Cryptanalysis and Improvement of a Multi-factor Authenticated Key Exchange Protocol

Zhiqiang Ma[1,2] and Jun He[2]
(Corresponding author: Jun He)

College of Computer and Cyber Security & Fujian Normal University[1]
School of Computer Science and Engineering & Chongqing University of Technology[2]
69 Hongguang Avenue, Banan District, Chongqing 400054, China
Email: hejun@cqut.edu.cn

(Received Nov. 3, 2022; Revised and Accepted June 20, 2023; First Online Aug. 25, 2023)

Abstract

Authenticated key exchange (AKE) protocol is one of the most fundamental cryptographic primitives for secure communication systems. It allows two parties to securely establish a common session key over an insecure public network. Recently, Zhang *et al.* proposed a multi-factor authenticated key exchange (MFAKE) protocol for mobile communications. This paper presents the cryptanalysis of Zhang's MFAKE protocol. We find out Zhang's MFAKE protocol has a security flaw that renders it insecure against Man-in-the-Middle (MITM) attacks and outsider Key Compromise Impersonation (KCI) attacks. We present a simple case of MITM attacks and illustrate how an adversary impersonates the client to the server if just compromising the key of the server. And an improved MFAKE protocol is proposed to overcome the weakness of Zhang's MFAKE protocol with minimum changes. Then we give the formal security proof of our improved MFAKE protocol in the random oracle model. The security features and performance of our improved protocol are compared with related protocols. The results show that our improved MFAKE protocol is more secure and efficient.

Keywords: Authenticated Key Exchange; Key Compromise Impersonation Attack; Multi-factor

1 Introduction

With the rapid development of communication technologies, mobile devices have become popular in daily life. Advances in mobile telecommunication technology lay the foundation for accessing critical infrastructure. (e.g., industrial manufacturing, energy, healthcare, transportation). People interact with these systems to obtain personal services. However, adversaries could intercept, modify or replay messages, as well as impersonate a legal user to access the protected resource. Communication security has become one of the most crucial issues when accessing critical infrastructure for services [13, 25]. To prevent unauthorized access, authentication is a dominant form of access control in various services.

Authenticated key exchange (AKE) protocol allows two parties to share a common session key for secure communication over insecure public channels and verify the legitimacy of each other. Legitimate access to any information system requires authentication of the user accessing the protected information. Thus, password-based authenticated key exchange (PAKE) protocols [4, 5, 8, 15, 17] have received significant attention in user authentication systems. PAKE protocols assume a realistic scenario in which secret keys are not uniformly distributed over a large keyspace, but chosen from a small and low-entropy keyspace [19]. It is a realistic scenario in which users tend to choose short, easily-rememberable passwords since they may require to remember many passwords and change the password frequently [16, 20]. Thus, passwords are vulnerable to many brute-force and dictionary-based attack tools [12]. Although a solution [26] that the server stores a one-way transform of password is introduced to strengthen the security in client/server setting, Jarecki *et al.* [15] pointed out that this solution allows for precomputation attacks that lead to the instantaneous compromise of user passwords upon server compromise. Simple password-based authentication has proven to be more and more inadequate [18] since the existing solutions cannot sufficiently prevent password-cracking, data-stealing, and data-phishing practices. Various schemes [8, 31, 34] have been proposed in succession to reduce the affection of password-cracking and compromised password database.

With the growing number of innovative ways to authenticate users, there are three main approaches [24] for authentication: something you *know* (e.g., passwords), something you *have* (e.g., smartphones and smart cards), and something you *are* (e.g., biometric characteristics). In certain circumstances, however, the above factors may be insecure. When the honest user types in the correct password, the malicious user could peep the input. The smart card might be lost, stolen, or cloned. Once an adversary obtains the smart card, all the information stored could be lost. The biometric characteristics are irrevocable. Once copied by the adversary, this will cause permanent damage. Various multi-factor authenticated key exchange (MFAKE) schemes [7,22,27,36] were proposed successively by combining three factors in an authentication process to reduce the damage caused by compromising an authentication factor.

1.1 Motivations

User authentication is becoming more widely used to protect sensitive information from the illegitimate user. However, research over the past decade has shown that designing a secure authenticated key exchange scheme is very difficult. MFAKE schemes aim to achieve higher security by combining three factors within the same authentication process. Intuitively, an adversary would have to break all three factors to break the MFAKE scheme. However, an adversary could compromise less than three factors to break the scheme if the scheme is not well designed.

An AKE protocol is provable security if and only if the security proof is correct. Several results [10,21,33,35] show that even several of the proposed AKE protocols that have provided security proof cannot achieve their security aims since the security proof might be flawed. Constructing a multi-factor authentication protocol remains hard work. Analysis of defects in existing protocols can make us avoid these shortcomings when designing a new scheme.

1.2 Contributions

In this paper, we revisit Zhang's MFAKE protocol [36] and analyze its security. We hope our analysis would help avoid such mistakes when designing a new MFAKE protocol in the future. This paper is an extension work of Ma *et al.* [23]. The first contribution was presented in [23] at the CIMSS of ACNS workshop in 2022. The second contribution has some minor changes for entity authentication, which is different from [23]. The last two contributions are our new results. All contributions of this paper are listed as follows:

1) First, we show this protocol has a vital security flaw, which may lead the protocol insecure against Man-in-the-Middle (MITM) attacks and outsider Key Compromise Impersonation (KCI) attacks. The main problem of Zhang's MFAKE is the protocol message transcript is not bound to the session key. We give the details of a simple MITM attack and an outsider KCI attack in Section 5.

2) We then propose an improved MFAKE protocol to fix the problem of Zhang's MFAKE protocol with minimum changes. A hash algorithm takes protocol messages as inputs and outputs the session key. And one party only computes the session key after it authenticates another party.

3) In addition to the key indistinguishability security experiment, we also define an entity authentication security experiment. We provide the formal security proof of entity authentication and key indistinguishability in the random oracle model.

4) Finally, we evaluate security features and performance of our improved MFAKE protocol.

1.3 Organization of the Rest Article

The rest of the paper is organized as follows. In Section 2, we review the related works. In Section 3, we introduce the basic definitions for Zhang's MFAKE protocol. In Section 4, we give the security model. In Section 5, we review Zhang's MFAKE protocol, analyze the drawback of Zhang's MFAKE protocol, propose an improved MFAKE protocol and provide the formal security proof in the random oracle model. In Section 6, we show security features and performance of some related protocols and our improved MFAKE protocol. We conclude the paper in Section 7.

2 Related Work

Bellovin and Merritt [4] proposed the first password-based authenticated key exchange protocol, Encrypted Key Exchange (EKE), which allows the client and server to share the plaintext password and exchange key material to derive a common session key. Then the augmented EKE protocol proposed by Bellovin and Merritt [5], replaced the requirement that the server stores the plaintext password with a one-way transformed value of the password. Augment EKE protocol prevents the adversary from impersonating the honest user. They presented two ways to accomplish this goal, digital signatures and a family of commutative one-way functions. However, the EKE and augment EKE are not given formal security analysis since the lack of a proper security model. The first formal security model of AKE protocols between two parties was introduced by Bellare and Rogaway [3]. Bellare *et al.* [2] proposed the security model of PAKE protocols by extending the definition of Bellare and Rogaway [3]. And this PAKE security model has been followed extensively in papers [1,23,27].

The protocols referred to above build on the single authentication factor. Recently, MFAKE, a valuable and challenging goal, has wildly caught researchers' attention [7,11,22,27]. Many papers claim security by combining all three factors in a protocol. Pointcheval and Zimmer [27] defined a new security model for MFAKE protocols and proposed a multi-factor AKE protocol that was proved to be secure in their security model. They claim their MFAKE protocol remains semantically secure if there are at most two corrupt queries. Namely, an adversary must break all three factors to win the game. Liu *et al.* [22] proposed a three-party MFAKE protocol by extending Pointcheval's protocol [27]. They provided the formal security proof of their three-party MFAKE protocol in the random oracle model. However, Hao and Clarke [10] found out Pointcheval's protocol and Liu's protocol are insecure. If an adversary has compromised the client's password, it could impersonate the server to compromise the other two factors, thus breaking the entire system. Fleischhacker *et al.* [7] introduced and modeled a general framework for (α, β, γ)-MFAKE by extending the three-factor AKE model from [27]. And they defined a generalized notion of *tag-based* multi-factor authentication, extending the preliminary concepts from [14] that considered the use of tags (auxiliary strings) in public key-based challenge-response scenarios. In this way, they avoided the problems identified in [10] for the protocol in [27]. Wang *et al.* [29] introduced a multi-factor authentication protocol using elliptic curve cryptography. But Wu *et al.* [32] demonstrated that Wang *et al.* [29] protocol was insecure against impersonation attacks. Therefore, they proposed an improved authentication scheme and fixed the problems in [29]. Wu *et al.* [30] proposed a lightweight scheme for wireless sensor networks with multi-factor authentication. Hossein *et al.* [6] proposed a hash-chain-based provably secure MFAKE scheme and analyzed the security of their scheme in the Real-or-Random (ROR) model [1].

Most recently, Zhang *et al.* [36] proposed a multi-factor authenticated key exchange (MFAKE) scheme based on the security model from [27]. It claims to reduce the security of protocol to the Decisional Diffie-Hellman (DDH) hard problem. In this work, however, we found two weaknesses that led to Zhang's MFAKE insecurity. One problem is that an adversary could easily modify the exchanged message to lead two non-partnered sessions to compute the same session key. Another is that once an adversary compromises the server, it could impersonate the client to the server.

3 Preliminaries

Let $\lambda \in \mathbb{N}$ be the security parameter and 1^λ be a string that consists of λ bits. $\emptyset$ denotes an empty string. $\|$ is the string concatenation operation. $\bigoplus$ is the XOR operation. For $n \in \mathbb{N}$, $[n] := \{1, 2, \ldots, n\}$ denotes the set of integers between 1 and n. If X is a set, $x \xleftarrow{\$} X$ denotes the operation of sampling a uniform random element x from X. If A is a probabilistic algorithm, $a \xleftarrow{\$} A$ means that a is the output of running A with fresh random coins. The hash function $\mathsf{h}(\cdot) : \{0,1\}^* \rightarrow \{0,1\}^\lambda$ is modeled as a random oracle.

3.1 Metric Space

A *metric space* is a set $\mathcal{M}$ with a distance function $\mathsf{Dist} : \mathcal{M} \times \mathcal{M} \rightarrow [0, \infty)$. Commonly, *Hamming distance* is used to measure the distance from one value to another value. $\mathsf{Dist}(w, w')$ is the number of positions in which the strings $w \in \mathcal{M}$ and $w' \in \mathcal{M}$ differ. For an element $w \in \mathcal{M}$, let $\mathsf{Dist}(w) := \mathsf{Dist}(w, 0)$.

3.2 Min-Entropy and Statistical Distance

Definition 1 (Min-Entropy)**.** *The min-entropy of X is $H_\infty(X) = -\log_2(\max_x \Pr[X = x])$.*

Definition 2 (Statistical Distance)**.** *The statistical distance between two random variables A and B with the same domain $\mathcal{M}$ is*

$$\mathbf{SD}(A, B) = \frac{1}{2} \sum_{w \in \mathcal{M}} |\Pr[A = w] - \Pr[B = w]|.$$

If $\mathbf{SD}(A, B) \leq \epsilon$, A and B are called ϵ-statistically indistinguishable.

3.3 Public Key Encryption Scheme

Generally, we consider a public key encryption scheme PKE that consists of three probabilistic polynomial time (PPT) algorithms $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$. The PKE scheme is associated with public keyspace $\mathcal{PK}_{\mathsf{PKE}}$, private keyspace $\mathcal{SK}_{\mathsf{PKE}}$, message space $\mathcal{M}_{\mathsf{PKE}}$ and ciphertext space $\mathcal{C}_{\mathsf{PKE}}$. The algorithms of PKE are defined as follows:

- $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{PKE.KeyGen}(1^\lambda)$: This algorithm takes as input the security parameter 1^λ and outputs a pair of public/private keys $(\mathsf{pk}, \mathsf{sk})$, where the public key $\mathsf{pk} \in \mathcal{PK}_{\mathsf{PKE}}$ and the private key $\mathsf{sk} \in \mathcal{SK}_{\mathsf{PKE}}$.

- $c \xleftarrow{\$} \mathsf{PKE.Enc}(\mathsf{pk}, m)$: This is the encryption algorithm that generates a ciphertext $c \in \mathcal{C}_{\mathsf{PKE}}$ for a message $m \in \mathcal{M}_{\mathsf{PKE}}$ with the public key pk.

- $m \xleftarrow{\$} \mathsf{PKE.Dec}(\mathsf{sk}, c)$: This is the decryption algorithm which takes as input a private key sk, a ciphertext c, and outputs a message m. The correctness requirement is for all pairs $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{PKE.KeyGen}(1^\lambda)$, we have $m \equiv \mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{PKE.Enc}(\mathsf{pk}, m))$.

Definition 3 (Public Key Encryption Scheme)**.** *We say that a public key encryption scheme* $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ *is $(q, t, \epsilon_{\mathsf{PKE}})$-secure (indistinguishable) against adaptive chosen-ciphertext attacks, if* $|\Pr[\mathsf{EXP}^{ind\text{-}cca}_{\mathsf{PKE},\mathcal{A}}(\lambda) = 1] - 1/2| \leq \epsilon_{\mathsf{PKE}}$ *holds for all adversaries $\mathcal{A}$ running in time at most t in* *the following experiment:*

```
$\mathsf{EXP}^{ind\text{-}cca}_{\mathsf{PKE},\mathcal{A}}(\lambda)$ :
    $(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{PKE.KeyGen}(1^\lambda)$;
    $(m_0, m_1) \xleftarrow{\$} \mathcal{A}(pk)$;
    $b \xleftarrow{\$} \{0,1\}$;
    $c^* \xleftarrow{\$} \mathsf{PKE.Enc}(\mathsf{pk}, m_b)$;
    $b' \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathsf{PKE.Dec}}(\mathsf{sk},\cdot)}(\mathsf{pk}, c^*)$;
    if $b = b'$ then return 1,
    otherwise return 0 ;
```

```
$\mathcal{O}_{\mathsf{PKE.Dec}}(\mathsf{sk}, c)$ :
    if $c = c^*$, return a failure $\perp$;
    otherwise $m \xleftarrow{\$} \mathsf{PKE.Dec}(\mathsf{sk}, c)$
    return $m$;
```

where $\epsilon_{\mathsf{PKE}} = \epsilon_{\mathsf{PKE}}(\lambda)$ is a negligible function in the security parameter λ and the number of queries q is bound by time t.

3.4 Message Authentication Code Scheme

We consider a message authentication code scheme MAC that consists of three probabilistic polynomial time (PPT) algorithms $\mathsf{MAC} = (\mathsf{MAC.KeyGen}, \mathsf{MAC.Tag}, \mathsf{MAC.Vfy})$. The MAC scheme is associated with tag space $\mathcal{T}_{\mathsf{MAC}}$, message space $\mathcal{M}_{\mathsf{MAC}}$ and private keyspace $\mathcal{SK}_{\mathsf{MAC}}$. The algorithms of MAC are defined as follows:

- $\mathsf{sk}_{\mathsf{MAC}} \xleftarrow{\$} \mathsf{MAC.KeyGen}(1^\lambda)$: This is the key generation algorithm which takes as input 1^λ and outputs a secret key $\mathsf{sk}_{\mathsf{MAC}} \in \mathcal{SK}_{\mathsf{MAC}}$.

- $\tau \xleftarrow{\$} \mathsf{MAC.Tag}(\mathsf{sk}_{\mathsf{MAC}}, m)$: The generation algorithm is run by a party. It generates a tag $\tau \in \mathcal{T}_{\mathsf{MAC}}$ for a message $m \in \mathcal{M}_{\mathsf{MAC}}$ with the generation key $\mathsf{sk}_{\mathsf{MAC}}$.

- $\{0,1\} \xleftarrow{\$} \mathsf{MAC.Vfy}(\mathsf{sk}_{\mathsf{MAC}}, \tau, m)$: The verification algorithm is run by the verifier. It takes as input a private key $\mathsf{sk}_{\mathsf{MAC}}$, a tag τ, and a message m. Then it outputs 1 if τ is a valid tag for m under $\mathsf{sk}_{\mathsf{MAC}}$, and 0 otherwise.

Definition 4 (Message Authentication Code Scheme)**.** *We say that a message authentication code scheme $\mathsf{MAC} = (\mathsf{MAC.KeyGen}, \mathsf{MAC.Tag}, \mathsf{MAC.Vfy})$ is $(q, t, \epsilon_{\mathsf{MAC}})$-secure against strongly existential forgeries under chosen message attacks, if $\Pr[\mathsf{EXP}^{seuf-cma}_{\mathsf{MAC},\mathcal{A}}(\lambda) = 1] \le \epsilon_{\mathsf{MAC}}$ holds for all adversaries $\mathcal{A}$ running in time at most t in the following experiment:*

```
$\mathsf{EXP}^{seuf\text{-}cma}_{\mathsf{MAC},\mathcal{A}}(\lambda)$ :
    $\mathsf{sk}_{\mathsf{MAC}} \xleftarrow{\$} \mathsf{MAC.KeyGen}(1^\lambda)$;
    $(m^*, \tau^*) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathsf{MAC.Tag}}(\mathsf{sk}_{\mathsf{MAC}},\cdot)}$;
    return 1 if the following conditions are held:
        1) $\mathsf{MAC.Vfy}(\mathsf{sk}_{\mathsf{MAC}}, \tau^*, m^*) = 1$ and
        2) $\mathcal{A}$ didn't submit $m^*$ to $\mathsf{MAC.Tag}(\mathsf{sk}_{\mathsf{MAC}}, \cdot)$,
    and 0 otherwise;
```

where $\epsilon_{\mathsf{MAC}} = \epsilon_{\mathsf{MAC}}(\lambda)$ is a negligible function in the security parameter λ, on input message m the oracle $\mathcal{O}_{\mathsf{MAC.Tag}}(\mathsf{sk}_{\mathsf{MAC}}, \cdot)$ returns $\tau \xleftarrow{\$} \mathsf{MAC.Tag}(\mathsf{sk}_{\mathsf{MAC}}, m)$ and the number of queries q is bound by time t.

If $\mathsf{sk}_{\mathsf{MAC}}$ is a one-time authentication key of MAC scheme, then MAC scheme is known as a one-time message authentication code (OTMAC) scheme which is $(1, t, \epsilon_{\mathsf{MAC}})$-secure.

3.5 Fuzzy Extractor

We consider a fuzzy extractor FE that consists of a pair of probabilistic polynomial time (PPT) algorithms $\mathsf{FE} = (\mathsf{FE.Gen}, \mathsf{FE.Rep})$. The FE is associated with metric space $\mathcal{M}_{\mathsf{FE}}$, randomness space $\mathcal{RS}_{\mathsf{FE}}$, extracted string space $\mathcal{ES}_{\mathsf{FE}}$ and helper string space $\mathcal{HS}_{\mathsf{FE}}$. The algorithms of FE are defined as follows:

- $(R, P) \xleftarrow{\$} \mathsf{FE.Gen}(crs, w)$: This is the generation algorithm that takes as input $crs \in \mathcal{RS}_{\mathsf{FE}}$ and $w \in \mathcal{M}_{\mathsf{FE}}$ and outputs an extracted string $R \in \mathcal{ES}_{\mathsf{FE}}$ and a helper string $P \in \mathcal{HS}_{\mathsf{FE}}$. Note that $\mathbf{SD}((R, P), (U_\lambda, P)) \le \epsilon_{\mathsf{FE}}$, where U_λ is uniform distribution on $\{0,1\}^\lambda$.

- $R \xleftarrow{\$} \mathsf{FE.Rep}(w', P)$: This is the reproduce algorithm that takes as input a string $w' \in \mathcal{M}_{\mathsf{FE}}$ and a helper string $P \in \mathcal{HS}_{\mathsf{FE}}$. If $\mathsf{Dist}(w, w')$ is no more than a predetermined threshold ts and $(R, P) \xleftarrow{\$} \mathsf{FE.Gen}(crs, w)$, this algorithm outputs $\mathsf{FE.Rep}(w', P) = R$. Otherwise, no guarantee is provided about the output of $\mathsf{FE.Rep}$.

Definition 5 (Fuzzy Extractor)**.** *Let $\mathcal{W}$ be a family of distributions over metric space $\mathcal{M}_{\mathsf{FE}}$ with $H_\infty(\mathcal{W}) \ge min$, where min is min-entropy of $\mathcal{M}_{\mathsf{FE}}$. We say that a fuzzy extractor $\mathsf{FE} = (\mathsf{FE.Gen}, \mathsf{FE.Rep})$ is $(min, ts, q, t, \epsilon_{\mathsf{FE}})$-secure (indistinguishable), if $|\Pr[\mathsf{EXP}^{ind}_{\mathsf{FE},\mathcal{A}}(\lambda) = 1] - 1/2| \le \epsilon_{\mathsf{FE}}$ holds for all adversaries $\mathcal{A}$ running in time at most t in the following experiment:*

```
$\mathsf{EXP}^{ind}_{\mathsf{FE},\mathcal{A}}(\lambda)$ :
    $crs \xleftarrow{\$} \mathcal{RS}_{\mathsf{FE}}$;
    $w \xleftarrow{\$} \mathcal{W}, U_\lambda \xleftarrow{\$} \{0,1\}^\lambda$;
    $b \xleftarrow{\$} \{0,1\}$;
    $(R^*, P^*) \xleftarrow{\$} \mathsf{FE.Gen}(crs, w)$;
    $R_0 = U_\lambda, R_1 = R^*$;
    $b' \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathsf{FE.Gen}}(\cdot,\cdot)}(crs, R_b, P^*)$;
    if $b = b'$ then return 1,
    and 0 otherwise;
```

```
$\mathcal{O}_{\mathsf{FE.Gen}}(crs', w')$ :
    if $\mathcal{A}$ submits $crs' \neq crs$ and
    $0 < \mathsf{Dist}(w, w') \le ts$,
    $(R_i, P_i) \xleftarrow{\$} \mathsf{FE.Gen}(crs', w')$,
    return $(R_i, P_i)$;
    else, return a failure $\perp$.
```

where $\epsilon_{\mathsf{FE}} = \epsilon_{\mathsf{FE}}(\lambda)$ is a negligible function in the security parameter λ and the number of queries q is always bound by time t.

Definition 6 (DDH Assumption)**.** *We say the DDH assumption holds, given parameters $(\mathbb{G}, p, g)$ where $\mathbb{G}$ is a cyclic group of prime order p and g as a generator of $\mathbb{G}$, if it is hard to distinguish triples of the form (g^x, g^y, g^{xy}) from triples of the form (g^x, g^y, g^z), where x, y, and z are random chosen from $\mathbb{Z}_p^*$. Namely, the DDH problem is $(t, \epsilon_{\mathsf{DDH}})$-hard, if $|\Pr[\mathsf{EXP}^{\mathsf{DDH}}_{\mathbb{G},p,g,\mathcal{A}}(\lambda) = 1] - 1/2| \le \epsilon_{\mathsf{DDH}}$ holds for all adversaries $\mathcal{A}$ running in time at most t in the following experiment:*

```
$\mathsf{EXP}^{\mathsf{DDH}}_{\mathbb{G},p,g,\mathcal{A}}(\lambda)$ :
    $g \xleftarrow{\$} \mathbb{G}, (x, y, z) \xleftarrow{\$} \mathbb{Z}_p^*$;
    $b \xleftarrow{\$} \{0, 1\}$;
    if $b = 0$ then $X \xleftarrow{\$} g^{xy}$, otherwise $X \xleftarrow{\$} g^z$;
    $b' \xleftarrow{\$} \mathcal{A}(\mathbb{G}, p, g, g^x, g^y, X)$;
    return 1, if $b = b'$, and 0 otherwise;
```

where $\epsilon_{\mathsf{DDH}} = \epsilon_{\mathsf{DDH}}(\lambda)$ is a negligible probability in the security parameter λ.

4 Security Model

4.1 Execution Environment

In the execution environment, we fix a set of honest parties $\mathcal{IDS} = \{\mathsf{id}_1, \ldots, \mathsf{id}_l\}$ for $l \in \mathbb{N}$, where id_i ($i \in [l]$) is the identity of client or server. Each identity id_i is associated with a pair of long-term keys $(\mathsf{pk}_i, \mathsf{sk}_i) \in (\mathcal{PK}_{\mathsf{PKE}}, \mathcal{SK}_{\mathsf{PKE}})$. Each honest party id_i can sequentially and concurrently execute the protocol multiple times with different intended partners. We may realize a collection of oracles $\{\Pi^s_{\mathsf{id}_i} : i \in [l], s \in [d]\}$ for $(l, d) \in \mathbb{N}$ that represent the protocol executions of a set of honest parties. Each oracle $\Pi^s_{\mathsf{id}_i}$ works as the s-th protocol instance performed by party id_i. Moreover, we assume each oracle $\Pi^s_{\mathsf{id}_i}$ maintains a list of independent internal state variables with semantics listed in Table 1.

Table 1: Internal states of oracles

Variable	Description
pid^s_i	Identity of id_i's intended partner.
sid^s_i	Session identity of $\Pi^s_{\mathsf{id}_i}$, $\mathsf{sid}^s_i \xleftarrow{\$} \{0,1\}^\lambda$.
Φ^s_i	Internal state of $\Pi^s_{\mathsf{id}_i}$, $\Phi^s_i \in \{\mathsf{accept}, \mathsf{reject}\}$.
K^s_i	Session Key of $\Pi^s_{\mathsf{id}_i}$, $K^s_i \in \mathcal{K}$.
sT^s_i	Transcript of messages sent by $\Pi^s_{\mathsf{id}_i}$.
rT^s_i	Transcript of messages received by $\Pi^s_{\mathsf{id}_i}$.

All those variables of each oracle are initialized with the empty string $\emptyset$. At some point, each oracle $\Pi^s_{\mathsf{id}_i}$ may complete the execution and decide the internal state $\Phi^s_i \in \{\mathsf{accept}, \mathsf{reject}\}$. Additionally, we assume that the real session key is assigned to the variable K^s_i iff oracle $\Pi^s_{\mathsf{id}_i}$ has reached an internal state $\Phi^s_i = \mathsf{accept}$.

4.2 Adversarial Model

The adversary $\mathcal{A}$ considers being a probabilistic polynomial time (PPT) Turing Machine, having complete control of the communication network. The adversary $\mathcal{A}$ could interact with the challenger $\mathcal{C}$ by issuing the following queries:

- $\mathsf{Execute}(\mathsf{id}_i, s, \mathsf{id}_j, t)$: If the client oracle $\Pi^s_{\mathsf{id}_i}$ and server oracle $\Pi^t_{\mathsf{id}_j}$ have not been used, this query will carry out an honest execution of the protocol between two oracles, and return the transcripts sT^s_i and sT^t_j to $\mathcal{A}$. This query models the capability of $\mathcal{A}$ passively eavesdrops on plenty of honest executions.
- $\mathsf{Send}(\mathsf{id}_i, s, m)$: This query allows $\mathcal{A}$ to send a message m of his own choice to the oracle $\Pi^s_{\mathsf{id}_i}$. The oracle $\Pi^s_{\mathsf{id}_i}$ will send back the response message m' (if any) according to the protocol specification and its internal states. After answering a Send query, the variables of $\Pi^s_{\mathsf{id}_i}$ will be updated depending on the specific protocol. This query models active attacks in the real world.
- $\mathsf{Reveal}(\mathsf{id}_i, s)$: If the oracle $\Pi^s_{\mathsf{id}_i}$ has reached an internal state $\Phi^s_i = \mathsf{accept}$ (holding a session key) and a Test query has not been made to $\Pi^s_{\mathsf{id}_i}$ or its partner oracle (if it exists), it responds with the contents of the variable K^s_i. Otherwise, a failure symbol $\perp$ is returned. This query models the leakage of the session key agreed by the two parties.
- $\mathsf{Corrupt}(\mathsf{client}, a)$: This query will respond with the password pwd for $a = 0$, biometric data W for $a = 1$, and private key sk for $a = 2$. By issuing this query, $\mathcal{A}$ could obtain a-th authenticated factor $\{\mathsf{pwd}, \mathsf{W}, \mathsf{sk}\}$ of client. This query models corrupt capabilities of $\mathcal{A}$.
- $\mathsf{Corrupt}(\mathsf{server})$: This query will return server's private key to $\mathcal{A}$.
- $\mathsf{Test}(\mathsf{id}_i, s)$: $\mathcal{C}$ first flips a coin $b \in \{0, 1\}$ uesd for all Test queries. If the oracle $\Pi^s_{\mathsf{id}_i}$ has state $\Phi^s_i = \mathsf{reject}$ or $K^s_i = \emptyset$, then this query returns a failure symbol $\perp$. Otherwise, $\mathcal{C}$ samples a random element K_r from session key space $\mathcal{K}$, and sets $K_0 = K_r$ and $K_1 = K^s_i$. Finally, this query responds with K_b. The oracle $\Pi^s_{\mathsf{id}_i}$ selected by the adversary in this query is called as *test oracle*. This query does not model any actual capabilities of $\mathcal{A}$. It is used to measure the semantic security of session keys.

4.3 Secure AKE Protocols

We first review the notion regarding the partnership of two oracles, i.e. *matching sessions* [2].

Definition 7 (Matching Sessions)**.** *In an MFAKE protocol, we say that the oracle $\Pi^s_{\mathsf{id}_i}$ and oracle $\Pi^t_{\mathsf{id}_j}$ are* matching sessions*, if both of them have been* accept*, hold $(K^s_i, \mathsf{sid}^s_i, \mathsf{pid}^s_i)$ and $(K^t_j, \mathsf{sid}^t_j, \mathsf{pid}^t_j)$, respectively, and all of the following conditions hold:*

1) $\mathsf{sid}^s_i = \mathsf{sid}^t_j$ and $K^s_i = K^t_j$.

2) $\mathsf{id}_i \in \mathsf{client}$, $\mathsf{id}_j \in \mathsf{server}$, and vice versa.

3) $\Pi^s_{\mathsf{id}_i}$ has $\mathsf{pid}^s_i = \mathsf{id}_j$ and $\Pi^t_{\mathsf{id}_j}$ has $\mathsf{pid}^t_j = \mathsf{id}_i$.

4) $sT^s_i = rT^t_j$ and $rT^s_i = sT^t_j$.

Correctness. We say an AKE protocol Π is correct, if an oracle $\Pi^s_{\mathsf{id}_i}$ has a *matching session* to an oracle $\Pi^t_{\mathsf{id}_j}$ and they both accept with the same session key, i.e. $K^s_i = K^t_j$.

To define the security of the session key, we need the notion of *freshness* of an oracle.

Definition 8 (Freshness)**.** *We assume that a* client *instance* $\Pi^s_{\mathsf{id}_i}$ *has been* accept *with its intended* server id_j*. And a* server *instance* $\Pi^t_{\mathsf{id}_j}$ *(if it exists) is an oracle with intended* client id_i*, such that* $\Pi^s_{\mathsf{id}_i}$ *has a matching session to* $\Pi^t_{\mathsf{id}_j}$*. Then the oracle* $\Pi^s_{\mathsf{id}_i}$ *is said to be fresh if none of the following conditions holds:*

1) *$\mathcal{A}$ queried* $\mathsf{Reveal}(\mathsf{id}_i, s)$*.*

2) *If $\Pi^t_{\mathsf{id}_j}$ exists, $\mathcal{A}$ queried* $\mathsf{Reveal}(\mathsf{id}_j, t)$*.*

3) *$\mathcal{A}$ queried* $\mathsf{Corrupt}(\mathsf{client}, a)$ *for all three factors.*

4) *If $\Pi^t_{\mathsf{id}_j}$ exists, $\mathcal{A}$ queried* $\mathsf{Corrupt}(\mathsf{server})$*.*

4.4 Entity Authentication Security Experiment $\mathsf{EXP}^{\mathsf{Ent\text{-}Auth}}_{\Pi,\mathcal{A}}(\lambda)$

The entity authentication security experiment is processed as a game between the challenger $\mathcal{C}$ and adversary $\mathcal{A}$ based on MFAKE protocol Π, where the following steps are performed:

1) With the security parameter λ, the challenger $\mathcal{C}$ first implements the collection of oracles $\{\Pi^s_{\mathsf{id}_i} : i \in [l], s \in [d]\}$, and generates l long-term key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)$ for all honest parties id_i where identity $\mathsf{id}_i \in \mathcal{IDS}$ of each party is chosen uniquely.

2) $\mathcal{A}$ could issue queries to oracles $\mathsf{Execute}$, Send, Reveal and $\mathsf{Corrupt}$ as defined above.

3) Finally, the experiment outputs 1 if and only if there exists Φ^s_i is accept and the following two conditions hold: both id_i and its intended partner id_j were not corrupted before query Test; there is no unique $\Pi^t_{\mathsf{id}_j}$, such that $\Pi^s_{\mathsf{id}_i}$ has a matching session to $\Pi^t_{\mathsf{id}_j}$.

Definition 9 (Entity Authentication)**.** *A correct MFAKE protocol Π is called $(t, \epsilon_{\mathsf{Ent\text{-}Auth}})$-entity-authentication-secure, if for all adversaries $\mathcal{A}$ running within time t in the above MFAKE security experiment $\mathsf{EXP}^{\mathsf{Ent\text{-}Auth}}_{\Pi,\mathcal{A}}(\lambda)$, it holds that:*

$$\Pr[\mathsf{EXP}^{\mathsf{Ent\text{-}Auth}}_{\Pi,\mathcal{A}}(\lambda) = 1] \leq \epsilon_{\mathsf{Ent\text{-}Auth}},$$

where $\epsilon_{\mathsf{Ent\text{-}Auth}} = \epsilon_{\mathsf{Ent\text{-}Auth}}(\lambda)$ is a negligible probability in the security parameter λ.

4.5 Key Indistinguishability Security Experiment $\mathsf{EXP}^{\mathsf{Key\text{-}Ind}}_{\Pi,\mathcal{A}}(\lambda)$

This security experiment is also processed as a game between the challenger $\mathcal{C}$ and adversary $\mathcal{A}$ based on MFAKE protocol Π, where the following steps are performed:

1) With the security parameter λ, the challenger $\mathcal{C}$ first implements the collection of oracles $\{\Pi^s_{\mathsf{id}_i} : i \in [l], s \in [d]\}$, and generates l long-term key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)$ for all honest parties id_i where identity $\mathsf{id}_i \in \mathcal{IDS}$ of each party is chosen uniquely. $\mathcal{C}$ flips a coin $b \in \{0, 1\}$ uesd for all Test queries. $\mathcal{C}$ will give all public parameters to $\mathcal{A}$ and keep track of all variables of the execution environment.

2) $\mathcal{A}$ may interact by issuing the polynomial number of queries as aforementioned, namely, $\mathcal{A}$ makes queries: $\mathsf{Execute}$, Send, Reveal and $\mathsf{Corrupt}$.

3) At some point of time during the game, $\mathcal{A}$ may issue a $\mathsf{Test}(\mathsf{id}_i, s)$ query.

4) $\mathcal{A}$ may continue to make the above queries. The binding constraints on this experiment are that: $\mathcal{A}$ cannot make a Reveal query on either the test session or its partnered session; $\mathcal{A}$ can make $\mathsf{Corrupt}$ query no more than twice if id_i is a client.

5) Finally, $\mathcal{A}$ terminates and outputs its guess b'. The experiment returns 1 if $b = b'$, and 0 otherwise.

Definition 10 (Key Indistinguishability)**.** *A correct MFAKE protocol Π is called $(t', \epsilon_{\mathsf{Key\text{-}Ind}})$-session-key-indistinguishability, if for all adversaries $\mathcal{A}$ running within time t' in the above MFAKE security experiment $\mathsf{EXP}^{\mathsf{Key\text{-}Ind}}_{\Pi,\mathcal{A}}(\lambda)$, it holds that:*

$$|\Pr[\mathsf{EXP}^{\mathsf{Key\text{-}Ind}}_{\Pi,\mathcal{A}}(\lambda) = 1] - 1/2| \leq \epsilon_{\mathsf{Key\text{-}Ind}},$$

where $\epsilon_{\mathsf{Key\text{-}Ind}} = \epsilon_{\mathsf{Key\text{-}Ind}}(\lambda)$ is a negligible probability in the security parameter λ.

5 Security Analysis and Improvement of Zhang's MFAKE Protocol

In this section, we first review Zhang's MFAKE protocol in Figure 1. Then we analyze the drawbacks of Zhang's MFAKE protocol. Finally, an improved scheme is proposed with slight modification on the generation of the session key. The formal security proof of our scheme is provided in the random oracle model.

5.1 Zhang's MFAKE Protocol

This MFAKE scheme [36] is specified by the following algorithms in the sense of definitions in Section 3:

- Public key encryption scheme $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$.

- Message authentication code scheme $\mathsf{MAC} = (\mathsf{MAC.Tag}, \mathsf{MAC.Vfy})$.

- Fuzzy extractor $\mathsf{FE} = (\mathsf{FE.Gen}, \mathsf{FE.Rep})$.

Initialization. Assuming that parameters are $(\mathbb{G}, p, g)$, where $\mathbb{G}$ is a cyclic group of prime order p and g is a generator of $\mathbb{G}$. Each party id_i runs $\mathsf{PKE.KeyGen}$ to generate

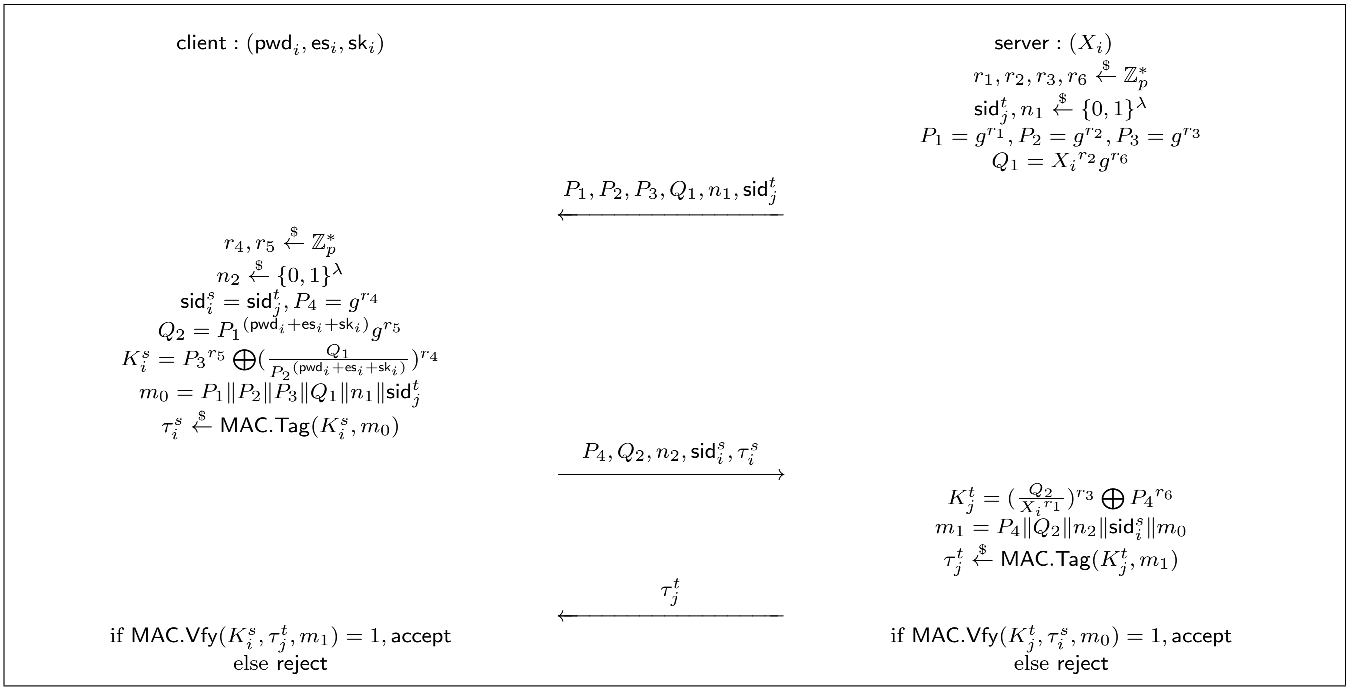
client : $(\mathsf{pwd}_i, \mathsf{es}_i, \mathsf{sk}_i)$
server : (X_i)
$r_1, r_2, r_3, r_6 \overset{\$}{\leftarrow} \mathbb{Z}_p^*$
$\mathsf{sid}_j^t, n_1 \overset{\$}{\leftarrow} \{0,1\}^\lambda$
$P_1 = g^{r_1}, P_2 = g^{r_2}, P_3 = g^{r_3}$
$Q_1 = {X_i}^{r_2} g^{r_6}$
$\xleftarrow{P_1, P_2, P_3, Q_1, n_1, \mathsf{sid}_j^t}$
$r_4, r_5 \overset{\$}{\leftarrow} \mathbb{Z}_p^*$
$n_2 \overset{\$}{\leftarrow} \{0,1\}^\lambda$
$\mathsf{sid}_i^s = \mathsf{sid}_j^t, P_4 = g^{r_4}$
$Q_2 = {P_1}^{(\mathsf{pwd}_i + \mathsf{es}_i + \mathsf{sk}_i)} g^{r_5}$
$K_i^s = {P_3}^{r_5} \bigoplus (\frac{Q_1}{{P_2}^{(\mathsf{pwd}_i + \mathsf{es}_i + \mathsf{sk}_i)}})^{r_4}$
$m_0 = P_1 \| P_2 \| P_3 \| Q_1 \| n_1 \| \mathsf{sid}_j^t$
$\tau_i^s \overset{\$}{\leftarrow} \mathsf{MAC.Tag}(K_i^s, m_0)$
$\xrightarrow{P_4, Q_2, n_2, \mathsf{sid}_i^s, \tau_i^s}$
$K_j^t = (\frac{Q_2}{{X_i}^{r_1}})^{r_3} \bigoplus {P_4}^{r_6}$
$m_1 = P_4 \| Q_2 \| n_2 \| \mathsf{sid}_i^s \| m_0$
$\tau_j^t \overset{\$}{\leftarrow} \mathsf{MAC.Tag}(K_j^t, m_1)$
$\xleftarrow{\tau_j^t}$
if $\mathsf{MAC.Vfy}(K_i^s, \tau_j^t, m_1) = 1$, accept
else reject
if $\mathsf{MAC.Vfy}(K_j^t, \tau_i^s, m_0) = 1$, accept
else reject

Figure 1: MFAKE protocol

key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)$. We denote the public parameters are $((\mathbb{G}, p, g), \mathsf{pk}_i)$, and sk_i is a private key.

Registration. We assume the registration phase accomplishes in a secure channel. A client id_i interacts with the server id_j as following steps:

- The client randomly chooses a password pwd_i from password dictionary $\mathcal{PW}$ and creates a biometric template $\mathsf{W}_i \in \mathcal{M}_{\mathsf{FE}}$. Its private key sk_i is regarded as device data. The client sents $(\mathsf{id}_i, \mathsf{pwd}_i, \mathsf{W}_i, \mathsf{sk}_i)$ to the server.

- The server runs FE.Gen with W_i to obtain an extracted string $\mathsf{es}_i \in \mathcal{ES}$ and a helper string $\mathsf{hs}_i \in \mathcal{HS}$, computes $X_i = g^{(\mathsf{pwd}_i + \mathsf{es}_i + \mathsf{sk}_i)}$, runs PKE.Enc to obtain the ciphertext Y_i of X_i. Then it deletes the template W_i and extracted string es_i and returns hs_i to the client.

- Finally, the client stores hs_i, and the server stores identity id_i of client and Y_i.

Login-Authentication. An honest client id_i first inputs $\mathsf{pwd}_i, \mathsf{W}'_i$, runs FE.Rep and sends an *authentication request* to a server id_j. If there exists a Y_i corresponding to id_i, the server computes $X_i \overset{\$}{\leftarrow} \mathsf{PKE.Dec}(\mathsf{sk}_j, Y_i)$. After that, the client id_i and server id_j hold $(\mathsf{pwd}_i, \mathsf{es}_i, \mathsf{sk}_i)$ and X_i, respectively, where $X_i = g^{(\mathsf{pwd}_i + \mathsf{es}_i + \mathsf{sk}_i)}$. The MFAKE protocol performs as the following steps (as shown in Figure 1):

- The server samples four ephemeral keys r_1, r_2, r_3, r_6 from $\mathbb{Z}_p^*$, a current session identity sid_j^t and a random nonce n_1. Then it computes $P_1 = g^{r_1}$, $P_2 = g^{r_2}$, $P_3 = g^{r_3}$ and $Q_1 = {X_i}^{r_2} g^{r_6}$. The *authentication challenge* $(P_1, P_2, P_3, Q_1, n_1, \mathsf{sid}_j^t)$ sends to the client.

- After receiving the *authentication challenge*, the client samples two ephemeral keys r_4, r_5 from $\mathbb{Z}_p^*$ and a random element n_2. It sets $\mathsf{sid}_i^s = \mathsf{sid}_j^t$, computes $P_4 = g^{r_4}$, $Q_2 = {P_1}^{(\mathsf{pwd}_i + \mathsf{es}_i + \mathsf{sk}_i)} g^{r_5}$ and $K_i^s = {P_3}^{r_5} \bigoplus (\frac{Q_1}{{P_2}^{(\mathsf{pwd}_i + \mathsf{es}_i + \mathsf{sk}_i)}})^{r_4}$. The client runs MAC.Tag to generate a tag τ_i^s of $m_0 = P_1 \| P_2 \| P_3 \| Q_1 \| n_1 \| \mathsf{sid}_j^t$, and sends $(P_4, Q_2, n_2, \mathsf{sid}_i^s, \tau_i^s)$ as *authentication response* to server.

- After receiving the *authentication response*, the server can compute $K_j^t = (\frac{Q_2}{{X_i}^{r_1}})^{r_3} \bigoplus {P_4}^{r_6}$. It runs MAC.Tag to generate a tag τ_j^t of $m_1 = P_4 \| Q_2 \| n_2 \| \mathsf{sid}_i^s \| m_0$, and sends τ_j^t to the client.

- Finally, the client and server run $\mathsf{MAC.Vfy}(K_i^s, \tau_j^t, m_1)$ and $\mathsf{MAC.Vfy}(K_j^t, \tau_i^s, m_0)$, respectively. Φ_i^s or Φ_j^t sets to be accept if the output is 1, and reject otherwise.

5.2 The Insecurity of Zhang's MFAKE Scheme

Man-in-the-Middle Attack. In the following, we present a Man-in-the-Middle (MITM) attack on Zhang's MFAKE scheme. We assume that an adversary $\mathcal{A}$ intervenes in communication between the client and server. $\mathcal{A}$ could receive, forward, and modify the message exchanged between them.

The concrete MITM attack steps are performed as below:

1) $\mathcal{A}$ arbitrarily chooses client oracle $\Pi_{\mathsf{id}_i}^s$ and server oracle $\Pi_{\mathsf{id}_j}^t$ as target oracles.

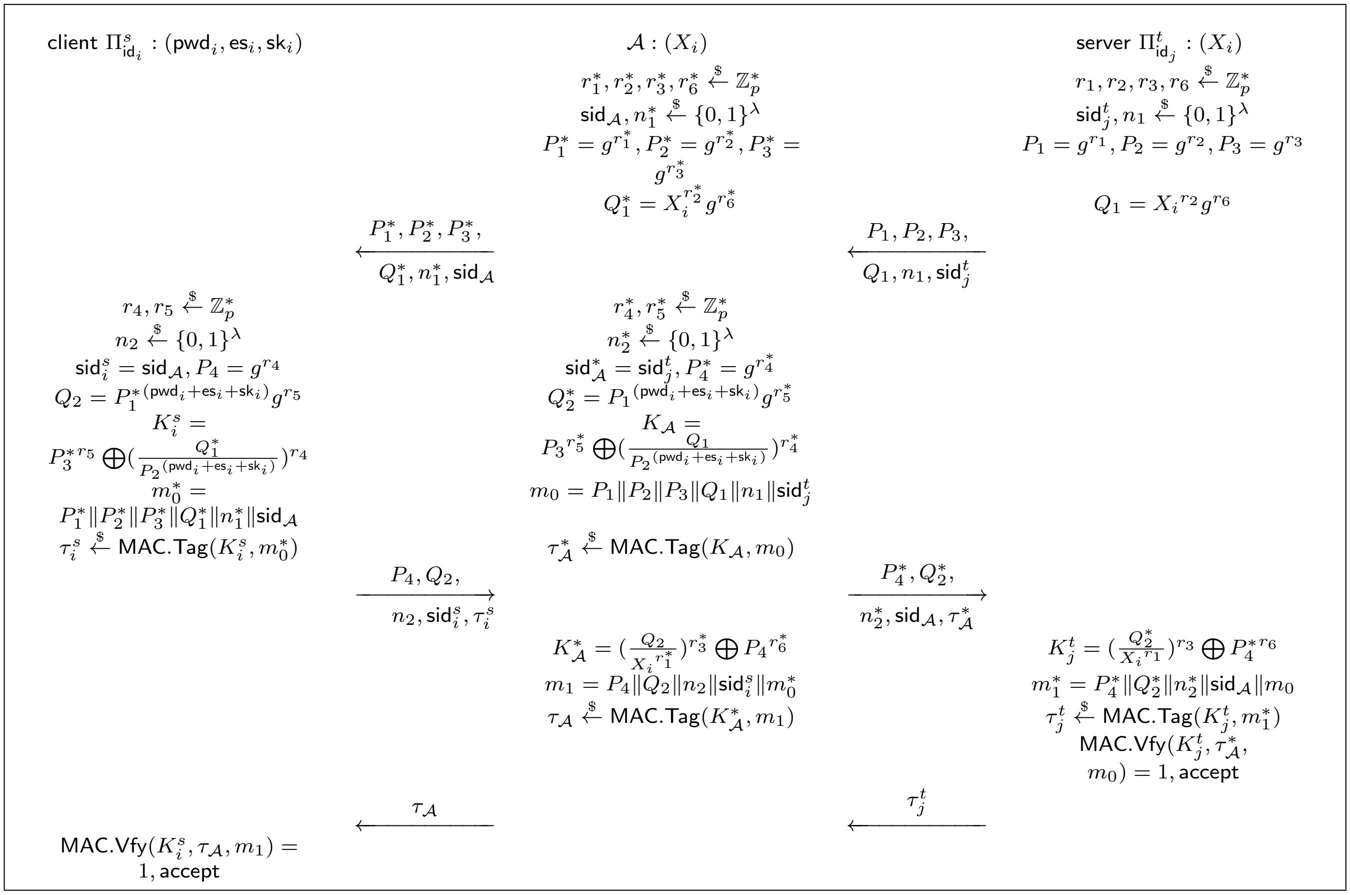
client $\Pi^s_{\mathsf{id}_i} : (\mathsf{pwd}_i, \mathsf{es}_i, \mathsf{sk}_i)$
$\mathcal{A} : (X_i)$
server $\Pi^t_{\mathsf{id}_j} : (X_i)$
$r_1^*, r_2^*, r_3^*, r_6^* \xleftarrow{\$} \mathbb{Z}_p^*$
$\mathsf{sid}_\mathcal{A}, n_1^* \xleftarrow{\$} \{0,1\}^\lambda$
$P_1^* = g^{r_1^*}, P_2^* = g^{r_2^*}, P_3^* = g^{r_3^*}$
$Q_1^* = X_i^{r_2^*} g^{r_6^*}$
$r_1, r_2, r_3, r_6 \xleftarrow{\$} \mathbb{Z}_p^*$
$\mathsf{sid}_j^t, n_1 \xleftarrow{\$} \{0,1\}^\lambda$
$P_1 = g^{r_1}, P_2 = g^{r_2}, P_3 = g^{r_3}$
$Q_1 = X_i^{r_2} g^{r_6}$
$P_1^*, P_2^*, P_3^*, Q_1^*, n_1^*, \mathsf{sid}_\mathcal{A}$
$P_1, P_2, P_3, Q_1, n_1, \mathsf{sid}_j^t$
$r_4, r_5 \xleftarrow{\$} \mathbb{Z}_p^*$
$n_2 \xleftarrow{\$} \{0,1\}^\lambda$
$\mathsf{sid}_i^s = \mathsf{sid}_\mathcal{A}, P_4 = g^{r_4}$
$Q_2 = P_1^{*(\mathsf{pwd}_i+\mathsf{es}_i+\mathsf{sk}_i)} g^{r_5}$
$K_i^s = P_3^{*r_5} \bigoplus (\frac{Q_1^*}{P_2^{(\mathsf{pwd}_i+\mathsf{es}_i+\mathsf{sk}_i)}})^{r_4}$
$m_0^* = P_1^* \| P_2^* \| P_3^* \| Q_1^* \| n_1^* \| \mathsf{sid}_\mathcal{A}$
$\tau_i^s \xleftarrow{\$} \mathsf{MAC.Tag}(K_i^s, m_0^*)$
$r_4^*, r_5^* \xleftarrow{\$} \mathbb{Z}_p^*$
$n_2^* \xleftarrow{\$} \{0,1\}^\lambda$
$\mathsf{sid}_\mathcal{A}^* = \mathsf{sid}_j^t, P_4^* = g^{r_4^*}$
$Q_2^* = P_1^{(\mathsf{pwd}_i+\mathsf{es}_i+\mathsf{sk}_i)} g^{r_5^*}$
$K_\mathcal{A} = P_3^{r_5^*} \bigoplus (\frac{Q_1}{P_2^{(\mathsf{pwd}_i+\mathsf{es}_i+\mathsf{sk}_i)}})^{r_4^*}$
$m_0 = P_1 \| P_2 \| P_3 \| Q_1 \| n_1 \| \mathsf{sid}_j^t$
$\tau_\mathcal{A}^* \xleftarrow{\$} \mathsf{MAC.Tag}(K_\mathcal{A}, m_0)$
$P_4, Q_2, n_2, \mathsf{sid}_i^s, \tau_i^s$
$P_4^*, Q_2^*, n_2^*, \mathsf{sid}_\mathcal{A}, \tau_\mathcal{A}^*$
$K_\mathcal{A}^* = (\frac{Q_2}{X_i^{r_1^*}})^{r_3^*} \bigoplus P_4^{r_6^*}$
$m_1 = P_4 \| Q_2 \| n_2 \| \mathsf{sid}_i^s \| m_0^*$
$\tau_\mathcal{A} \xleftarrow{\$} \mathsf{MAC.Tag}(K_\mathcal{A}^*, m_1)$
$K_j^t = (\frac{Q_2^*}{X_i^{r_1}})^{r_3} \bigoplus P_4^{*r_6}$
$m_1^* = P_4^* \| Q_2^* \| n_2^* \| \mathsf{sid}_\mathcal{A} \| m_0$
$\tau_j^t \xleftarrow{\$} \mathsf{MAC.Tag}(K_j^t, m_1^*)$
$\mathsf{MAC.Vfy}(K_j^t, \tau_\mathcal{A}^*, m_0) = 1$, accept
$\tau_\mathcal{A}$
τ_j^t
$\mathsf{MAC.Vfy}(K_i^s, \tau_\mathcal{A}, m_1) = 1$, accept

Figure 2: Outsider KCI attack

2) $\mathcal{A}$ asks $\Pi^s_{\mathsf{id}_i}$ to execute the protocol instance.

3) $\mathcal{A}$ intercepts $(P_4, Q_2, n_2, \mathsf{sid}_i^s, \tau_i^s)$ and changes n_2 to n_3, where $n_3 \in \{0,1\}^\lambda$ is randomly chosen by $\mathcal{A}$.

4) $\mathcal{A}$ does not forge the keying materials of session key. Thus $\Pi^t_{\mathsf{id}_j}$ could compute a session key $K_j^t = K_i^s$ and accept for $\mathsf{MAC.Vfy}(K_j^t, \tau_i^s, m_0) = 1$.

5) At this moment, however, $sT_i^s \neq rT_j^t$, the oracle $\Pi^s_{\mathsf{id}_i}$ doesn't have a *matching session* to an oracle $\Pi^t_{\mathsf{id}_j}$.

6) $\mathcal{A}$ could queries $\mathsf{Reveal}(\mathsf{id}_j, t)$ to get the session key K_j^t. Then $\mathcal{A}$ generates a tag $\tau_j^{t^*}$ of $m_1 = P_4 \| Q_2 \| n_2 \| \mathsf{sid}_i^s \| m_0$ to make $\Pi^s_{\mathsf{id}_i}$ be accept. $K_j^t = K_i^s$ means that $\mathcal{A}$ has the session key K_i^s of oracle $\Pi^s_{\mathsf{id}_i}$ while $\Pi^s_{\mathsf{id}_i}$ is fresh.

7) Finally, $\mathcal{A}$ can query $\mathsf{Test}(\mathsf{id}_i, s)$ and wins the game by comparing K_b with K_i^s.

Outsider KCI Attack. In the following, we show that if $\mathcal{A}$ corrupts the server id_j, it could impersonate an uncorrupted client id_i to the server id_j. $\mathcal{A}$ corrupts id_j to get X_i (this is allowed due to the modeling of KCI attacks) and behaves as if the server interacts with the client. We use the superscript * of a value to be an element chosen by $\mathcal{A}$. Then $\mathcal{A}$ could get the session key $K_\mathcal{A}^* = g^{r_3^* r_5} \bigoplus P_4^{r_6^*} = K_i^s$ just like the server. $\mathcal{A}$ then computes g^{r_5} since it has r_3^*, r_6^*, P_4. The keying material $P_1^{*(\mathsf{pwd}_i+\mathsf{es}_i+\mathsf{sk}_i)}$ is easily computed from $Q_2 = P_1^{*(\mathsf{pwd}_i+\mathsf{es}_i+\mathsf{sk}_i)} g^{r_5}$ and it leads the protocol insecure. $\mathcal{A}$ could violate the security of the MFAKE protocol via the following steps:

1) $\mathcal{A}$ first chooses a client oracle $\Pi^s_{\mathsf{id}_i}$ and a server oracle $\Pi^t_{\mathsf{id}_j}$ and executes the MFAKE protocol instances between them.

2) $\mathcal{A}$ corrupts the oracle $\Pi^t_{\mathsf{id}_j}$ to get X_i, and intercepts $P_1, P_2, P_3, Q_1, n_1, \mathsf{sid}_j^t$.

3) Meanwhile, $\mathcal{A}$ executes protocol instance with the client id_i. If $\mathcal{A}$ replaces P_1^* with P_1, it can get $P_1^{(\mathsf{pwd}_i+\mathsf{es}_i+\mathsf{sk}_i)}$. If $\mathcal{A}$ replaces P_1^* with P_2, it can get $P_2^{(\mathsf{pwd}_i+\mathsf{es}_i+\mathsf{sk}_i)}$.

4) $\mathcal{A}$ computes $Q_2^* = P_1^{(\mathsf{pwd}_i+\mathsf{es}_i+\mathsf{sk}_i)} g^{r_5^*}$ and $K_\mathcal{A} = P_3^{r_5^*} \bigoplus (\frac{Q_1}{P_2^{(\mathsf{pwd}_i+\mathsf{es}_i+\mathsf{sk}_i)}})^{r_4^*}$. $\mathcal{A}$ generates a tag $\tau_\mathcal{A}^*$ of message m_0, and sends $P_4^*, Q_2^*, n_2^*, \mathsf{sid}_\mathcal{A}, \tau_\mathcal{A}^*$ to $\Pi^t_{\mathsf{id}_j}$. The oracle $\Pi^t_{\mathsf{id}_j}$ would compute $K_j^t = K_\mathcal{A}$ and accept the session but it does not have a *matching session* to $\Pi^s_{\mathsf{id}_i}$.

5) $\mathcal{A}$ selects the oracle $\Pi^t_{\mathsf{id}_j}$ as the test oracle which should generate the session key K_j^t. Then $\mathcal{A}$ could win the game by impersonating a client and computing the session key $K_\mathcal{A} = K_j^t$.

The details of this attack are shown in Figure 2. $\mathcal{A}$ succeeds in impersonating the honest $\mathsf{client}\ \mathsf{id}_i$ to $\mathsf{server}\ \mathsf{id}_j$'s oracle $\Pi^t_{\mathsf{id}_j}$ and id_i has no *matching session* to $\Pi^t_{\mathsf{id}_j}$.

5.3 An Improvement Solution of Zhang's MFAKE Scheme

We have shown that Zhang's MFAKE scheme is vulnerable to MITM and outsider KCI attacks since the protocol message transcript is not fully bound to the keying material. We are trying to circumvent the above attacks by modifying the key derivation function. A hash function takes as input $K_i(K_j), n_1, n_2, \mathsf{sid}^s_i(\mathsf{sid}^t_j)$ and outputs the session key $K^s_i(K^t_j)$. More specifically, our improved scheme is shown in Figure 3.

Theorem 1. *Suppose that the message authentication code scheme* MAC *is* $(1, t, \epsilon_{\mathsf{MAC}})$*-secure against strongly existential forgeries under chosen message attacks. Then our improved MFAKE scheme is* $(t, \epsilon_{\mathsf{Ent\text{-}Auth}})$*-entity-authentication-secure provided that*

$$\epsilon_{\mathsf{Ent\text{-}Auth}} \leq \frac{(9dl)^2}{2^\lambda} + dl \cdot \epsilon_{\mathsf{MAC}}.$$

Proof. We consider the proof following a sequence of games. Generally speaking, the values processed in $\Pi^{s^*}_{\mathsf{id}_i}$ are highlighted with *. Let S_ξ be the event that the adversary wins the security experiment in Game ξ, and $\mathsf{Adv}_\xi = \Pr[\mathsf{S}_\xi]$ denotes the advantage of $\mathcal{A}$ in Game ξ.

Game 0. This is the original entity authentication security game between $\mathcal{A}$ and $\mathcal{C}$. So we can write the following:

$$\mathsf{Adv}_0 = \Pr[\mathsf{S}_0].$$

Game 1. The challenger proceeds exactly like the previous game but aborts if event E_1 happens, where E_1 denotes two oracles generate the nonce, $((r^*_1, r^*_2, r^*_3, r^*_6, n^*_1, \mathsf{sid}^{t^*}_j), (r^*_4, r^*_5, n^*_2))$, which has been sampled before. The probability of the collision of those values is negligible since the nonces are chosen uniformly at random. There are l parties and at most d oracles for each party, the birthday paradox results provide that the event E_1 happens with the probability $\Pr[\mathsf{E}_1] \leq \frac{(9dl)^2}{2^\lambda}$. Thus we have that

$$\mathsf{Adv}_0 \leq \mathsf{Adv}_1 + \frac{(9dl)^2}{2^\lambda}.$$

Game 2. In this game, $\mathcal{C}$ aborts when event E_2 happens. We define the event E_2 which happens if $\Pi^s_{\mathsf{id}_i}$ receives messages with a valid tag τ^t_j which is not send by its intended partner oracle $\Pi^t_{\mathsf{id}_j}$. We have $\mathsf{Adv}_1 \leq \mathsf{Adv}_2 + \Pr[\mathsf{E}_2]$.

If the event E_2 happens with overwhelming probability, then we could construct a tag forger $\mathcal{F}_2$ against the security of the message authentication code scheme as follows. The forger $\mathcal{F}_2$ simulates the challenger for $\mathcal{A}$. It first guesses an oracle that the adversary can forge, i.e. $\Pi^{t^*}_{\mathsf{id}_j}$. Next $\mathcal{F}_2$ generates all other secret keys honestly as the challenger in the previous game. If $\mathcal{A}$ outputs a message with a valid tag not generated by $\mathcal{F}_2$, then $\mathcal{F}_2$ could use the tag to break security. Since there are at most dl oracles for all parties, the event E_2 happens with the probability $\Pr[\mathsf{E}_2] \leq dl \cdot \epsilon_{\mathsf{MAC}}$. Thus it holds that

$$\mathsf{Adv}_1 \leq \mathsf{Adv}_2 + dl \cdot \epsilon_{\mathsf{MAC}}.$$

Summing up all the probabilities from Game 0 to Game 2, we hold the result of Theorem 1.

□

Theorem 2. *Suppose that the public key scheme* PKE *is* $(d, t, \epsilon_{\mathsf{PKE}})$*-secure against adaptive chosen-ciphertext attacks, the fuzzy extractor* FE *is* $(min, ts, d, t, \epsilon_{\mathsf{FE}})$*-secure, the hash function* h *is collision-resistant and the DDH problem is* $(t, \epsilon_{\mathsf{DDH}})$*-hard. Assume that the bit-length of* pwd *is* μ_1*, the bit-length of* W *is* μ_2*, and the bit-length of* sk *is* μ_3*. Then the improved MFAKE scheme is* (t', ϵ)*-session-key-secure with* $t \approx t'$ *and*

$$\epsilon_{\mathsf{Key\text{-}Ind}} \leq \epsilon_{\mathsf{Ent\text{-}Auth}} + dl \cdot (max\{\frac{1}{2^{\mu_1}}, \epsilon_{\mathsf{FE}}, \frac{1}{2^{\mu_3}}\} + 2\epsilon_{\mathsf{DDH}}).$$

Proof. We consider the proof following a sequence of games. $\mathcal{A}$ chooses the test oracle $\Pi^{s^*}_{\mathsf{id}_i}$ executed between its owner id_i and its intended partner id_j. Generally speaking, the values processed in $\Pi^{s^*}_{\mathsf{id}_i}$ are highlighted with *. Let S_ξ be the event that the adversary wins the security experiment in Game ξ, and $\mathsf{Adv}_\xi = \Pr[\mathsf{S}_\xi] - \frac{1}{2}$ denotes the advantage of $\mathcal{A}$ in Game ξ.

Game 0. This is the original security game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The bit b is chosen at the beginning of Game 0. $\mathcal{C}$ will answer the queries of $\mathcal{A}$ on behalf of the instances. By definition, it holds that

$$\Pr[\mathsf{S}_0] = \frac{1}{2} + \epsilon = \frac{1}{2} + \mathsf{Adv}_0.$$

Game 1. The challenger proceeds exactly like the previous game but aborts if event E_1 happens, where E_1 denotes an oracle accepts maliciously. From Theorem 1, we have that

$$\mathsf{Adv}_0 \leq \mathsf{Adv}_1 + \epsilon_{\mathsf{Ent\text{-}Auth}}.$$

Game 2. In this game, $\mathcal{C}$ aborts if $\mathcal{A}$ asks the Send query with client's keys $(\mathsf{pwd}^*_i, \mathsf{es}^*_i, \mathsf{sk}^*_i)$ or server's key X^*_i. We let $pes^* = \mathsf{pwd}^*_i + \mathsf{es}^*_i + \mathsf{sk}^*_i$. Due to definition of three-factors security, $\mathcal{A}$ can only compromise two factors. Since there are l parties and at most d oracles for each party, $\mathcal{A}$ can ask dl Send queries. The three possible cases might occur as follows:

1) If W^*_i and sk^*_i are leaked, $\mathcal{A}$ could try to guess low-entropy passwords using the password dictionary attacks. $\mathcal{A}$ could guess correctly in this case with probability $\frac{dl}{2^{\mu_1}}$.

2) If pwd^*_i and sk^*_i are leaked, $\mathcal{A}$ could guess the extracted string es^*_i from helper string hs^*_i with the $\mathsf{FE.Rep}(\cdot)$ function. Due to the use of the fuzzy extractor, $\mathcal{A}$ has an additional advantage ϵ_{FE}. Namely, $\mathcal{A}$ could guess correctly in this case with probability $dl \cdot \epsilon_{\mathsf{FE}}$.

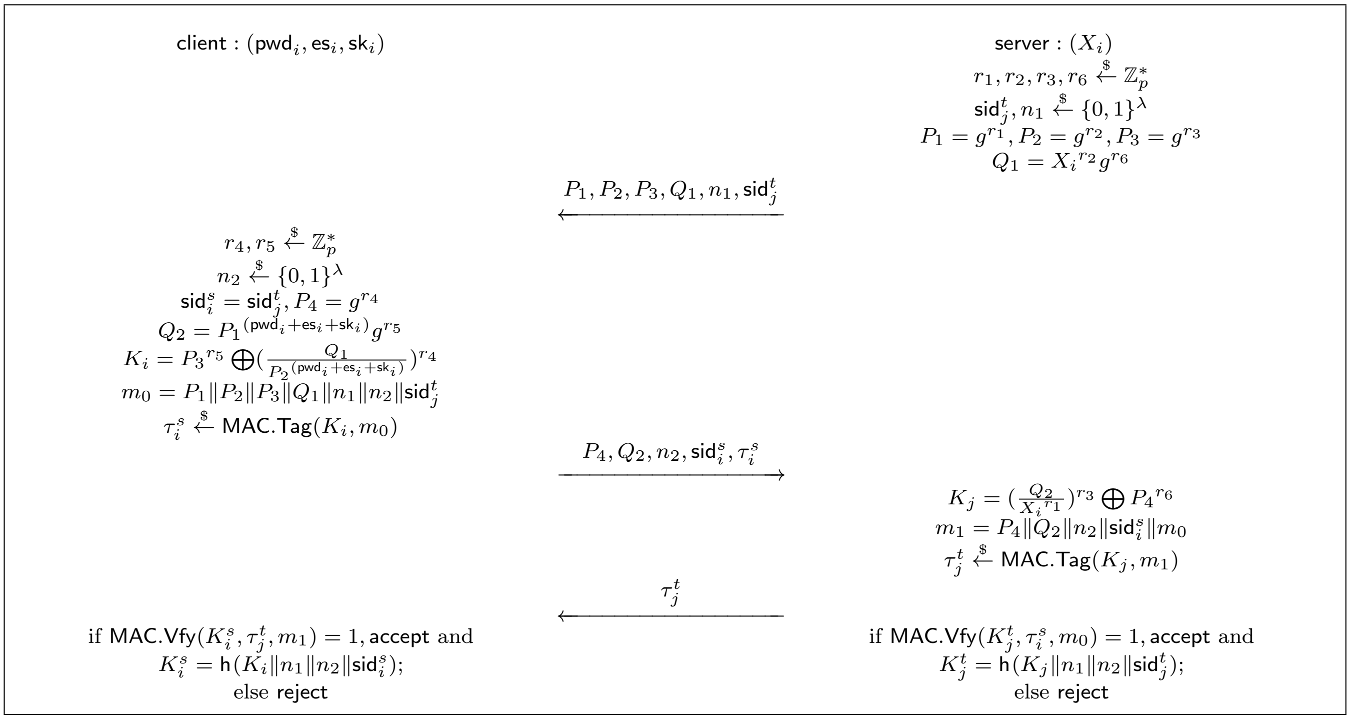
$\mathsf{client} : (\mathsf{pwd}_i, \mathsf{es}_i, \mathsf{sk}_i)$
$\mathsf{server} : (X_i)$
$r_1, r_2, r_3, r_6 \xleftarrow{\$} \mathbb{Z}_p^*$
$\mathsf{sid}_j^t, n_1 \xleftarrow{\$} \{0,1\}^\lambda$
$P_1 = g^{r_1}, P_2 = g^{r_2}, P_3 = g^{r_3}$
$Q_1 = {X_i}^{r_2} g^{r_6}$
$\xleftarrow{P_1, P_2, P_3, Q_1, n_1, \mathsf{sid}_j^t}$
$r_4, r_5 \xleftarrow{\$} \mathbb{Z}_p^*$
$n_2 \xleftarrow{\$} \{0,1\}^\lambda$
$\mathsf{sid}_i^s = \mathsf{sid}_j^t, P_4 = g^{r_4}$
$Q_2 = {P_1}^{(\mathsf{pwd}_i + \mathsf{es}_i + \mathsf{sk}_i)} g^{r_5}$
$K_i = {P_3}^{r_5} \bigoplus (\frac{Q_1}{{P_2}^{(\mathsf{pwd}_i + \mathsf{es}_i + \mathsf{sk}_i)}})^{r_4}$
$m_0 = P_1 \| P_2 \| P_3 \| Q_1 \| n_1 \| n_2 \| \mathsf{sid}_j^t$
$\tau_i^s \xleftarrow{\$} \mathsf{MAC.Tag}(K_i, m_0)$
$\xrightarrow{P_4, Q_2, n_2, \mathsf{sid}_i^s, \tau_i^s}$
$K_j = (\frac{Q_2}{{X_i}^{r_1}})^{r_3} \bigoplus {P_4}^{r_6}$
$m_1 = P_4 \| Q_2 \| n_2 \| \mathsf{sid}_i^s \| m_0$
$\tau_j^t \xleftarrow{\$} \mathsf{MAC.Tag}(K_j, m_1)$
$\xleftarrow{\tau_j^t}$
if $\mathsf{MAC.Vfy}(K_i^s, \tau_j^t, m_1) = 1, \mathsf{accept}$ and
$K_i^s = \mathsf{h}(K_i \| n_1 \| n_2 \| \mathsf{sid}_i^s)$;
else reject
if $\mathsf{MAC.Vfy}(K_j^t, \tau_i^s, m_0) = 1, \mathsf{accept}$ and
$K_j^t = \mathsf{h}(K_j \| n_1 \| n_2 \| \mathsf{sid}_j^t)$;
else reject

Figure 3: Improved MFAKE protocol

3) If pwd_i^* and W_i^* are leaked, $\mathcal{A}$ still has no information about sk_i^* which means pes^* is still random for $\mathcal{A}$. $\mathcal{A}$ could guess correctly in this case with probability $\frac{dl}{2^{\mu_3}}$.

Then, we have that

$$\mathsf{Adv}_1 \leq \mathsf{Adv}_2 + dl \cdot max\{\frac{1}{2^{\mu_1}}, \epsilon_{\mathsf{FE}}, \frac{1}{2^{\mu_3}}\}.$$

Game 3. In this game, $\mathcal{C}$ change the computations of Q_1^* and Q_2^* by $Q_1^* = g^{r_6^*}$ and $Q_2^* = g^{r_5^*}$. Similarly, the computations of K_i^* and K_j^* change to $K_i^* = P_3^{*r_5^*} \bigoplus Q_1^{*r_4^*}$ and $K_j^* = Q_2^{*r_3^*} \bigoplus P_4^{*r_6^*}$. We change this game that $\mathcal{C}$ will answer the Test oracle with a random key and abort if event E_3 happens. We define the event E_3 which happens if $\mathcal{A}$ asks hash oracle with valid K_i^*. If E_3 happens with non-negligible probability, we can build an algorithm $\mathcal{A}_3$ against the DDH challenge. The $\mathcal{A}_3$ receives values (g^x, g^y, g^z) such that either $z = xy$ or $z \xleftarrow{\$} \mathbb{Z}_p^*$ and runs the adversary $\mathcal{A}$ as a subroutine. If $\mathcal{A}_3$ receives a Diffie-Hellman triple, this game proceeds exactly as Game 2, otherwise it is identical to Game 3. If $\mathcal{A}$ can distinguish with non-negligible probability whether $g^z = g^{xy}$ or not, then $\mathcal{A}_3$ can use $\mathcal{A}$ to break the DDH assumption. There are at most dl oracles for all parties. Due to the security of DDH assumption, it holds that

$$\mathsf{Adv}_2 \leq \mathsf{Adv}_3 + 2dl \cdot \epsilon_{\mathsf{DDH}}.$$

In this game, the answer of each Test query is a random key that is independent of the bit b. Thus, the advantage that $\mathcal{A}$ wins is $\mathsf{Adv}_3 = 0$.

Summing up all the probabilities from Game 0 to Game 3, we hold the result of Theorem 2.

□

Table 2: Security features

	[27]	[11]	[7]	[36]	[28]	[9]	Ours
Session key security	×	×	×	×	√	√	√
Mutual authentication	√	√	√	√	√	√	√
Impersonation attack resilience	×	√	×	×	×	×	√
Man-in-the-Middle attack resilience	×	√	×	×	×	×	√
Replay attack resilience	√	√	√	√	√	√	√
Forward secrecy	√	×	√	√	√	√	√
Password guessing attack resilience	√	√	√	√	√	√	√
Biometrics template privacy	×	√	×	√	√	√	√
Stolen smartcard attack resilience	×	×	√	√	√	√	√
Known session key security	√	×	√	√	√	√	√

6 Comparison

In this section, we compare our improved protocol with some proposed MFAKE schemes in terms of security features and performance, i.e. Pointcheval-Zimmer [27], Huang *et al.* [11], Fleischhacker *et al.* [7], Zhang *et al.* [36], Wan *et al.* [28] and Guo *et al.* [9].

6.1 Security Features

The major security properties of the listed schemes are shown in Table 2. The terms √/× represent that a security property is satisfied/unsatisfied by a protocol.

In Table 2, the protocols [7, 11, 27, 36] fail to provide session key security. The protocols [7, 9, 27, 28, 36] cannot resist impersonation attack and Man-in-the-Middle attack. Huang *et al.* [11] cannot provide forward secrecy and known session key security. Biometrics template privacy leaks in Pointcheval-Zimmer [27] and Fleischhacker *et al.* [7] protocols. Pointcheval-Zimmer [27] and Huang *et al.* [11] are vulnerable to stolen smartcard attack. In contrast to these protocols, it is easy to see that our scheme can provide all of those security properties as listed in

Table 3: Performance comparison

	Computation Cost		Communication Cost (Bytes)		Storage Cost (Bytes)		Rounds
	Client	Server	Client	Server	Client	Server	
[27]	$(2N+4)T_{\mathsf{pm}} + NT_{\mathsf{h}}$	$(2N+4)T_{\mathsf{pm}} + NT_{\mathsf{h}}$	60 + 20N	60 + 60N	20 + N/8	80 + 80N	4
[11]	$T_{\mathsf{pm}} + T_{\mathsf{FE}} + T_{\mathsf{SIG}} + T_{\mathsf{PKE}} + T_{\mathsf{MAC}} + 2T_{\mathsf{h}}$	$T_{\mathsf{pm}} + T_{\mathsf{SIG}} + T_{\mathsf{PKE}} + T_{\mathsf{MAC}} + T_{\mathsf{h}}$	232	120	192	80	3
[7]	$9T_{\mathsf{pm}} + T_{\mathsf{SIG}} + 2T_{\mathsf{PKE}} + (N+7)T_{\mathsf{h}}$	$9T_{\mathsf{pm}} + T_{\mathsf{SIG}} + 2T_{\mathsf{PKE}} + (N+7)T_{\mathsf{h}}$	204 + 32N	172	40	80 + N/8	6
[36]	$6T_{\mathsf{pm}} + T_{\mathsf{FE}} + 2T_{\mathsf{MAC}}$	$8T_{\mathsf{pm}} + T_{\mathsf{PKE}} + 2T_{\mathsf{MAC}}$	140	220	40	60	3
[28]	$2T_{\mathsf{pm}} + T_{\mathsf{FE}} + T_{\mathsf{PKE}} + 9T_{\mathsf{h}}$	$2T_{\mathsf{pm}} + T_{\mathsf{FE}} + T_{\mathsf{PKE}} + 5T_{\mathsf{h}}$	100	40	120	40	3
[9]	$3T_{\mathsf{pm}} + T_{\mathsf{FE}} + T_{\mathsf{SIG}} + 7T_{\mathsf{h}}$	$3T_{\mathsf{pm}} + T_{\mathsf{SIG}} + 4T_{\mathsf{h}}$	132	112	176	120	2
Ours	$6T_{\mathsf{pm}} + T_{\mathsf{FE}} + 2T_{\mathsf{MAC}} + T_{\mathsf{h}}$	$8T_{\mathsf{pm}} + T_{\mathsf{PKE}} + 2T_{\mathsf{MAC}} + T_{\mathsf{h}}$	140	220	40	60	3

Table 2.

6.2 Performance Evaluation

In Table 3, we will compare the performance of our scheme with schemes [7, 9, 11, 27, 28, 36] in terms of computation cost, communication cost, storage cost, and rounds, respectively. We consider the computation cost and communication cost of the login and authentication phase. The experiment uses 160 bits standard elliptic curve as in [36]. To compare the computation cost, we define the time of the primary functions required in each protocol. Let T_{pm} denote the time of executing an elliptic curve point multiplication operation. Let T_{FE} denote the time of executing a fuzzy extractor/reproduce operation. Let T_{SIG} denote the time of executing a signature/verify operation. Let T_{PKE} denote the time of executing a encryption/decryption operation. Let T_{MAC} denote the time of executing a tag/verify operation. Let T_{h} denote the time of executing a one-way hash function operation.

For computing the communication and storage overhead, the length of elliptic curve group value, random nonce, and session identity are 160 bits, respectively. Message authentication code is instantiated with HMAC-SHA1, which outputs 160 bits hash value. We simulate the hash function with SHA256 hashing algorithm, which outputs 256 bits hash value. N is bit length of biometrics.

In terms of computation cost, our protocol is more efficient than Pointcheval-Zimmer [27] and Fleischhacker *et al.* [7] protocols, and almost equal to Zhang *et al.* [36] protocol. Our protocol takes more computation cost than Huang *et al.* [11], Wan *et al.* [28] and Guo *et al.* [9]. In contrast to our protocol, however, the protocol in [11] includes a multi-factor authentication without providing session key agreement and is insecure against stolen smartcard attack. Furthermore, our protocol resists impersonation attack and Man-in-the-Middle attack, which are not satisfied in Wan *et al.* [28] and Guo *et al.* [9] protocol. Although our scheme is less efficient than Guo *et al.* [9] in terms of communication cost and rounds, we can provide more security properties. Our scheme takes more communication cost than Wu *et al.* [28]. However, storage cost of our scheme is more efficient. And our scheme resists impersonation attack and Man-in-the-Middle attack, which is not provided by Wu *et al.* [28]. The storage cost of our protocol is optimal.

6.3 Summary

Our protocol satisfies all security features, while each of the others has some weaknesses. We argue that our protocol is more in accordance with the actual application requirements while ensuring security and efficiency.

7 Conclusion

In this paper, we have studied the MFAKE protocol proposed by Zhang *et al.* [36]. As described above, we prove that the security of the MFAKE protocol has some flaws. A simple Man-in-the-Middle attack and an outsider Key Compromise Impersonation attack have been shown in detail. To remedy these weaknesses, an improvement MFAKE scheme has been proposed, which is secure against the attacks mentioned above. The security of the improved protocol was verified in the random oracle model. The results of the formal security proof, security features, and performance evaluation show our improved protocol is more suitable for practical application.

Acknowledgments

Portions of this work were presented at the CIMSS of ACNS workshop in 2022. We would like to thank Zengpeng Li for insightful comments and discussions. This work was supported by the Natural Science Foundation of China (Grant No.61872051) and the Action Plan for high-quality Development of Graduate Education of Chongqing University of Technology (No.gzlcx20223226).

References

[1] M. Abdalla, P. Fouque, and D. Pointcheval, “Password-based authenticated key exchange in the three-party setting,” in *8th International Workshop on Theory and Practice in Public Key Cryptography (PKC 2005)*, pp. 65–84, Les Diablerets, Switzerland, January 2005.

[2] M. Bellare, D. Pointcheval, and P. Rogaway, “Authenticated key exchange secure against dictionary attacks,” in *International Conference on the Theory*

and Application of Cryptographic Techniques (EUROCRYPT 2000), pp. 139–155, Bruges, Belgium, May 2000.

[3] M. Bellare and Phillip Rogaway, "Entity authentication and key distribution," in *13th Annual International Cryptology Conference (CRYPTO 1993)*, pp. 232–249, Santa Barbara, California, USA, August 1993.

[4] S. M. Bellovin and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," in *1992 IEEE Computer Society Symposium on Research in Security and Privacy (IEEE S&P 1992)*, pp. 72–84, Oakland, CA, USA, May 1992.

[5] S. M. Bellovin and M. Merritt, "Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise," in *Proceedings of the 1st ACM Conference on Computer and Communications Security (CCS 1993)*, pp. 244–250, Fairfax, Virginia, USA, November 1993.

[6] H. A. N. Far, M. Bayat, A. K. Das, M. Fotouhi, S. M. Pournaghi, and M. A. Doostari, "LAPTAS: lightweight anonymous privacy-preserving three-factor authentication scheme for wsn-based iiot," *Wireless Networks*, vol. 27, no. 2, pp. 1389–1412, 2021.

[7] N. Fleischhacker, M. Manulis, and A. Azodi, "A modular framework for multi-factor authentication and key exchange," in *Security Standardisation Research - First International Conference (SSR 2014)*, pp. 190–214, London, UK, December 2014.

[8] Y. Gu, S. Jarecki, and H. Krawczyk, "KHAPE: asymmetric PAKE from key-hiding key exchange," in *41st Annual International Cryptology Conference (CRYPTO 2021)*, pp. 701–730, Virtual Event, August 2021.

[9] J. Guo, S. Lu, C. Gu, X. Chen, and F. Wei, "Security analysis and design of authentication key agreement protocol in medical internet of things," in *International Conference on Networking and Network Applications (NaNA 2020)*, pp. 233–240, Haikou City, China, December 2020.

[10] F. Hao and D. Clarke, "Security analysis of a multi-factor authenticated key exchange protocol," in *Applied Cryptography and Network Security - 10th International Conference (ACNS 2012)*, pp. 1–11, Singapore, June 2012.

[11] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu, "Robust multi-factor authentication for fragile communications," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 6, pp. 568–581, 2014.

[12] M. S. Hwang, J. W. Lo, S. C. Lin, "An efficient user identification scheme based on ID-based cryptosystem", *Computer Standards & Interfaces*, vol. 26, no. 6, pp. 565-569, 2004.

[13] M. S. Hwang and W. P. Yang, "Conference key distribution protocols for digital mobile communication systems", *IEEE Journal on Selected Areas in Communications*, vo1. 13, no. 2, pp. 416-420, Feb. 1995.

[14] T. Jager, F. Kohlar, S. Schäge, and J. Schwenk, "Generic compilers for authenticated key exchange," in *16th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2010)*, pp. 232–249, Singapore, December 2010.

[15] S. Jarecki, H. Krawczyk, and J. Xu, "Opaque: An asymmetric pake protocol secure against precomputation attacks," in *37th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2018)*, pp. 456–486, Tel Aviv, Israel, 2018.

[16] C. C. Lee, M. S. Hwang, L. H. Li, "A new key authentication scheme based on discrete logarithms", *Applied Mathematics and Computation*, vol. 139, no. 2, pp. 343-349, July 2003.

[17] C. C. Lee, C. H. Liu, M. S. Hwang, "Guessing attacks on strong-password authentication protocol", *International Journal of Network Security*, vol. 15, no. 1, pp. 64-67, 2013.

[18] Z. Li, Z. Yang, P. Szalachowski, and J. Zhou, "Building low-interactivity multifactor authenticated key exchange for industrial internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 844–859, 2021.

[19] I. C. Lin, C. C. Chang, M. S. Hwang, "Security enhancement for the simple authentication key agreement algorithm", in *Proceedings 24th Annual International Computer Software and Applications Conference (COMPSAC'00)*, 2000.

[20] C. H. Ling, C. C. Lee, C. C. Yang, and M. S. Hwang, "A secure and efficient one-time password authentication scheme for WSN", *International Journal of Network Security*, vol. 19, no. 2, pp. 177-181, Mar. 2017.

[21] W. Liu, X. He, and Z. Ji, "An improved authentication protocol for telecare medical information system," *International Journal of Electronics and Information Engineering*, vol. 12, no. 4, pp. 170–181, 2020.

[22] Y. Liu, F. Wei, and C. Ma, "Multi-factor authenticated key exchange protocol in the three-party setting," in *Information Security and Cryptology - 6th International Conference (Inscrypt 2010)*, pp. 255–267, Shanghai, China, October 2011.

[23] Z. Ma and J. He, "Outsider key compromise impersonation attack on a multi-factor authenticated key exchange protocol," in *Applied Cryptography and Network Security Workshops (ACNS Workshops 2022)*, pp. 320–337, Rome, Italy, June 2022.

[24] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen, and Y. Koucheryavy, "Multi-factor authentication: A survey," *Cryptography*, vol. 2, no. 1, pp. 1–31, 2018.

[25] H. H. Ou, M. S. Hwang and J. K. Jan, "A cocktail protocol with the authentication and key agreement

on the UMTS", *Journal of Systems and Software*, vol. 83, no. 2, pp. 316-325, Feb. 2010.

[26] D. Pointcheval and G. Wang, "VTBPEKE: verifier-based two-basis password exponential key exchange," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (AsiaCCS 2017)*, pp. 301–312, Abu Dhabi, United Arab Emirates, April 2017.

[27] D. Pointcheval and S. Zimmer, "Multi-factor authenticated key exchange," in *Applied Cryptography and Network Security, 6th International Conference (ACNS 2008)*, pp. 277–180, New York, USA, June 2008.

[28] T. Wan, X. Liu, W. Liao, and N. Jiang, "Cryptanalysis and improvement of a biometric-based authentication scheme for multi-server architecture," *International Journal of Network Security*, vol. 22, no. 3, pp. 490–501, 2020.

[29] F. Wang, G. Xu, C. Wang, and J. Peng, "A provably secure biometrics-based authentication scheme for multiserver environment," *Security and Communication Networks*, vol. 2019, no. 4, pp. 1–15, 2019.

[30] F. Wu, X. Li, L. Xu, P. Vijayakumar, and N. Kumar, "A novel three-factor authentication protocol for wireless sensor networks with iot notion," *IEEE System Journal*, vol. 15, no. 1, pp. 1120–1129, 2021.

[31] L. Wu, J. Wang, K. R. Choo, and D. He, "Secure key agreement and key protection for mobile device user authentication," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 319–330, 2019.

[32] T. Y. Wu, L. Yang, Z. Lee, C. M. Chen, J. S. Pan, and S. H. Islam, "Improved ecc-based three-factor multiserver authentication scheme," *Security and Communication Networks*, vol. 2021, no. 1, pp. 1–14, 2021.

[33] Q. Xie, Z. Tang, and K. Chen, "Cryptanalysis and improvement on anonymous three-factor authentication scheme for mobile networks," *Computers & Electrical Engineering*, vol. 59, pp. 218–230, 2017.

[34] Z. Yang, C. Jin, J. Ning, Z. Li, A. Dinh, and J. Zhou, "Group time-based one-time passwords and its application to efficient privacy-preserving proof of location," in *Annual Computer Security Applications Conference (ACSAC 2021)*, pp. 497–512, Virtual Event, USA, December 2021.

[35] Z. Yang and S. Li, "On security analysis of an after-the-fact leakage resilient key exchange protocol," *Information Processing Letters*, vol. 116, no. 1, pp. 33–40, 2016.

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

Biography

Zhiqiang Ma is a master student at School of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China. His main research interests include cryptography and security protocol.

Jun He received the Ph.D. degree from the School of Information Science Technology, East China Normal University, Shanghai, China, in 2010. She is currently a Teacher with the School of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China. Her main research interests include information security and cryptography.