from the user's smart card SC_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 $DPW_i^* = h(ID_i^*||PW_i^*)$ and $b_i^* = L_i \oplus h(\sigma_i^*||PW_i^*)$, where L_i is extracted SC_i.

Step 4. $\mathcal{A}$ computes $C_i^* = C_i' \oplus h(\sigma_i^*||ID_i^*)$, $DID_i^* = h(ID_i^*||b_i^*)$, $J_i^* = C_i^* \oplus DID_i^* \oplus DPW_i^*$ and $E_i^* = h(J_i^*||h(\sigma_I^*||PW_i^*)||TS_1)$, where C_i' is extracted from the user's smart card SC_i and TS_1 is obtained from previously intercepted transcripts.

Step 5. $\mathcal{A}$ computes $E_i'^* = E_i^* \oplus h(J_i^*||DID_i^*||TS_1)$, and then, validates the correctness of (ID_i^*, PW_i^*) by comparing the calculated $E_i'^*$ and the E_i' obtained from previously intercepted transcripts.

Step 6. $\mathcal{A}$ will repeat the step 2∼5 until she finds the correct pair of (ID_i^*, PW_i^*).

The time complexity of this attack is $\mathcal{O}(|D_{pw}| \times |D_{id}| \times 7T_h)$. The compute cost of the $\oplus$ operation can be omitted.

C. Insider Attack

According to the capability A6 of the adversary, an attacker could be a legitimate user of the system. Suppose $\mathcal{A}$ is a registered user and she can obtain the private key S_{key_j} of the target sensing device SID_j through the following attack:

Step 1. $\mathcal{A}$ inserts her smart card SC_A into the card reader, and then, inputs her identity ID_A, password PW_A and biometrics BIO_A'. After that, $\mathcal{A}$ can log-in into the system as a normal user.

Step 2. $\mathcal{A}$ eavesdrops on the open channel and obtains the transmitted message $MSG_2 = \{H_j, V_{SN_j}, E_i'', TS_2, SID_j''\}$ between the target IoT sensing device ISD_j and the gateway node GWN.

Step 3. $\mathcal{A}$ computes $S_{key_j} = H_j \oplus A_g$, where the parameter H_j is extracted from the message MSG_2 and the parameter A_g is calculated by $\mathcal{A}$ herself.

After acquiring the private key S_{key_j} of the ISD_j, $\mathcal{A}$ may help an unregistered user to access the sensing device ISD_j, since $\mathcal{A}$ can forge the message MSG_2. Besides, using the secret key S_{key_j}, $\mathcal{A}$ can impersonate the ISD_j to communicate with other legitimate users. What is worse, through the above attack steps, the adversary $\mathcal{A}$, as a legitimate user of the system, can obtain the private key of any target IoT sensing device she wants.

D. Flaws in the Formal Security Proof

Srinivas et al. formally prove their scheme under the ROM model. However, their scheme still suffers from three kinds of attacks, i.e., offline password guessing attacks I, offline password guessing attacks II, and insider attacks, which correspond to the security proof failure type I, type VII, and type III in Table I. Next, we will show the flaws in their security proof.

First of all, we examine the formal proof of Srinivas et al.'s scheme. The tactic adopted by them consists of five independent games (i.e., Gm_0, ..., Gm_4), where the Gm_0 represents the real attack of the adversary; Gm_1 simulates the passive eavesdropping of the adversary on the open channel; Gm_2 simulates the active attack of the adversary; Gm_3 is used to simulate the loss of the smart card, and finally ends in Gm_4 with the adversary's advantage is 0. The threat model adopted by Srinivas et al. has considered that the smart card could be lost and the user credentials, such as identity, password, and biometrics, can be extracted (see Section 1.2.2 in [42]).

However, when conducting the security proof, they only define the *CorruptSmartcard(I)* query, and there is no query related to the biometric factor. They treat the biometric factor as a secure "black box" (*Type I in Table I*), and the only way for $\mathcal{A}$ to break this factor is to guess the l-bits biometrics key σ_i with the probability of $\frac{1}{2^l}$. Such reduction is not rigorous in two aspects. For one thing, biological information can be obtained through malicious devices; For another, multi-factor security requires that each authentication factor is independent and can provide corresponding security, that is, when the adversary holds $n-1$ authentication factors, the n-th authentication factor cannot be obtained by calculation. The offline password guessing attack I in V-A shows that once the user's smart card and the biometric factor are compromised, the user's password can be successfully guessed.

Besides, when giving the adversary's advantage of password guessing in Gm_3, Srinivas et al. do not correctly reduce it (*Type VII in Table I*). They make a rash conclusion that $\mathcal{A}$ requires both the secret credentials b_i and biometric key σ_i. They believe the probability of successfully guessing the user's password is at most $\frac{q_s}{|D|}$, where q_s denotes the number of *Send(I, M)* queries and $|D|$ denotes the size of password space. While the offline password guessing attack II in V-B shows that both the b_i and the σ_i can be calculated. As a result, $\mathcal{A}$ can successfully determine the user's correct password.

The security proof failure *type III in Table I* is reflected in Gm_1 of the Srinivas et al.'s proof process. Gm_1 is modeled as an eavesdropping attack, thus, all of the communicated messages among U_i, GWN, and ISD_j are intercepted by $\mathcal{A}$, when she launches the *Execute(I)* query. If $\mathcal{A}$ wants to derive the session key SK_{ij}, she needs the temporal secrets r_i, r_j and long-term secrets $DID_i, SID_j, X_{GWN}, X_{GWN-U_i}$. Srinivas et al. believe that the $\mathcal{A}$'s advantage in winning Gm_1 has not increased, because the intercepted messages $MSG_1 \sim MSG_3$ do not lead to compromise any one of the temporal/long-term secret credentials. Unfortunately, their adversary model does not consider an attacker who is a legitimate user, and this attacker could obtain the temporal/long-term secret credentials. Essentially, the GWN confirms the legitimacy of the user U_i by checking if she holds the long-term secret X_{GWN-U_i}. That is, U_i can calculate the correct A_g; GWN confirms the legitimacy of the ISD_j by checking if it holds the long-term secret X_{GWN}, i.e., the key S_{key_j} stored in ISD_j in advance.

As a legitimate user, $\mathcal{A}$ can calculate the correct A_g and use it to calculate the ISD_j's long-term key S_{key_j}. As a result, $\mathcal{A}$ can finally impersonate the ISD_j or compute any target IoT sensing device's long-term key S_{key_k}.

VI. Review of Fotouhi et al.'s scheme

In this section, we briefly review the two-factor authentication scheme for wireless body area networks proposed by Fotouhi et al. [41]. This scheme consists of four phases: initialization, registration, authentication, and password change phase. Due to space constraints, we omit the last phase.

A. Initialization Phase

In this scheme, the gateway is assumed to be a trusted party. The gateway identified with GID_j generates a secret key G_j and selects a collision-resistant hash function $h(\cdot)$ to initialize the wireless body area networks.

B. Sensor Registration Phase

Each sensor node called SN_k has an identifier SID_k. In addition, each set of sensors that belong to the same network have uniform network identifier N_1. Before deploying into the network, the corresponding gateway GWN_j compute a shared secret key $SG_k = h(SID_k \| G_j \| N_1)$ for each sensor node. Then, GWN_j selects two random numbers R_y, R_z, and a pseudo-identity QID_k for each sensor. After that, GWN_j injects $(SID_k, SG_k, GID_j, R_y, R_z, QID_k)$ into each sensor node's memory. GW_j also stores $(SID_k, QID_k, N_1, R_y, h(R_z))$ in its database.

C. User Registration Phase

The two authentication factors are the user selected password PW, and the mobile device with some secret information stored, respectively. In this phase, a user registers to the gateway GWN_j in following three steps.

R1: U_i selects an identity ID_i, a password PW_i and a random nonce R_0 to compute $HPW_i = h(PW_i \| R_0)$. Then, she sends ID_i and HPW_i to GWN_j via a secure channel.

R2: If ID_i is unregistered, GWN_j generates a pseudo-identity CID_i and a random number R_x for U_i, and then, GWN_j stores them with ID_i and HPW_i in its database. Then, it computes $A_1 = h(CID_i \| R_x \| GID_j \| G_j) \oplus HPW_i$ and $A_2 = h(ID_i \| G_j) \oplus h(ID_i \| HPW_i)$ and sends $A_1,\ A_2, CID_i, GID_j$ to U_i via a secure channel.

R3: U_i calculates $A_3 = h(ID_i \| PW_i) \oplus R_0$ and stores $A_1,\ A_2,\ A_3, CID_i$, GID $_j$ into her mobile device.

D. Authentication Phase

There are five steps in the authentication phase. Through this phase, GWN_j authenticates the U_i and establishes a shared session key between the user U_i and the sensor node SN_k.

A1: U_i inputs ID_i and PW_i to the mobile device. Then, the mobile device calculates $R_0 = A_3 \oplus h(ID_i \| PW_i)$ and $HPW_i = h(PW_i \|\ R_0)$. Next, it generates a random number R_u, selects the target sensor node's SID_k and computes $B_1 = A_1 \oplus HPWi$, $B_2 = B_1 \oplus HPWi \oplus R_u$, $B_3 = SID_k \oplus h(ID_i \| R_u)$, $B_4 = h(CID_i \| GID_j \| SID_k \| B_1 \| ID_i \| R_u)$. Finally, the mobile device sends the message $M_1 = \{CID_i, GID_i, B_2, B_3, B_4\}$ to GWN_j.

A2: On receiving the message M_1, GWN_j checks GID_j and CID_i and fetches the corresponding ID_i, R_x and HPW_i from its database. Then, it computes $B_1 = h(CID_i \| R_x \| GID_j \| G_j)$, $R_u = B_2 \oplus B_1 \oplus HPW_i$, and generates two random numbers R_g and R_z^{new}. After that, GWN_j computes $SID_k = B_3 \oplus h(ID_i \| R_u)$ and gets R_y from its database. Then, it generates a new pseudonym QID_k^{new} for the sensor. In the following, GWN_j computes $SG_k = h(SID_k \| G_j \| N_1)$, $S = h(SG_k \| GID_j)$, $B_5 = (R_u \oplus HPW_i) \oplus S \oplus R_y$, $B_6 = R_g \oplus S \oplus SID_k \oplus R_y$, $B_7 = QID_k^{new} \oplus R_g \oplus\ R_y$, $B_8 = h(R_g \| R_y \| S) \oplus R_z^{\text{new}}$, $B_9 = h(QID_k \| B_7 \| B_8 \| SG_k \| R_u \oplus HPW_k \| R_g)$ and sends $M2 = \{OID_k, B_5, B_6, B_7, B_8, B_9\}$ to the sensor.

A3: After receiving the message, the sensor first checks the QID_k and calculates $S = h(SG_k \| GIDj)$, $(R_u \oplus HPW_i) = B_5 \oplus S \oplus R_y$ and $R_g = B_6 \oplus S \oplus SID_i \oplus R_y$. Then if B_9 is correct, it generates a random number R_s and computes $R_z^{\text{new}} = h(R_g \| R_y \| S) \oplus B_8$, $QID_k^{\text{new}} = B_7 \oplus R_g \oplus R_y$. After generating $B_{10} = R_g \oplus S \oplus R_z$, it stores QID_k^{new}, R_z^{new} and $R_y^{\text{new}} = h(R_y)$ and generates the common session key $sk_s = h(R_u \oplus HPW_i \| R_g \| R_s)$. Finally, the sensor generates $B_{11} = h(SG_k \| R_g) \oplus\ h(R_y) \oplus R_s$, $B_{12} = h(B_{10} \| B_{11} \| sk_s \| \text{SID}_k \| GID_j \| R_s)$ and sends $M_3 = \{B_{10}, B_{11}, B_{12}\}$ to the GWN_j as the response.

A4: On receiving the response from sensor, the GWN_j computes $R_y' = h(R_y)$, where R_y is fetched from GWN_j's database, and $R_z' = R_g \oplus S \oplus B_{10}$. Then, GWN_j checks if the equation $h(R_z) = h(R_z')$ holds, it calculates $R_s = B_{11} \oplus h(SG_k \| R_g) \oplus R_y'$ and obtains the $sk_g = h(R_u \oplus HPW_i \| R_g \| R_s)$ as the common session key. Then, it checks B_{12} and generates a new CID_i for U_i and stores QID_k^{new} and R_z^{new}. In addition, it replaces the R_y' with the previous R_y and $h(R_x)$ with the R_x. Next, GW_j computes $B_{13} = h(CID_i^{new} \| h(R_x) \| GID_j \| G_j) \oplus h(R_u \| HPW_i)$, $B_{14} = h(R_u \| ID_i) \oplus R_g$, $B_{15} = h(R_u \| R_g \| HPW_i) \oplus R_s$, $B_{16} = h(h(ID_i \| G_j) \| R_s) \oplus CID_i^{new}$, $B_{17} = h(sk_g \| ID_i \| B_{13} \| CID_i^{new})$ and sends $M_4 = \{B_{13}, B_{14}, B_{15}, B_{16}, B_{17}\}$ to U_i.

A5: U_i computes $R_g = B_{14} \oplus h(R_u \| ID_i)$, $R_s = B_{15} \oplus h(R_u \| R_g \| HPW_i)$ and U_i 's new pseudonym from $CID_i^{new} = B_{16} \oplus h((A_2 \oplus h(ID_i \| HPW_i)) \| R_s)$. Then it computes $sk_u = h(R_u \oplus HPW_i \| R_g \| R_s)$ to get the common session key. After checking B_{17}, it stores CID_i^{new} and A_1^{new} which are derived by calculating $A_1^{new} = B_{13} \oplus h(R_u \| HPW_i)$.

VII. Analysis of Fotouhi et al.'s scheme

Fotouhi et al. [41] claim that the proposed scheme is secure against various known attacks and give the formal proof under the ROM model. However, we find this scheme cannot resist the insider attack and the de-synchronization attack. Also, we point out the flaws in their formal proof.

A. Offline Password Guessing Attack

Based on the capability A6 of the adversary, an attacker could be an administrator of the GWN_j, so she can obtain the parameters stored in the GWN_j's database. Meanwhile, she can also obtain the secure parameters stored in the user's mobile device, according to the capability A3. Thus, for a potential victim user U_i, the adversary $\mathcal{A}$ can determine her password PW_i as follows:

Step 1. $\mathcal{A}$ fetches the vicitm's identity ID_i and the corresponding parameter HPW_i.

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

Step 3. $\mathcal{A}$ computes $R_0^* = A_3 \oplus h(ID_i||PW_i^*)$ and $HPW_i^* = h(PW_i^*||R_0^*)$, where the parameter A_3 is extracted from the U_i's mobile device.

Step 4. $\mathcal{A}$ validates the correctness of PW_i^* by checking the equation $HPW_i = HPW_i^*$ holds or not.

Step 5. $\mathcal{A}$ will repeat the step 2∼4 until she finds the correct password PW_i^*.

The time complexity of this attack is $\mathcal{O}(|D_{pw}| \times 2T_h)$. As an administrator of the GWN_j, $\mathcal{A}$ can directly choose the identity of the target victim and enjoy more convenience than external attackers in determining the user password.

B. User Impersonation Attack

Similar to above password guessing attack, the attacker $\mathcal{A}$ is also an administrator of the GWN_j, so she knows the victim's identity ID_i and the corresponding parameter HPW_i. Besides, according to the capability A5 of the adversary, suppose $\mathcal{A}$ compromise the SN_k and she has the parameters $\{SID_k, SG_k, GID_j, R_y, R_z, QID_k\}$ stored in it. Then, $\mathcal{A}$ impersonates the U_i as follows:

Step 1. $\mathcal{A}$ eavesdrops on the public channel and obtains messages $M_1 = \{CID_i, GID_i, B_2, B_3, B_4\}$ and $M2 = \{QID_k, B_5, B_6, B_7, B_8, B_9\}$.

Step 2. $\mathcal{A}$ computes $S = h(SG_k||GID_j)$ and $R_g = B_6 \oplus S \oplus SID_k \oplus R_y$, where SG_k, GID_j, R_y are extracted from the memory of SN_k and B_6 is from M_2.

Step 3. $\mathcal{A}$ computes $B_1 \oplus R_g \oplus SID_k = B_2 \oplus B_5 \oplus B_6$ and $B_1 = B_1 \oplus R_g \oplus SID_k$, where the parameter B_2 is extracted from the message M_1.

Step 4. $\mathcal{A}$ intercepts the massage M_2 and generates a new random number R_a. Then, $\mathcal{A}$ choose another sensor node SID_l and computes $B_2^* = B_1 \oplus HPW_i \oplus R_a$, $B_3^* = SID_l \oplus h(ID_i||R_a)$, and $B_4^* = h(CID_i||GID_j||SID_l||B_1||ID_i||R_a)$.

Step 5. $\mathcal{A}$ sends the new message $M_1^* = \{CID_i, GID_j, B_2^*, B_3^*, B_4^*\}$ to the GWN_j.

Two important parameters in Fotouhi et al.'s scheme, B_1 and HPW_i, represent two authentication factors, i.e., mobile devices and passwords, respectively. With these two parameters, $\mathcal{A}$ can impersonate U_i, since the M_1 sent by $\mathcal{A}$ could pass GWN_j's verification. Further, $\mathcal{A}$ can obtain the GWN_j's updates on all parameters (i.e., CID_i and B_1) of U_i.

C. De-synchronization Attack

Based on the capability A2 of the adversary, the attacker has full control of the public channel, thus, she can intercept the massages transmitted among the participants. In this scheme, Fotouhi et al. applies the hash-chain to achieve the forward security. Specifically, after each communication, the sensor node and the GWN_j will update the secret parameter R_y by computing $R_y^{new} = h(R_y)$. The shared session key between the user and sensor node is calculated as $Sk_s = h(R_u \oplus HPW_i||R_g||R_s)$, where R_g and R_s are two random numbers generated by GWN_j and SN_k, respectively. To protect these two parameters, GWN_j computes $B_6 = R_g \oplus S \oplus SID_i \oplus R_y$ and SN_k computes $B_{11} = h(SG_k||R_g) \oplus h(R_y) \oplus R_s$.

The de-synchronization attack can be launched as follows, $\mathcal{A}$ intercepts the massage $M_3 = \{B_{10}, B_{11}, B_{12}\}$, which will lead to the authentication session time out because GWN_j cannot receive the response from SN_k. After that, when GWN_j communicates with SN_k next time, the session will be reject by SN_k. Because the parameter R_y stored in SN_k has been updated as $R_y^{new} = h(R_y)$, while GWN_j still use the old R_y to compute $B_6 = R_g \oplus S \oplus SID_k \oplus R_y$. As a result, SN_k cannot obtain R_g through computing $R_g = B_6 \oplus S \oplus SID_k \oplus R_y^{new}$ and cannot compute the correct B_9. Further, SN_k cannot authenticate GWN_j.

D. Flaws in the Formal Security Proof

Fotouhi et al. formally prove their scheme under the ROM model. However, this scheme still cannot resist three kinds of attacks, i.e., the offline password guessing attack, the user impersonation attack, and the de-synchronization attack, where the user impersonation attack and the de-synchronization attack correspond to the security proof failure type II and type VIII that are shown in Table I.

In this section, we will show the flaws in their security proof. First of all, we examine the formal proof of Fotouhi et al.'s scheme. The tactic adopted by them consists of three independent games (i.e., Game$_0^{\mathcal{A}}$, Game$_1^{\mathcal{A}}$, Game$_2^{\mathcal{A}}$), where the Game$_0^{\mathcal{A}}$ represents the real attack of the adversary; the Game$_1^{\mathcal{A}}$ simulates the passive eavesdropping of the adversary on the open channel; the Game$_2^{\mathcal{A}}$ simulates the active attack, such as hash query, compromising the user's mobile devices, and compromising the sensor nodes, launched by the adversary.

Fotouhi et al. adopted the threat model that the security parameters stored in the user's mobile devices and sensor nodes could be extracted by the adversary (see [41] Section 2.1.2). They model the adversary's capability by defining various queries. Specifically, they use *CorruptMobileDevice(U_i)* query to model the compromise of the user's mobile devices; and use *CorruptSensor(SN_k)* query to model the compromise of the sensor node. But they do not define the query used to model the user password disclosure (*Type II in Table I*). As we know, passwords are low-entropy strings that can be easily leaked, which means that Fotouhi et al. define an incomplete adversary model. As we showed in Section VII-B, the two authentication factors in this scheme, i.e., mobile devices and the user password, are represented by parameters

B_1 and HPW_i. In this case, an adversary who holds the user's password HPW_i can successfully impersonate the user because she can calculate the parameter B_1, and thus, she holds all two authentication factors.

VIII. Review of Zhang et al.'s scheme

In this section, we briefly review the multi-factor authentication scheme proposed by Zhang et al. [40].

A. Initialization Phase

There are three ingredients in Zhang et al.'s protocol. Specifically, a public key encryption scheme {PKE. KeyGen, PKE. Enc, PKE. Dec}, a message authenticated code scheme {MAC. KeyGen, MAC. Mac, MAC. Vrfy}, and a fuzzy extractor algorithm {Gen, Rep}. With a security parameter λ, the initialization phase performs as follows:

I1: The server selects a cyclic group $\mathbb{G} = \langle H \rangle$ with prime order p, where H is a generator of $\mathbb{G}$;

I2: Run PKC.KeyGen $\left(1^{\lambda}\right)$ to obtain a tuple of (PK, SK);

I3: The public parameters $\mathrm{Para} = ((\mathbb{G}, p, \mathrm{H}), \mathrm{PK})$, and the private parameter SK is the secret key.

B. Registration Phase

The registration phase performs in a secure environment. A user interacts with the server as follows:

R1: The user U_i randomly chooses a password PW_i from the distribution $\mathcal{D}_{pwd}$, which is denoted as α;

R2: U_i creates a good biometrics template W;

R3: The server runs $\mathrm{Gen}(W)$ to obtain a random number β ($\beta \in$ distribution $\mathcal{D}_{bio\_data}$) and an auxiliary string T, then deletes the template W;

R4: U_i randomly chooses an element from group $\mathbb{Z}_p^*$ (as distribution $\mathcal{D}_{\text{device \_ data}}$) denoted as γ;

R5: U_i may also need to input other information according to different scenes, all denoted as $userinfo$;

R6: The server computes $\mathrm{Z} = \mathrm{H}^{(\alpha+\beta+\gamma)}$, deletes β, runs PKC.Enc(PK,(Z, $userinfo$)) to obtain $SData$, generates an unique identifier ID_i corresponding to the U_i's identity, and stores a record $(ID_i, SData)$ into database;

R7: U_i stores the algorithm Rep and the parameters $(\gamma, T, Para)$ into her mobile device.

C. Login-Authentication Phase

In this phase, a user U_i with identity ID_i uses a registered device and sends an log-in request to server. After registration, U_i and server holds (α, β, γ) and Z respectively, where $\mathrm{Z} = \mathrm{H}^{(\alpha+\beta+\gamma)}$. Then, the server verifies the U_i as follows:

L1: To begin with, the user U_i sends her identity ID_i to the server as a log-in request.

L2: Server selects four random elements r, r', k and d' from group $\mathbb{Z}_p^*$, a random nonce $N_1 \in \{0,1\}^{\lambda}$; and computes $\mathrm{U}{=}\mathrm{H}^r$, $\mathrm{U}' = \mathrm{H}^{r'}$, $\mathrm{V} = \mathrm{H}^k$, $\mathrm{C}' = \mathrm{Z}^{r'}\mathrm{H}^{d'}$. Then, server gets current sid, which is used to mark this unique session. After that, the server sends $(\mathrm{U}, \mathrm{U}', \mathrm{V}, \mathrm{C}', N_1, \mathrm{sid})$ as an authentication challenge message to the user U_i.

L3: After receiving the authentication challenge, U_i random chooses two elements d and k' from group $\mathbb{Z}_p^*$, a random nonce $N_2 \in \{0,1\}^{\lambda}$ and computes $\mathrm{V}' = \mathrm{H}^{k'}$, $\mathrm{C} = \mathrm{U}^{(\alpha+\beta+\gamma)}\mathrm{H}^d$. It can now compute $\mathrm{K} = \mathrm{V}^d \oplus (\frac{\mathrm{C}'}{\mathrm{U}^{(\alpha+\beta+\gamma)}})^{k'}$. Lets $m_0 = \mathrm{U} \| \mathrm{U}' \| \mathrm{V} \| \mathrm{C}' \| N_1 \| sid$, then run MAC.Mac$_K(m_0)$ to obtain a tag τ_0. Finally it sends $(\mathrm{V}', \mathrm{C}, N_2, \mathrm{sid})$ as authentication response and τ_0 as authentication confirmation to server.

L4: On receiving the authentication response, server computes $\mathrm{K}' = \left(\frac{\mathrm{C}}{\mathrm{Z}^r}\right)^k \oplus \mathrm{V}'^{d'}$. Let $m_1 = (\mathrm{V}', \mathrm{C}, N_2, sid)$. Server runs MAC.Mac$_{K'}(m_1)$ to obtain a tag τ_1, and sends τ_1 as authentication confirmation to U_i.

L5: Both sides now have the pairs of (τ_0, m_0) and (τ_1, m_1). Server runs MAC. Verify$_{K'}(\tau_0, m_0)$, if the output is 1, then the server authenticate the identity of the U_i, else rejects; U_i runs MAC. Verify$_K(\tau_1, m_1)$, if the output is 1, then U_i accepts the server's identity, else rejects.

IX. Analysis of Zhang et al.'s scheme

Zhang et al. [40] propose this scheme to provide efficient multi-factor security. Also, they define security and give formal security proof under the ROM model. However, we find that a password guessing attack exists in Zhang et al.'s scheme, and we point out the flaws in their security proof.

A. Password Guessing Attack

Based on the capability A3 of the adversary, we suppose there is an attacker who has obtained the user's biometric factor β and the device factor γ. Then, she can launch the password guessing attack as follows:

Step 1. $\mathcal{A}$ selects four random elements r_a, r_a', k_a and d_a' from group $\mathbb{Z}_p^*$, a random nonce $N_a \in \{0,1\}^{\lambda}$; and computes $U_a{=}H^r$, ${U_a}' = H^{r'}$, $V_a = H^k$.

Step 2. $\mathcal{A}$ computes $C_a' = Z_a^{r_a'}H^{d_a'}$, where $Z_a = H^{(\beta+\gamma)}$.

Step 3. $\mathcal{A}$ sends the authentication challenge $\{U_a, {U_a}', (V_a, C_a'), N_a, sid\}$ to the U_i.

Step 4. On receiving the challenge, the U_i will execute the protocol honestly and random chooses two elements d and k' from group $\mathbb{Z}_p^*$, a random nonce $N_2 \in \{0,1\}^{\lambda}$ and computes $V' = H^{k'}$, $C = U^{(\alpha+\beta+\gamma)}H^d$. Then, she computes $K^* = V^d \oplus (\frac{{C_a}'}{{U_a}^{(\alpha+\beta+\gamma)}})^{k'}$, $m_0 = U \| U' \| V \| C' \| N_1 \| sid$, and τ_0 =MAC.Mac$_K(m_0)$.

Step 5. U_i sends the authentication response $\{(V', C) \| N_2 || sid || \tau_0\}$ to the server.

Step 6. $\mathcal{A}$ guesses the U_i's password α^* from the password dictionary space D_{pw}.

Step 7. $\mathcal{A}$ computes the parameter $Z_{a_1}^{r_a} = H^{(\alpha^*+\beta+\gamma)r_a}$ and $K_a = (\frac{C}{Z_{a_1}^{r_a}})^{k_a} \oplus \frac{V'^{d'}}{V'^{r'\alpha^*}}$.

Step 8. $\mathcal{A}$ validates the correctness of α^* by running MAC. Verify$_{K_a}(\tau_0, m_0)$. If the output is 1, $\mathcal{A}$ gets the correct password. Otherwise, $\mathcal{A}$ repeats the step 6∼8 until she finds the correct password α^*.

The main idea of this attack is to regarded user side as an oracle, which could be used by an adversary to determine the user's password α. In step 2, $\mathcal{A}$ sets $\alpha = 0$, which can reduce the adversary's calculation. Then, $\mathcal{A}$ computes $Z_a = H^{(\beta+\gamma)}$ and $C_a' = Z_a^{r_a'}H^{d_a'}$. With the received C_a', the calculation of the user's message authenticated code (MAC) key K is fixed, i.e., $K = H^{kd} \oplus (H^{d'-r'\alpha})^{k'}$. Further, it can be changed into $K_a = (\frac{C}{Z_{a_1}^{r_a}})^{k_a} \oplus \frac{V'^{d'}}{V'^{r'\alpha^*}}$, where the user's password α is the only unknown parameter. Although this attack looks complicated, the first five steps only need to be performed once, so it can be neglected in practice. The time complexity of this attack is $\mathcal{O}(|D_{pw}| \times (5T_m + T_{MAC}))$, where T_m is the time complexity of point multiplication and T_{MAC} is the time complexity of MAC. Both of these two operations are lightweight (see Section 6.2 of [40]).

B. Flaws in the Formal Security Proof

Zhang et al. give the formal proof of their scheme under the ROM model, while their scheme is still not immune to offline password guessing attacks. This section will show the flaws in Zhang et al.'s security proof. As usual, we examine the formal proof of Zhang et al.'s scheme. The tactic they adopted consists of two independent steps (i.e., Step 1 and Step 2), where Step 1 simulates the passive eavesdropping of the adversary on the open channel; Step 2 simulates the active attack that $\mathcal{A}$ can play the roles of server and client respectively.

The threat model adopted by Zhang et al. has considered the multi-factor security. Specifically, there are three authentication factors in this scheme, namely password, user devices and biometrics, which are represented by α, β and γ, respectively. In step 2, Zhang et al. assume that the adversary $\mathcal{A}$ can hold at most two of the three authentication factors at the same time. Since only partial information of the core parameter $Z = H^{(\alpha+\beta+\gamma)}$, used for constructing the session key, can be obtained, the adversary $\mathcal{A}$ can calculate the session key only by correctly guessing the remaining authentication factor.

Compared with the formal proof of Srinivas et al.'s scheme and Fotouhi et al.'s scheme, the definition of the adversary model in Zhang et al.'s scheme is harsher and there are no obvious errors in their proof process. However, their scheme still suffers from the offline password guessing attack shown in Section IX-A. This is caused by Zhang et al.'s declaration of a wrong security goal (*Type IV in Table I*). Essentially, the formal proof process of Zhang et al. serves the security goal of "semantic security" (see [40] Section 5), which requires that an adversary cannot distinguish the session key from a random string of the same length in polynomial time. The proposed attack can assist $\mathcal{A}$ to uniquely determine the user's password through the execution of the protocol, which violates the requirement of the security goal of "password protection".

X. Suggestions to the Provable Security

After decades of intensive research, provable security has become an indispensable tool in showing the security of a newly proposed cryptographic protocol. If the protocol evaluation criteria are the "touchstone" of the protocol design, then provable security is the safeguard of the protocol design. However, this safeguard is not a panacea, which cannot make a vulnerable protocol secure, and there may even be flaws in the formal security proof. According to the taxonomy of security proof failures in this paper, we find that the incompletely defined adversary models, the incorrectly declared security goals, and the complex attacks that are difficult to capture in existing models are the most common and direct reasons for the failures of security proofs. In this section, we will give some suggestions to deal with these three issues.

Define a complete adversary model: Defining the adversary model is the first step of the formal security proof. The adversary model in conventional PAKE assumed that $\mathcal{A}$ can eavesdrop, intercept, alter or insert messages exchanged among the communication parties, which accords to the widely accepted Dolev-Yao model [47]. However, with the increased number of authentication factors, this assumption is inadequate for capturing $\mathcal{A}$'s capabilities in multi-factor authentication environments. Notably, user passwords, smart cards (user devices), and biometrics are all independent authentication factors, and each factor should provide corresponding security. This is the essential difference between multi-factor and password-only protocols and the most significant advantage of multi-factor authentication protocols. Multi-factor security requires that $\mathcal{A}$ cannot calculate unknown authentication factors through the compromised authentication factors.

In recent years, side-channel attacks have developed rapidly [19], [110], and the conditionally tamper-resistance assumption [5], [35] for smart cards (user devices) has been widely accepted. As a result, most of the protocols take this into account and define the corresponding oracle query *CorruptSmartCard(I)* (*I* represents a communication instance), but the passwords and biometrics may be neglected, which is also reflected in the typical protocols we analyzed. Therefore, when defining the adversary models for a multi-factor authentication protocol, we suggest defining a corresponding corrupt query for each authentication factor (i.e., use *CorruptPW(I)* query to obtain the passwords factor, use *CorruptBio(I)* query to obtain the biometrics factor). Then, when conducting the formal security proof for an n-factor authentication scheme, the adversary is allowed to ask at most $n-1$ corrupt queries to meet the multi-factor security requirements. For example, when proving the security of a three-factor authentication scheme with passwords, biometrics, and smart cards, suppose that the adversary can ask *CorruptBio(I)* query and *CorruptSmartCard(I)* query to obtain the user's biometrics factor and security parameters stored in the smart card, respectively, and then give the adversary's advantages in this case.

Consider comprehensive security goals: In Section 5.4 of the [35], Wang et al. have discussed the role of provable security in the authentication protocol design. They believe that the security requirements that can resist certain attacks for most protocols can eventually be attributed to two security goals, namely, "semantic security" and "mutual authentication" (Authentication). According to the taxonomy of failures in security proofs shown in Table I, five of eight security proof failures are related to offline password guessing attacks, which also shows the importance of "password protection". In Section IX-B, we point out the flaws in Zhang et al.'s scheme, and their formal proof shows that under an inappropriate security goal, the security of the proposed scheme cannot be guaranteed even if the proof is correct.

Indeed, our analysis for Zhang et al.'s scheme [40] is some kind of an afterthought, but proposing a protocol and then breaking as well as repairing it is the research rhythm in multi-factor authentication [5]. The protocol's security is measured according to its effectiveness against a variety of known attacks, which usually depends on the experiences of the protocol designers (i.e., whether they can be aware of the potential threats). Therefore, when conducting the formal proof, it is recommended that the protocol designers can declare multiple security goals. For example, declare two security

TABLE III: Security and Efficiency Comparison among Relevant User Authentication Schemes

Protocol	Year	Ref.	Evaluation Criteria*																				Computational Cost⋆		
			C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	Proof success/failure								User	GWN/Server	Sensor Node
															I	II	III	IV	V	VI	VII	VIII			
Guo et al.	2022	[55]	√	√	√	×	×	√	√	×	√	√	×	×	×	√	√	√	×	×	√	√	$28T_H$	$46T_H$	$22T_H$
Xia et al.	2022	[56]	√	√	√	×	×	√	√	×	√	√	×	×	√	√	√	√	√	×	×	√	$T_B + 11T_H + T_S$	$10T_H + 4T_S$	$T_B + 2T_S + 6T_H$
Sutrala et al.	2022	[57]	√	√	√	×	×	√	√	×	√	√	×	√	×	√	√	×	√	√	×	√	$16T_H + T_B + 5T_P$	$9T_H + 3T_P$	$8T_H + 4T_P$
Rafique et al.	2022	[58]	√	√	√	×	×	√	√	×	√	√	×	×	×	√	√	√	√	√	×	√	$T_S + T_B + 10T_H$	$4T_S + 8T_H$	$4T_H + 2T_S$
Zhang et al.	2022	[59]	√	√	√	×	×	×	√	√	√	√	√	×	√	×	√	×	×	×	×	×	$T_B + 5T_H + 2T_S$	$T_B + 4T_H + 6T_S$	$T_B + 2T_H + 2T_S$
Li et al.	2022	[60]	√	√	√	×	×	√	√	×	√	√	×	×	√	√	√	√	√	×	×	×	$T_B + 11T_H$	$13T_H$	$6T_H$
Srinivas et al.	2021	[33]	√	√	√	×	×	√	√	×	√	×	×	×	×	√	√	√	√	√	×	√	$T_B + 17T_H + 6T_P$	$12T_H + 3T_P$	$8T_H + 4T_P$
Li et al.	2021	[32]	√	√	√	×	×	√	√	×	√	×	×	×	√	√	√	√	×	×	√	×	$10T_H$	$9T_H$	$7T_H$
Meshram et al.	2021	[29]	√	√	√	×	×	√	√	√	√	√	×	×	×	√	√	×	√	×	√	√	$12T_H + T_B$	$11T_H$	$11T_H$
Chaudhry et al.	2021	[30]	√	√	√	×	×	×	√	×	√	√	√	×	×	√	√	√	×	×	√	√	$17T_H + T_B$	$13T_H$	$8T_H$
Wu et al.	2021	[31]	√	√	√	×	×	×	√	√	√	√	×	×	×	√	√	√	√	×	√	√	$11T_H + T_B$	$14T_H$	$6T_H$
Sahoo et al.	2021	[26]	√	√	√	×	×	√	√	×	√	√	×	×	×	√	√	√	×	√	√	√	$T_B + 8T_H + 2T_S + 3T_P$	$4T_H + 2T_S + 2T_P$	–
Guo et al.	2021	[44]	×	√	×	×	×	√	×	√	√	√	×	×	√	√	√	√	×	×	×	√	$21T_H$	$18T_H$	$8T_H$
Li et al.	2020	[3]	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	×	√	√	√	$3T_P + 10T_H + T_B$	$T_P + 8T_H$	$2T_P + 4T_H$
Mo et al.	2020	[61]	√	√	√	√	×	√	√	×	√	√	√	×	√	√	√	√	×	√	×	√	$11T_H + T_E$	$13T_H + T_E$	$7T_H$
Fotouhi et al.	2020	[41]	×	√	×	×	×	√	√	√	×	√	×	√	√	×	√	√	√	×	√	×	$10T_H$	$17T_H$	$6T_H$
Hajian et al.	2020	[51]	√	√	√	×	×	√	√	×	√	√	√	×	√	√	√	√	×	×	√	√	$13T_H$	$7T_H$	$9T_H$
Wazid et al.	2020	[52]	√	√	√	×	×	√	√	×	√	√	√	×	×	√	√	√	×	×	√	√	$16T_H + T_B$	$8T_H$	$8T_H$
Lee et al.	2020	[53]	√	√	√	×	×	√	√	×	√	√	√	×	×	√	√	√	×	×	√	√	$9T_H$	$7T_H$	$7T_H + 2T_S$
Vinoth et al.	2020	[62]	√	√	√	×	×	√	√	√	×	√	√	×	√	√	√	√	√	√	×	√	$9T_H + T_B + T_S$	$6T_H + 3T_S + 2T_E$	$4T_H + 2T_S$
Srinivas et al.	2020	[42]	√	√	√	×	×	√	√	×	√	√	×	√	×	√	×	√	√	√	×	√	$2T_C + 14T_H + T_B$	$10T_H$	$2T_C + 6T_H$
Lin	2019	[63]	√	√	√	×	×	√	√	×	√	√	√	×	√	√	√	√	×	√	×	×	$3T_H + 2T_E$	$3T_H + T_E$	–
Lu et al.	2019	[27]	√	√	√	×	×	√	√	×	√	√	×	×	√	√	×	√	×	√	×	×	$T_B + 6T_H + T_S + 3T_P$	$T_P + 4T_H + T_S$	$2T_P + 2T_H + 2T_S$
Zhang et al.	2019	[40]	×	√	√	×	×	√	√	√	×	√	×	√	√	√	√	×	√	√	√	√	$6T_P + 2T_M + T_B$	$T_S + 8T_P + 2T_M$	–
Wei et al.	2018	[64]	√	√	√	×	×	√	√	×	×	√	×	×	√	√	√	√	√	√	×	√	$5T_H + 2T_S$	$4T_H + 2T_S$	$2T_S + 3T_H$
Li et al.	2018	[65]	√	√	√	×	×	√	√	√	√	√	√	√	√	√	√	√	×	√	×	√	$3T_P + 7T_H + T_B$	$T_P + 6T_H$	$2T_P + 4T_H$
Wazid et al.	2018	[66]	√	√	√	×	×	√	√	×	√	√	×	√	×	√	√	√	×	√	×	√	$T_B + 13T_H + 2T_S$	$5T_H + 4T_S$	$4T_H + 2T_S$
Wu et al.†	2018	[67]	√	×	√	√	×	×	√	√	×	√	×	×	–	–	–	–	–	–	–	–	$11T_H$	$15T_H$	$5T_H$
Amin et al. †	2018	[68]	√	√	×	×	×	×	√	×	√	√	√	×	–	–	–	–	–	–	–	–	$12T_H$	$15T_H$	$6T_H$
Amin et al.†	2018	[69]	√	√	√	×	×	√	√	×	√	√	√	×	–	–	–	–	–	–	–	–	$T_B + 14T_H$	$16T_H$	$4T_H$
Li et al.†	2018	[70]	√	√	√	×	×	√	√	√	√	√	×	×	–	–	–	–	–	–	–	–	$2T_P + T_B + 8T_H$	$T_P + 9T_H$	$4T_H$
Ali et al.	2018	[71]	√	√	√	×	×	√	√	×	√	√	×	×	√	√	√	√	×	×	×	×	$T_B + 7T_H + 2T_S$	$11T_H + 5T_S$	$4T_H + T_S$
Challa et al.	2018	[72]	√	√	√	×	×	√	√	×	√	√	×	×	√	√	√	×	×	√	×	×	$T_P + T_B + 9T_H$	$T_P + 4T_H$	$5T_H$
Xiong et al.†	2017	[73]	√	√	√	×	×	√	√	×	√	√	√	×	–	–	–	–	–	–	–	–	$9T_H + 2T_S$	$12T_H + 2T_S$	$5T_H$
Wu et al.	2017	[74]	√	√	√	×	×	√	√	√	×	√	√	√	√	√	√	√	×	√	×	√	$2T_P + T_S + 12T_H$	$2T_S + 11T_H$	$2T_P + T_S + 4T_H$
Wang et al.†	2017	[75]	√	√	×	√	√	√	√	×	√	√	√	√	–	–	–	–	–	–	–	–	$3T_P + T_B + 10T_H$	$T_P + 11T_H$	$2T_P + 4T_H$
Jung et al.	2017	[76]	√	√	×	×	×	√	√	×	√	√	×	×	×	√	√	×	×	×	×	√	$T_B + 7T_H$	$9T_H$	$4T_H$
Tai et al.†	2017	[77]	√	√	√	×	×	√	√	×	√	√	√	×	–	–	–	–	–	–	–	–	$8T_H$	$10T_H$	$6T_H$
Dhillon et al.†	2017	[78]	√	√	√	×	×	√	√	×	√	√	×	×	–	–	–	–	–	–	–	–	$T_B + 9T_H$	$6T_H$	$7T_H$
Jiang et al.†	2017	[79]	√	√	√	√	√	√	√	×	√	√	√	×	–	–	–	–	–	–	–	–	$T_E + T_B + 8T_H$	$T_E + 12T_H$	$5T_H$
Park et al.†	2016	[80]	√	√	×	×	×	√	√	×	√	√	×	√	–	–	–	–	–	–	–	–	$2T_P + T_B + 10T_H$	$11T_H$	$2T_P + 4T_H$
Gope et al.†	2016	[81]	√	√	√	×	×	√	√	×	√	√	√	×	–	–	–	–	–	–	–	–	$12T_H$	$9T_H$	$4T_H$
Reddy et al.†	2016	[82]	√	√	√	×	×	√	√	√	√	√	√	√	–	–	–	–	–	–	–	–	$3T_P + 10T_H$	$2T_P + 7T_H$	$2T_P + 5T_H$
Chang et al.	2016	[83]	√	√	√	×	×	√	√	×	×	√	×	√	√	√	√	√	×	√	×	√	$2T_P + 5T_H$	$8T_H$	$2T_P + 5T_H$
Das et al.	2016	[84]	√	√	√	×	×	√	√	×	√	√	×	√	×	√	√	√	×	√	×	×	$2T_P + T_B + 12T_H$	$2T_P + 9T_H$	$10T_H$
Lu et al.	2016	[85]	√	√	√	×	×	√	√	×	√	√	√	×	√	√	√	√	×	×	×	×	$7T_H + 2T_S$	$9T_H + 4T_S$	$3T_H + 2T_S$
Kumari et al.†	2016	[86]	√	√	√	×	×	√	√	√	√	×	×	√	–	–	–	–	–	–	–	–	$2T_P + 4T_H + 2T_S$	$6T_H + T_S$	$2T_P + 3T_H$
Jiang et al.†	2016	[87]	√	√	√	√	×	×	√	×	×	√	√	√	–	–	–	–	–	–	–	–	$2T_P + 8T_H$	$T_P + 9T_H$	$6T_H$
Amin et al.†	2016	[88]	√	√	√	×	×	√	√	×	√	√	×	×	–	–	–	–	–	–	–	–	$12T_H$	$15T_H$	$5T_H$
Farash et al.†	2016	[89]	√	√	√	×	×	×	√	×	√	√	×	×	–	–	–	–	–	–	–	–	$11T_H$	$14T_H$	$7T_H$
He et al.†	2015	[90]	√	×	√	×	×	×	√	×	×	√	√	×	–	–	–	–	–	–	–	–	$8T_H$	$9T_H$	$6T_H$
Das et al.	2015	[91]	√	√	√	×	×	√	√	×	√	√	√	×	√	√	√	√	×	×	×	×	$8T_H + T_S$	$2T_H + T_S$	$2T_H + T_S$
Chang et al.	2015	[92]	√	√	√	×	×	×	√	×	√	√	√	×	√	√	√	√	×	×	√	√	$11T_H$	$10T_H$	$4T_H$
Jiang et al.†	2015	[93]	√	√	√	×	×	×	√	×	×	√	√	×	–	–	–	–	–	–	–	–	$7T_H$	$10T_H$	$5T_H$
Choi et al.†	2014	[94]	√	√	√	×	×	√	√	×	√	√	×	√	–	–	–	–	–	–	–	–	$3T_P + 9T_H$	$2T_P + 6T_H$	$T_P + 5T_H$
Kim et al.†	2014	[95]	√	√	×	×	×	×	√	×	√	√	×	×	–	–	–	–	–	–	–	–	$8T_H$	$8T_H$	$2T_H$
Turkanovic et al.†	2014	[96]	√	√	√	×	×	√	√	×	√	√	×	×	–	–	–	–	–	–	–	–	$7T_H$	$7T_H$	$5T_H$
Turkanovic et al.†	2013	[97]	√	√	√	×	×	√	√	×	√	√	×	×	–	–	–	–	–	–	–	–	$4T_H + T_S$	$2T_H + 3T_S$	$T_H + T_S$
Xue et al.†	2013	[98]	×	√	×	×	×	×	√	×	√	√	×	×	–	–	–	–	–	–	–	–	$9T_H$	$14T_H$	$6T_H$
Shi et al.†	2013	[99]	√	√	√	×	×	√	√	×	√	√	×	√	–	–	–	–	–	–	–	–	$3T_P + 6T_H$	$T_P + 5T_H$	$2T_P + 4T_H$
He et al.†	2012	[100]	√	√	√	×	×	√	×	×	×	√	×	×	–	–	–	–	–	–	–	–	$3T_H + T_S$	$4T_H + 2T_S$	$2T_H + T_S$
Das et al.†	2012	[101]	√	√	√	×	×	√	√	×	√	×	×	×	–	–	–	–	–	–	–	–	$5T_H + T_S$	$3T_H + T_S$	$2T_H + T_S$
Kumari et al.†	2012	[102]	√	√	×	×	×	×	√	×	√	√	√	×	–	–	–	–	–	–	–	–	$3T_H + 2T_S$	$3T_S + T_H$	$T_H + 2T_S$
Vaidya et al.†	2012	[103]	√	√	×	×	×	×	√	×	√	√	×	×	–	–	–	–	–	–	–	–	$8T_H$	$6T_H$	$3T_H$
Kumari et al.†	2011	[104]	√	√	√	×	×	×	√	×	√	√	×	×	–	–	–	–	–	–	–	–	$6T_H + 2T_S$	$3T_H + 3T_S$	$2T_H + 3T_S$
Fan et al.†	2011	[105]	√	√	×	×	×	×	√	×	√	×	×	×	–	–	–	–	–	–	–	–	$8T_H$	$6T_H$	$2T_H$
Yeh et al.†	2011	[106]	×	×	√	×	×	×	√	×	√	×	×	×	–	–	–	–	–	–	–	–	$2T_P + 4T_H$	$4T_P + 4T_H$	$2T_P + 3T_H$
Vaidya et al.†	2010	[107]	√	√	×	×	×	×	×	×	×	×	×	×	–	–	–	–	–	–	–	–	$6T_H$	$4T_H$	$2T_H$
He et al.†	2010	[108]	√	√	√	×	×	×	×	×	√	×	×	×	–	–	–	–	–	–	–	–	$5T_H$	$5T_H$	T_H
Das†	2009	[109]	√	×	×	×	×	×	×	×	×	×	×	×	–	–	–	–	–	–	–	–	$4T_H$	$4T_H$	T_H

*: "√" denotes the scheme can provide the corresponding attribute. "×" in the white area denotes that the scheme cannot provide the corresponding attribute; "×" in the gray area indicates that the scheme has the corresponding security proof failure. "-" means the attribute is not applied to the scheme.

⋆: T_E, T_P, T_C, $\mathsf{T_B}$, T_H, T_S, T_M denote the operation time for modular exponentiation, elliptic curve point multiplication, chebysev chaotic-map, fuzzy extractor, hash, symmetric encryption, and message authentication code respectively, Some lightweight operations like XOR and $\|$ are omitted.

†: The authentication scheme that do not perform the formal security proof.

goals, semantic security and password protection, and then prove them respectively. Proof from multiple perspectives may help designers to realize if there are potential vulnerabilities in the proposed protocol as early as possible.

Supplemented by heuristic security analysis: As we have shown in Section VII-C, Fotouhi et al.'s scheme cannot resist the de-synchronization attack. This complex attack aims to make the system unable to authenticate the user's identity, and it is essentially a denial of service attack. As we have mentioned earlier, the known attacks in multi-factor authentication are the utilization of protocol design vulnerabilities. Different attack targets will cause different attack consequences even for the same design vulnerability. For example, not using public key cryptography to design multi-factor authentication schemes may lead to offline password guessing or loss of user anonymity [32], [54]. As a result, capturing all cryptanalysis attacks in a formal adversary model is impossible, which is also a limitation of the provable security. Therefore, we suggest using heuristic analysis to assist formal proofs, focusing on the attack scenarios that the formal adversary model cannot capture to enhance protocol designers' confidence in the security of their cryptographic protocols.

XI. A Comparative Evaluation of Exiting Multi-factor Authentication Schemes

Based on our taxonomy of security proof failures in Sec. III, we naturally form new evaluation criteria by combining the 12 protocol criteria shown in Section II-C and eight types of security failures proposed in this paper. We then perform a large-scale assessment of 70 multi-factor user authentication schemes for WSNs environments and general environments under our new criteria in Table III. The selected protocols range from 2009 to 2022, which are typical schemes that have attracted much attention and most of them are equipped with formal security proof under the ROM model. It can be seen from Table III that, although Goldwasser and Micali [37] proposed the concept of provable security in 1984, this tool was not used in the early stages of multi-factor authentication. The formal security proof is not gradually adopted until 2015. Since 2018, formal proof has almost become a routine for security analysis of protocols. Based on the 12 evaluation criteria, our comparison results show that, overall, schemes with formal security proof perform better (i.e., meet more criteria) than those without, which is in line with our understanding of the development of things. However, from the view of the failures in security proofs, the relevant research in the past ten years has not presented a trend of improvement, which may be attributed to the fact that few studies reveal the reasons for the failures in formal security proofs. This also highlights the necessity of our work as a first step towards exploring the failures of provable security.

XII. Conclusion

In this paper, we have taken a substantial first step towards systematically exploring the security proof failures in multi-factor authentication schemes for mobile devices. We first investigate the root causes of provable security failures in vulnerable multi-factor authentication schemes under the ROM model and categorize them into eight different types in terms of five steps when conducting a formal proof. Second, we combine the existing protocol evaluation criteria and our taxonomy of failures in security proofs, and develop an enhanced evaluation set. Then, we elaborate on each type of these eight proof failures by examining three typical protocols and then suggest corresponding countermeasures. Third, we conduct a large-scale comparative measurement of 70 representative multi-factor authentication schemes. Considering the unsatisfactory situation of formal proofs, we believe that understanding failures in security proofs is necessary to design secure multi-factor authentication protocols for mobile devices.

References

[1] M. Weiser, "The computer for the 21st century," *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 3, no. 3, pp. 3–11, 1999.

[2] L. Zhou, "Mobile device-to-device video distribution: Theory and application," *ACM Trans. Multim. Comput. Commun. Appl.*, vol. 12, no. 3, pp. 38:1–38:23, 2016.

[3] X. Li, J. Peng, M. S. Obaidat *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, 2020.

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

[5] 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, 2018.

[6] J. Bonneau, C. Herley, P. C. Van Oorschot *et al.*, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *Proc. IEEE S&P 2012*, pp. 553–567.

[7] M. Shirvanian, S. Jarecki, N. Saxena *et al.*, "Two-factor authentication resilient to server compromise using mix-bandwidth devices," in *Proc. NDSS 2014*.

[8] N. Karapanos, C. Marforio, C. Soriente *et al.*, "Sound-proof: Usable two-factor authentication based on ambient sound," in *Proc. USENIX 2015*, pp. 483–498.

[9] A. Acar, S. Ali, K. Karabina *et al.*, "A lightweight privacy-aware continuous authentication protocol-paca," *ACM Trans. Priv. Secur.*, vol. 24, no. 4, pp. 24:1–24:28, 2021.

[10] V. Zimmermann, "From the quest to replace passwords towards supporting secure and usable password creation," Ph.D. dissertation, Technical University of Darmstadt, Germany, 2021.

[11] S. M. Bellovin and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," in *Proc. IEEE S&P 1992*, pp. 72–84.

[12] A. Groce and J. Katz, "A new framework for efficient password-based authenticated key exchange," in *Proc. ACM CCS 2010*, pp. 516–525.

[13] S. Jarecki, H. Krawczyk, and J. Xu, "OPAQUE: an asymmetric PAKE protocol secure against pre-computation attacks," in *Proc. EUROCRYPT 2018*, pp. 456–486.

[14] *Yahoo's 2013 Email Hack Actually Compromised Three Billion Accounts*, Oct. 2017, https://www.wired.com/story/yahoo-breach-three-billion-accounts/.

[15] E. Mikalauskas, *RockYou2021: largest password compilation of all time leaked online with 8.4 billion entries*, Jun. 2021, https://cybernews.com/security/rockyou2021-alltime-largest-password-compilation-leaked/.

[16] *Report: FlexBooker Suffers Another Data Breach Exposing Millions of Bookings*, January 2022, https://www.vpnmentor.com/blog/report-flexbooker-leak/.

[17] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, 2002.

[18] K. Nohl, D. Evans, Starbug, and H. Plötz, "Reverse-engineering a cryptographic RFID tag," in *Proc. USENIX SEC 2008*, pp. 185–194.

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

[20] U. Uludag, S. Pankanti, S. Prabhakar *et al.*, "Biometric cryptosystems: Issues and challenges," in *Proc. IEEE 2004*, pp. 948–960.

[21] H. Feng and D. Clarke, "Security analysis of a multi-factor authenticated key exchange protocol," in *Proc. ACNS 2012*, pp. 1–11.

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

[23] X. Huang, X. Chen *et al.*, "Further observations on smart-card-based password-authenticated key agreement in distributed systems," *IEEE Trans. Parallel Distributed Syst.*, vol. 25, no. 7, pp. 1767–1775, 2014.

[24] S. Kumari, M. K. Khan, and M. Atiquzzaman, "User authentication schemes for wireless sensor networks: A review," *Ad Hoc Networks*, vol. 27, pp. 159–194, 2015.

[25] T. Hwang, Y. Chen *et al.*, "Non-interactive password authentications without password tables," in *Proc. IEEE TENCON 1990*, pp. 429–431.

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

[27] Y. Lu, G. Xu, L. Li *et al.*, "Anonymous three-factor authenticated key agreement for wireless sensor networks," *Wirel. Networks*, vol. 25, no. 4, pp. 1461–1475, 2019.

[28] C. Wang, D. Wang, Y. Tu *et al.*, "Understanding node capture attacks in user authentication schemes for wireless sensor networks," *IEEE Trans. Dependable Secur. Comput.*, vol. 19, no. 1, pp. 507–523, 2022.

[29] C. Meshram, M. S. Obaidat *et al.*, "An efficient, robust, and lightweight subtree-based three-factor authentication procedure for large-scale dwsn in random oracle," *IEEE Syst. J.*, vol. 15, no. 4, pp. 4927–4938, 2021.

[30] S. A. Chaudhry, A. Irshad *et al.*, "Rotating behind privacy: An improved lightweight authentication scheme for cloud-based iot environment," *ACM Trans. Internet Techn.*, vol. 21, no. 3, pp. 78:1–78:19, 2021.

[31] T. Wu, L. Yang *et al.*, "A provably secure three-factor authentication protocol for wireless sensor networks," *Wirel. Commun. Mob. Comput.*, vol. 2021, pp. 5 537 018:1–5 537 018:15, 2021.

[32] J. Li, Z. Su, D. Guo *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. 13 183–13 195, 2021.

[33] J. Srinivas, A. K. Das *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, 2021.

[34] R. Madhusudhan and R. C. Mittal, "Dynamic id-based remote user password authentication schemes using smart cards: A review," *J. Netw. Comput. Appl.*, vol. 35, no. 4, pp. 1235–1248, 2012.

[35] D. Wang, D. He, P. Wang *et al.*, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Trans. Dependable Secur. Comput.*, vol. 12, no. 4, pp. 428–442, 2015.

[36] D. Wang, W. Li *et al.*, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Trans. Ind. Informatics*, vol. 14, no. 8, pp. 4081–4092, 2018.

[37] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270–299, 1984.

[38] M. Bellare, D. Pointcheval *et al.*, "Authenticated key exchange secure against dictionary attacks," in *Proc. EUROCRYPT 2000*, pp. 139–155.

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

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

[41] M. Fotouhi, M. Bayat *et al.*, "A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care iot," *Comput. Networks*, vol. 177, p. 107333, 2020.

[42] J. Srinivas, A. K. Das, M. Wazid *et al.*, "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, 2020.

[43] D. He and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment," *IEEE Syst. J.*, vol. 9, no. 3, pp. 816–823, 2015.

[44] Y. Guo, Z. Zhang, and Y. Guo, "Anonymous authenticated key agreement and group proof protocol for wearable computing," *IEEE Trans. Mob. Comput.*, 2021, doi: 10.1109/TMC.2020.3048703.

[45] J. Bonneau, "The science of guessing: Analyzing an anonymi- zed corpus of 70 million passwords," in *Proc. IEEE S&P 2012*.

[46] D. Wang, Z. Zhang, P. Wang *et al.*, "Targeted online password guessing: An underestimated threat," in *Proc. ACM CCS 2016*.

[47] D. Dolev and A. C. Yao, "On the security of public key proto- cols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–207, 1983.

[48] S. Palka, H. Wechsler, and B. A. Hamilton, "Fingerprint readers: vulnerabilities to front-and back-end attacks," in *Proc. IEEE BTAS 2007*.

[49] M. Abdalla, O. Chevassut, P. Fouque, and D. Pointcheval, "A simple threshold authenticated key exchange from short secrets," in *Proc. ASIACRYPT 2005*, pp. 566–584.

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

[51] R. Hajian, S. ZakeriKia *et al.*, "SHAPARAK: scalable healthcare authentication protocol with attack-resilience and anonymous key-agreement," *Comput. Networks*, vol. 183, p. 107567, 2020.

[52] M. Wazid, A. K. Das, V. B. K. *et al.*, "Lam-ciot: Lightweight authentication mechanism in cloud-based iot environment," *J. Netw. Comput. Appl.*, vol. 150, 2020.

[53] H. Lee, D. Kang, J. Ryu *et al.*, "A three-factor anonymous user authentication scheme for internet of things environments," *J. Inf. Secur. Appl.*, vol. 52, p. 102494, 2020.

[54] 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.

[55] Y. Guo, Z. Zhang, and Y. Guo, "Secfhome: Secure remote authentication in fog-enabled smart home environment," *Comput. Networks*, vol. 207, p. 108818, 2022.

[56] Y. Xia, R. Qi, S. Ji *et al.*, "Puf-assisted lightweight group authentication and key agreement protocol in smart home," *Wirel. Commun. Mob. Comput.*, 2022, doi: 10.1155/2022/8865158.

[57] A. K. Sutrala, M. S. Obaidat *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, 2022.

[58] F. Raque, M. Obaidat, K. Mahmood *et al.*, "An efficient and provably secure certificateless protocol for industrial internet of things," *IEEE Trans. Ind. Informatics*, 2022.

[59] L. Zhang, Y. Zhu, W. Ren *et al.*, "Privacy-preserving fast authentication and key agreement for e-health systems in iot, based on three-factor authentication," *IEEE Trans. Serv. Comput.*, 2022, doi: 10.1109/TSC.2022.3149940.

[60] 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.*, 2022, doi: 10.1109/JSYST.2022.3152561.

[61] J. Mo, W. Shen, and W. Pan, "An improved anonymous authentication protocol for wearable health monitoring systems," *Wirel. Commun. Mob. Comput.*, vol. 2020, pp. 5 686 498:1–5 686 498:13, 2020.

[62] R. Vinoth, L. J. Deborah, P. Vijayakumar *et al.*, "Secure multi-factor authenticated key agreement scheme for industrial IoT," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3801–3811, 2020.

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

[64] F. Wei, R. Zhang, and C. Ma, "A provably secure anonymous two-factor authenticated key exchange protocol for cloud computing," *Fundam. Informaticae*, vol. 157, no. 1-2, pp. 201–220, 2018.

[65] X. Li, J. Niu *et al.*, "A robust ecc based provable secure authentication protocol with privacy preserving for industrial internet of things," *IEEE Trans. Ind. Inform.*, vol. 14, no. 8, pp. 3599–3609, 2018.

[66] M. Wazid, A. K. Das *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, 2018.

[67] F. Wu, X. Li *et al.*, "A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks," *Future Gener. Comput. Syst.*, vol. 82, pp. 727–737, 2018.

[68] R. Amin, S. H. Islam *et al.*, "A robust and anonymous patient monitoring system using wireless medical sensor networks," *Future Gener. Comput. Sys.*, vol. 80, pp. 483–495, 2018.

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

[70] X. Li, J. Niu, S. Kumari *et al.*, "A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments," *J. Netw. Comput. Appl.*, vol. 103, pp. 194–204, 2018.

[71] R. Ali, A. K. Pal, S. Kumari *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.

[72] S. Challa, A. K. Das, V. Odelu *et al.*, "An efficient ecc-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks," *Comput. Electr. Eng.*, vol. 69, pp. 534–554, 2018.

[73] L. Xiong, D. Peng, T. Peng *et al.*, "A lightweight anonymous authentication protocol with perfect forward secrecy for wireless sensor networks," *Sensors*, vol. 17, no. 11, p. 2681, 2017.
[74] F. Wu, L. Xu, S. Kumari *et al.*, "A new and secure authentication scheme for wireless sensor networks with formal proof," *Peer Peer Netw. Appli.*, vol. 10, no. 1, pp. 16–30, 2017.
[75] C. Wang, G. Xu, and J. Sun, "An enhanced three-factor user authentication scheme using elliptic curve cryptosystem for wireless sensor networks," *Sensors*, vol. 17, no. 12, pp. 2946:1–20, 2017.
[76] J. Jung, J. Moon, D. Lee *et al.*, "Efficient and security enhanced anonymous authentication with key agreement scheme in wireless sensor networks," *Sensors*, vol. 17, no. 3, p. 644, 2017.
[77] W.-L. Tai, Y.-F. Chang, and W.-H. Li, "An iot notion-based authentication and key agreement scheme ensuring user anonymity for heterogeneous ad hoc wireless sensor networks," *J. Inform. Secur. Appl.*, vol. 34, pp. 133–141, 2017.
[78] P. K. Dhillon and S. Kalra, "Secure multi-factor remote user authentication scheme for internet of things environments," *Int. J. Commun. Syst.*, vol. 30, no. 16, p. e3323, 2017.
[79] Q. Jiang, S. Zeadally, J. Ma *et al.*, "Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376–3392, 2017.
[80] Y. Park and Y. Park, "Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks," *Sensors*, vol. 16, no. 12, p. 2123, 2016.
[81] P. Gope, T. Hwang *et al.*, "A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7124–7132, 2016.
[82] A. G. Reddy, A. K. Das, E.-J. Yoon *et al.*, "A secure anonymous authentication protocol for mobile services on elliptic curve cryptography," *IEEE Access*, vol. 4, pp. 4394–4407, 2016.
[83] C. C. Chang and H. D. Le, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Trans. Wirel. Commun.*, vol. 15, no. 1, pp. 357–366, 2016.
[84] A. K. Das, S. Kumari *et al.*, "Provably secure user authentication and key agreement scheme for wireless sensor networks," *Secur. Commun. Netw.*, vol. 9, no. 16, pp. 3670–3687, 2016.
[85] Y. Lu, L. Li, H. Peng *et al.*, "An energy efficient mutual authentication and key agreement scheme preserving anonymity for wireless sensor networks," *Sensors*, vol. 16, no. 6, p. 837, 2016.
[86] S. Kumari, X. Li, F. Wu *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.
[87] Q. Jiang, J. Ma, F. Wei *et al.*, "An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 76, pp. 37–48, 2016.
[88] R. Amin, S. H. Islam, G. P. Biswas *et al.*, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Comput. Networks*, vol. 101, pp. 42–62, 2016.
[89] M. S. Farash, Turkanovi, Muhamed *et al.*, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment," *Ad Hoc Networks*, vol. 36, pp. 152–176, 2016.
[90] D. He, N. Kumar *et al.*, "A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks," *Inf. Sci.*, vol. 321, pp. 263–277, 2015.
[91] A. K. Das, "A secure and efficient user anonymity-preserving three-factor authentication protocol for large-scale distributed wireless sensor networks," *Wireless Pers. Commun.*, vol. 82, no. 3, pp. 1377–1404, 2015.
[92] I. Chang, T. Lee *et al.*, "Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks," *Sensors*, vol. 15, no. 12, pp. 29 841–29 854, 2015.
[93] Q. Jiang, J. Ma *et al.*, "An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks," *Peer Peer Netw. Appl.*, vol. 8, no. 6, pp. 1070–1081, 2015.
[94] Y. Choi, D. Lee *et al.*, "Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 14, no. 6, pp. 10 081–10 106, 2014.
[95] J. Kim, D. Lee, W. Jeon *et al.*, "Security analysis and improvements of two-factor mutual authentication with key agreement in wireless sensor networks," *Sensors*, vol. 14, no. 4, pp. 6443–6462, 2014.
[96] M. Turkanovic, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion," *Ad Hoc Networks*, vol. 20, pp. 96–112, 2014.
[97] M. Turkanovic and M. Holbl, "An improved dynamic password-based user authentication scheme for hierarchical wireless sensor networks," *Elektronika Ir Elektrotechnika*, vol. 19, no. 6, pp. 109–116, 2013.
[98] K. Xue, C. Ma, P. Hong *et al.*, "A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 316–323, 2013.
[99] W. Shi and P. Gong, "A new user authentication protocol for wireless sensor networks using elliptic curves cryptography," *Int. J. Distributed Sens. Networks*, vol. 9, no. 4, pp. 51–59, 2013.
[100] D. He, Y. Zhang, and J. Chen, "Robust biometric-based user authentication scheme for wireless sensor networks," *Ad Hoc Sens. Wirel. Networks*, vol. 25, no. 3-4, pp. 309–321, 2015.
[101] A. Das, S. Sharma, P.and Chatterjee *et al.*, "A dynamic password-based user authentication scheme for hierarchical wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 35, no. 5, pp. 1646–1656, 2012.
[102] P. Kumar, S.-G. Lee, and H.-J. Lee, "E-sap: Efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks," *Sensors*, vol. 12, no. 2, pp. 1625–1647, 2012.
[103] B. Vaidya, D. Makrakis *et al.*, "Two factor mutual authentication with key agreement in wireless sensor networks," *Secur. Commun. Netw.*, vol. 9, no. 2, pp. 171–183, 2016.
[104] P. Kumar, A. J. Choudhury, M. Sain *et al.*, "RUASN: a robust user authentication framework for wireless sensor networks," *Sensors*, vol. 11, no. 5, pp. 5020–5046, 2011.
[105] R. Fan, D. He, X. Pan *et al.*, "An efficient and dos-resistant user authentication scheme for two-tiered wireless sensor networks," *J. Zhejinag Univ. Sci. C*, vol. 12, no. 7, pp. 550–560, 2011.
[106] H. Yeh, T. Chen, P. Liu *et al.*, "A secured authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 11, no. 5, pp. 4767–79, 2011.
[107] B. Vaidya, J. J. Rodrigues, and P. J. Hyuk, "User authentication schemes with pseudonymity for ubiquitous sensor network in ngn," *Int. J. Commun. Sys.*, vol. 23, no. 9-10, pp. 1201–1222, 2010.
[108] D. He, Y. Gao, S. Chan *et al.*, "An enhanced two-factor user authentication scheme in wireless sensor networks," *Ad Hoc Sens. Wirel. Networks*, vol. 10, no. 4, pp. 361–371, 2010.
[109] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Trans. Wirel. Commun.*, vol. 8, no. 3, pp. 1086–1090, 2009.
[110] T. Roche, V. Lomné, C. Mutschler *et al.*, "A side journey to titan," in *Proc. USENIX SEC 2021*, pp. 231–248.

Qingxuan Wang received the MS degree in information security from the China University of Geosciences, Wuhan, P. R. China, in Jun. 2020. He is currently working toward the PhD degree from the College of Cyber science, Nankai University, Tianjin, P. R. China. He has published papers at venues like IEEE TDSC, IEEE TSC and JSA. His research interests include applied cryptography and password-based authentication.

Ding Wang received his Ph.D. degree in Information Security at 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 at Nankai University. As the first author (or corresponding author), he has published more than 80 papers at venues like IEEE S&P, ACM CCS, NDSS, Usenix Security, IEEE TDSC and IEEE TIFS. His research has been reported by over 200 medias like Daily Mail, Forbes, IEEE Spectrum and Communications of the ACM, appeared in the Elsevier 2017 "Article Selection Celebrating Computer Science Research in China", and resulted in the revision of the authentication guideline NIST SP800-63-2. He has been involved in the community as a PC Chair/TPC member for over 60 international conferences such as NDSS 2023, ACM CCS 2022, PETS 2023/2022, ACSAC 2020-2022, ACM AsiaCCS 2022/2021, IFIP SEC 2018-2021, ICICS 2018-2022. He has received the "ACM China Outstanding Doctoral Dissertation Award", the Best Paper Award at INSCRYPT 2018, the Outstanding Youth Award of China Association for Cryptologic Research, the Young Scientist Nomination Award for Powerful Nation, and the First Prize of Natural Science Award of Ministry of Education. His research interests focus on passwords, authentication and provable security.