Continuous After-the-Fact Leakage-Resilient Group Password-Authenticated Key Exchange

Ou Ruan, Zihao Wang, Qingping Wang, and Mingwu Zhang
(Corresponding author: Ou Ruan)

School of Computer Science, Hubei University of Technology
No.28, Nanli Road, Hongshan District, Wuhan City, Hubei Province, PR China
(Email: ruanou@163.com)

(Received May 29, 2018; Revised and Accepted Sept. 31, 2018; First Online Aug. 5, 2019)

Abstract

During the past two decades, side-channel attacks have become a familiar method of attacking cryptographic systems, which allow an attacker to learn partial information about the internal secrets such as the secret key. A scheme that is secure in the traditional model will be vulnerable in the leakage environments, thus designing a strong, meaningful, and achievable security scheme to capture the practical leakage attacks is one of the primary goals of leakage-resilient cryptography. In this work, we first formalize a continuous after-the-fact (AF) security model for leakage-resilient (LR) group password-authenticated key exchange (GPAKE) protocol, where the leakages are continuous and are allowed even after the adversary is given the challenges. Then, by combining Diffie-Hellman group key exchange protocol and Dziembowski-Faust leakage-resilient storage scheme appropriately, we propose the first LR GPAKE protocol and present a formal security proof in the standard model.

Keywords: Group Setting; Leakage-Resilience; Password-based Authenticated Key Exchange; Provable Security; Side-channel Attacks

1 Introduction

With the development of the Internet of things, the mobile Internet, the Industrial Internet and the Ad Hoc network, there are more and more group communication applications such as audio or video conferencing, collaborative computing, group chatting, online teaching, and so on. In order to ensure the security of group applications, group authenticated key exchange (GAKE) scheme was proposed, which is used to generate a secure session key in the public networks for all group members. Among GAKE schemes, group password-authenticated key exchange (GPAKE) is most practical because group members could generate a shared secure session key by only using their human-memorable passwords. In 2000, Asokan and Ginzboorg [5] first proposed a GPAKE protocol. Then, many scholars have studied GPAKE protocols [1, 9, 13, 16, 20, 22, 39, 41, 42, 44].

All above GPAKE protocols were secure in the traditional security model that assumed the adversary could not get any information of the secret keys. Recently, many researches showed that an adversary could obtain some information about the secret keys by the side-channel attacks [25, 28]. This kind of attacks can obtain the internal state of the system by observing the physical properties of the devices, such as running time, power consumption, electromagnetic effect, and so on. For example, in the Internet of things, the mobile Internet or the Ad Hoc network, most nodes are very vulnerable to side-channel attacks because they are exploded in the public environments. Thus, traditional GPAKEs are completely insecure in the leakage environments. Then, it is very necessary to model and construct the leakage-resilient (LR) GPAKE protocols. However, there is no previous work for standardizing the security models and designing the LR GPAKE protocols. In this paper, we propose a continuous after-the-fact (AF) LR (λ-CAFLR) security model for GPAKE protocol. In this model the users can periodically refresh the secret using some additional fresh local randomness. The adversary can attack the system for arbitrarily many instances, where, in each instance, he can adaptively learn up to λ bits of arbitrary information about the current witness for some leakage parameter λ. The secret is then refreshed for the next instance. Nevertheless, after attacking the system for any polynomial number of instances, the attacker still cannot produce a valid witness. Notice that, there is a necessary bound on the amount of leakage in each instance and the overall amount of leakage during the attack is unbounded. Then, we present a LR GPAKE protocol based on Diffie-Hellman (DH) group key exchange protocol [10], key derivation function (KDF) [29], leakage-resilient storage (LRS) [18] and leakage-resilient refreshing of LRS. At last, we show a formal security proof in the standard model based on the new λ-CAFLR security model.

The main contributions are shown as follows:

- First, we first define a λ-CAFLR eCK security model for GPAKE by extending the eCK security PAKE model properly. In the model, the leakages are continuous and are allowed even after the adversary selects the test session, and the whole leakage size may be infinitely large, and for each protocol instance the amount of leakage is bounded by λ.
- Second, we propose the first LR GPAKE protocol by combining DH GKE protocol and Dziembowski-Faust (DF) LRS (DF-LRS) scheme appropriately.
- Third, we formally prove the CAFLR eCK security in the standard model based on the game simulation techniques.

Our paper is organized as follows. In Section 2, we review related works. In Section 3, we present the used cryptography tools. In Section 4, we define the CAFLR security model for GPAKE protocol. In Section 5, we describe the new protocol and its provable security. Finally, In Section 6, we show the conclusion of the paper.

2 Related Works

2.1 Traditional GAKE

GAKE protocols allow a group of parties communicating over a public network to come up with a common secret session key. Due to their critical role in building secure multicast channels, a number of GAKE protocols have been suggested over the years for a variety of settings. The first pioneering work for GAKE is the Ingemarsson *et al.* [26]. Their protocol was a natural extension of DH key exchange protocol [19]. The protocol required a synchronous startup and $(n-1)$ rounds communications. In 1994, Burmester and Desmedt (BD) [10] proposed a much efficient GAKE protocol with only two rounds communications. In 1996, Steiner *et al.* [36] showed that BD protocol was insecure even under the passive attacks, and then presented a more practical protocol and gave a formal security proof. But, their protocol was only secure against the passive attacks. In order to resist the active attacks, Bresson *et al.* [8] first introduced a formal security model for GAKE and showed the first provably secure protocol in this model. Their protocol required $\mathrm{O}(n)$ rounds to establish a secure shared group key among n users, and therefore was not scalable. Boyd *et al.* [7] presented a much efficient constant-round GAKE protocol with a security proof in the random oracle (RO) model. But it was also not scalable. In 2003, Katz *et al.* [27] first showed a scalable GAKE protocol with a formal security proof in the standard model under the Decision Diffie-Hellman (DDH) assumption, where users are allowed to securely join and leave the group at any time. Recently, Teng *et al.* [38] proposed a scalable GAKE protocol for wireless mobile networks; Halford *et al.* presented the energy-efficient GAKE protocols for Ad Hoc networks [23] and wireless networks [24], which aimed to increase the energy-efficiency of GAKE and were secure in the information-theoretic model against out-of-network eavesdroppers.

2.2 Traditional GPAKE

A password is the ideal authentication means to exchange a session key in the absence of public-key infrastructures or pre-distributed symmetric keys. In a group, the sharing of a password among the members greatly simplifies the setup of distributed applications. Therefore, in this way the GPAKE was introduced. In 2000, Asokan and Ginzboorg [5] proposed the first GPAKE protocol, but they didn't gave the formal security proofs. In 2002, Bresson et al. [9] proposed the first provably secure GPAKE protocol in the RO model under the DDH assumption. These two protocols required $\mathrm{O}(n)$ rounds communications and $\mathrm{O}(n)$ exponentiations per each user, where n is the number of group users. In 2006, Dutta *et al.* [20] presented much efficient GPAKE protocol with only two rounds communications. Later, Abdalla *et al.* [1] showed that the protocol [20] was vulnerable to the off-line dictionary attack, and proposed a GPAKE protocol with constant-round communications that was secure against the off-line dictionary attack. All above protocols were not scalable. In 2009, Wu *et al.* [39] presented an efficient scalable GPAKE protocol with a formal security proof. Recently, Zhou *et al.* [42] designed a cross-realm GPAKE protocol; Dai *et al.* [16] showed cross-realm GPAKE protocols using different passwords; Zhu *et al.* [44] presented a novel cross-domain GPAKE protocol with explicit authentication and contributiveness in the universally composable (UC) framework.

2.3 LR Authenticated Key Exchange

The last decade, there were lots of researches [6, 17, 31, 37, 40, 43] focusing on the LR cryptography that aims to provide secure solutions for the leakage environments. Authenticated key exchange (AKE) protocols allow two parties communicating over an insecure network to establish a common secret key. They are among the most widely used cryptographic protocols in practice. In order to resist key-leakage attacks, several LR AKE protocols have been proposed recently in the leakage model. The first LR security model for AKEs was introduced by Moriyama and Okamoto (MO) [32] in 2011. The central limitation of the MO model is that the leakages are only allowed until the adversary learns the challenge. Leakage that occurs after the adversary learns the challenge is called after-the-fact (AF) leakage. In 2014, Alawatugoda *et al.* [2] first presented an AFLR security model and a continuous AFLR (CAFLR) AKE protocol. Their security model was based on the CK security model [11] where the adversary can access only the long-term secret key. Alawatugoda *et al.* [3] gave the first AFLR eCK security model [30] where the adversary can access both the long-term secret key and the ephemeral secret randomness, and proposed the first bounded AFLR (BAFLR) eCK-secure AKE protocol. Then, Alawatugoda *et al.* [4] showed the first CAFLR eCK-secure AKE protocol. In 2016, Chen *et al.* [14, 15] first introduced a strong security model for AKEs that considered leakage attacks on both the long-term secret private key and the ephemeral secret randomness. Then, they proposed a BAFLR eCK-secure AKE protocol under this new model. In 2017, the first ID-based BAFLR AKE protocol was introduced by Ruan *et al.* [35]. Recently, Ruan *et al.* [33] first presented an LR eCK security model for PAKE and constructed an LR PAKE protocol; Ruan *et al.* [34] first define an LR eCK-security model for 3PAKE and propose an LR 3PAKE protocol. Chakraborty *et al.* [12] first proposed an LR non-interactive key exchange in continuous-memory leakage model, which could be used as a building block to construct LR PKE schemes, interactive key exchange and low-latency key exchange protocols.

3 Preliminaries

In this section, we describe the used primitives, such as PDDH assumption, KDF, LRS and leakage-resilient refreshing of LRS.

3.1 Notation

Let s $\xleftarrow{\$} \Omega$ denote that s is picked uniformly from a finite set Ω at random.

Definition 1 (Negligible function)**.** *A negligible function $\varepsilon(k)$ means for each positive integer $c \geq 0$ there exists an integer k_c that $\varepsilon(k) < k^{-c}$ holds for each $k \geq k_c$.*

Definition 2 (Parallel decision diffie-hellman (PDDH) Assumption)**.** *PDDH assumption is a variant of the DDH assumption. A distinguishing game is used to formally define PDDH assumption:*

1) *A challenger **C** generates (G, g) and sends them to an adversary **A**, where G is a cyclic multiplicative group with a large prime order p and g is a random generator of G.*

2) ***C** randomly chooses $x_1, \cdots, x_n, y_1, \cdots, y_n \xleftarrow{\$} Z_p^*$ and $b \xleftarrow{\$} (0,1)$. If $b = 1$, **C** sends $(g^{x_1}, \cdots, g^{x_n}, g^{x_1 x_2}, \cdots, g^{x_n x_1})$ to **A**, else **A** is given $(g^{x_1}, \cdots, g^{x_n}, g^{y_1}, \cdots, g^{y_n})$.*

3) ***A** outputs his guessed bit b', and **A** wins if $b' = b$.*

PDDH assumption means that:

$$Adv_{PDDH}(A) = |\Pr[b' = b] - 1/2| = \varepsilon(\cdot),$$

*where $Adv_{PDDH}(A)$ represents the advantage that **A** wins the above game and $\varepsilon(\cdot)$ is a negligible function.*

Definition 3 (λ-Leakage-resilient storage)**.** *A λ-LRS includes two probabilistic polynomial time (PPT) algorithms (**Encode**, **Decode**) and a bounded leakage parameter $\lambda = (\lambda_1, \lambda_2)$.*

Encode: $Encode(s) = s_L \times s_R$, where s is an element chosen from the message space M, $s_L \times s_R$ is the encoded output element in the encoding space $L \times R$.

Decode: $Decode(s_L \times s_R) = s$.

A LRS must satisfy the following two properties:

1) Correctness of LRS. For each $s \xleftarrow{\$} M$, there has $Decode(Encode(s)) = s$.

2) Security of LRS. A distinguishing game is shown as follows:

   a. An adversary ***A*** picks two elements $(s_0, s_1) \xleftarrow{\$} M$ at random and sends (s_0, s_1) to a challenger ***C***.

   b. ***C*** randomly selects a bit $b \xleftarrow{\$} (0,1)$ and generates $Encode(s_b) = (s_b)_L \times (s_b)_R$.

   c. For each round $i = 1, \cdots, t$, ***A*** selects leakage functions $f = (f_i^L, f_i^R)$ and get the leakage $(f_i^L((s_b)_L), f_i^R((s_b)_R)$ back from ***C***, where the total leakage size should be bounded by (λ_1, λ_2), *i.e.*, $\sum_1^t f_i^L((s_b)_L) \leq \lambda_1 \wedge \sum_1^t f_i^R((s_b)_R) \leq \lambda_2$.

   d. ***A*** outputs his guessed bit b', and ***A*** wins if $b' = b$.

The security of LRS means that

$$Adv_{LRS}(A) = \varepsilon(\cdot),$$

where $Adv_{LRS}(A)$ denotes the advantage of ***A*** in winning the above game and $\varepsilon(\cdot)$ is a negligible function.

Definition 4 (($\lambda_{Refresh}, \lambda$)-Leakage-resilient refreshing of LRS)**.** *A leakage-resilient refreshing is a PPT algorithm **Refresh** with λ-LRS (**Encode**, **Decode**), a secret sand a bounded leakage amount $\lambda_{Refresh} = (\lambda_{Refresh1}, \lambda_{Refresh2})$.*

Refresh: $Refresh(s_L \times s_R) = s'_L \times s'_R$ where $s_L \times s_R$ is the encoding value of the secret s.

A leakage-resilient refreshing of LRS should satisfy the following two properties:

1) Correctness of leakage-resilient refreshing. For each $s \xleftarrow{\$} M$, there has

$$Decode(s'_L \times s'_R) = Decode(s_L \times s_R).$$

2) $(\lambda_{Refresh}, \lambda)-$Security of leakage-resilient refreshing. A distinguishing game is shown as follows:

   a. An adversary ***A*** picks two elements $(s_0, s_1) \xleftarrow{\$} M$ at random and sends (s_0, s_1) to a challenger ***C***.

   b. ***C*** randomly selects a bit $b \xleftarrow{\$} (0,1)$ and generates $Encode(s_b) = (s_b)_L^0 \times (s_b)_R^0$.

c. For each $i = 1, \cdots, \ell$, $\boldsymbol{A}$ selects the i^{th} round leakage functions $f_{Refresh\text{-}i} = (f^L_{Refresh\text{-}i}, f^R_{Refresh\text{-}i})$ and gets back the leakages $(f^L_{Refresh\text{-}i}((s_b)^i_L), f^R_{Refresh\text{-}i}((s_b)^i_R))$ from $\boldsymbol{C}$, where $f^L_{Refresh\text{-}i}((s_b)^i_L) \leq \lambda_{Refresh1} \wedge f^R_{Refresh\text{-}i}((s_b)^i_R) \leq \lambda_{Refresh2}$; then, $\boldsymbol{C}$ refreshes the encodings,

$$Refresh((s_b)^{i-1}_L \times (s_b)^{i-1}_R) = (s_b)^i_L \times (s_b)^i_R.$$

d. $\boldsymbol{A}$ outputs his guessed bit b', and $\boldsymbol{A}$ wins if $b' = b$.

The $(\lambda_{Refresh}, \lambda)$-security of leakage-resilient refreshing means that:

$$Adv_{Refresh-LRS}(A) = \varepsilon(\cdot),$$

where $Adv_{Refresh-LRS}(A)$ denotes the advantage of $\boldsymbol{A}$ in winning the above game and $\varepsilon(\cdot)$ is a negligible function.

Definition 5 (Dziembowski-faust(DF) LRS scheme)**.** *Suppose $s \in (Z^*_p)^m$ is a secret value with any $n \in N$.*

Encode *: Choose a random $s_L \xleftarrow{\$} (Z^*_p)^n \backslash \{(0^n)\}$ and generate $s_R \in (Z^*_p)^{n \times m}$ such that $s_L \times s_R = s$, where $n \in$ N. Output (s_L, s_R).*

Decode *: $Decode(s_L \times s_R) = s$.*

Lemma 1. *[21]. If $m < n/20$, Definition 5 is a λ-secure LRS scheme with $\lambda = (0.3 \cdot n \cdot \log p, 0.3 \cdot n \cdot \log p)$, named $\Phi^{n,m}_{Z^*_p}$.*

Lemma 2. *[21]. If $m/3 \leq n \wedge n \geq 16$, there has a $(\lambda/2, \lambda)$-secure leakage-resilient refreshing $Refresh^{n,m}_{Z^*_p}$ for $\Phi^{n,m}_{Z^*_p}$, where $\Phi^{n,m}_{Z^*_p}$ is a λ-secure DF-LRS.*

Definition 6 (Key derivation function)**.** *KDF is a PPT algorithm that is used to compute a secret key with inputs (σ, ℓ, r, c),* i.e., *k = KDF(σ, ℓ, r, c), where σ denotes the source material of k, ℓ is some public knowledge about σ such as its length, r is a salt value and c represents a context variable.*

Security of KDF. A distinguishing game is defined as follows:

1) *The challenger $\boldsymbol{C}$ chooses (σ, ℓ) and sends them to an adversary$\boldsymbol{A}$.*

2) *$\boldsymbol{A}$ randomly selects a value c and a salt value r.*

3) *$\boldsymbol{C}$ picks a random bit $b \xleftarrow{\$} (0, 1)$. If $b = 1$, $\boldsymbol{C}$ calculates k = KDF(σ, ℓ, r, c), else $\boldsymbol{C}$ picks a string s at random, and then give it to $\boldsymbol{A}$, where the length of s and k is equal.*

4) *$\boldsymbol{A}$ outputs his guessed bit b', and $\boldsymbol{A}$ wins if $b' = b$.*

The security of KDF means that:

$$Adv_{KDF}(A) = \varepsilon(\cdot),$$

where Adv_{KDF}(A) denotes the advantage of $\boldsymbol{A}$ in winning the above game and $\varepsilon(\cdot)$ is a negligible function.

4 The CAFLR Security Model For GPAKE Protocol

This section formally defines the λ-CAFLR security model for GPAKE protocol. The new model follows the only computation leakage (OCL) model, which assumes that leakage only occurs in the calculations associated with the secret password. In the λ-CAFLR security model an adversary $\boldsymbol{A}$ could continuously get arbitrarily leakages of the secret password, but for each instantiation of the protocol the amount of leakage is bounded by λ. In each instantiation, $\boldsymbol{A}$ could adaptively select any PPT leakage functions $f = (f_1, \cdots, f_n)$ to obtain leakage of the long-term secret password pw, and the overall amount of leakages is bounded by λ, *i.e.*, $\sum |f_i(pw)| \leq \lambda$. After receiving a leakage function f_i chosen by $\boldsymbol{A}$, $\boldsymbol{A}$ will be given the leakage $f_i(pw)$.

4.1 System Framework

The typical system model of GPAKE protocols is shown in Figure 1, in which a group of parties $U_1, \cdots, U_n, n = poly(\kappa)$ share a short common human-memory password pw and seek to generate a shared and secure group session key k.

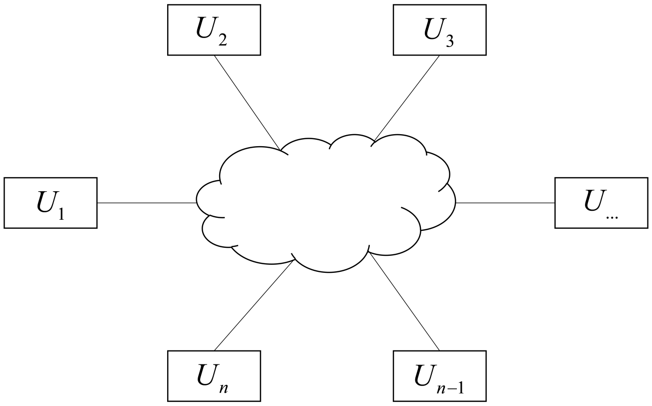

Figure 1: System model

Notations in the system framework:

Principal: Is a party involved into a protocol instance.

Session: Represent a protocol instance with principals.

Oracle $\Pi^t_{U_i}$: Is the principal U_i in the t^{th} session.

Session ID: Each protocol instance at a party is identified by a unique session ID. The session ID of $\Pi^t_{U_i}$ is denoted by $sid^t_{U_i}$.

Partner ID: The partner ID $pid^t_{U_i}$ of $\Pi^t_{U_i}$, is a set of identities of the principals with whom U_i wishes to establish a common group key, *i.e.*, $pid^t_{U_i} = \{\Pi^t_{U_1}, \cdots, \Pi^t_{U_n}\}$. Note that it includes the identity of U_i itself.

4.2 Adversarial Powers

Adversarial powers are modelled by the following queries:

Send $(\Pi^t_{U_i}, m, f)$ **query:** Upon receiving ***Send*** query with a message m and a leakage function f, $\Pi^t_{U_i}$ of the t^{th} session will generate a normal protocol message based on the protocol specifications and the leakage $f(pw)$ of the long-term password, and send them to the adversary $\boldsymbol{A}$. $\boldsymbol{A}$ can activate a new protocol instance by asking ***Send*** $(\Pi^t_{U_1}, \text{(start)}, (\ ))$ to the initiator principal.

RevealSessionKey$(\Pi^t_{U_i})$ **query:** $\Pi^t_{U_i}$ gives the session key of the t^{th} session to $\boldsymbol{A}$.

RevealEphemeralKey$(\Pi^t_{U_i})$ **query:** $\Pi^t_{U_i}$ gives his random ephemeral key of the t^{th} session to $\boldsymbol{A}$.

Corrupt() query: Any oracle gives his secret password pw to $\boldsymbol{A}$.

Test$(\Pi^t_{U_i})$ **query:** Upon receiving a ***Test*** query, the challenger randomly chooses a bit $b \xleftarrow{\$} (0,1)$, if b = 1 then $\boldsymbol{A}$ is given the actual session key, while a random key is given to $\boldsymbol{A}$.

4.3 λ−CAFLR Security Model

In the λ-CAFLR security model, the total leakage amount of the secret password are bounded by the parameter λ, *i.e.*, $\sum |f_i(pw)| \leq \lambda$.

Definition 7 (Partners in CAFLR eCK security model)**.** *Two oracles $\Pi^t_{U_i}$ and $\Pi^{t'}_{U_j}$ are called partners if the followings satisfy:*

1) *Two oracles $\Pi^t_{U_i}$ and $\Pi^{t'}_{U_j}$ have produced a common group session key;*

2) $sid^t_{U_i} = sid^{t'}_{U_j}$*;*

3) $pid^t_{U_i} = pid^{t'}_{U_j}$*;*

Definition 8 (λ-CAFLR-freshness)**.** *Assume $f = (f_1, \cdots, f_n)$ be n arbitrary PPT leakage functions for an instantiation of the protocol selected by the adversary $\boldsymbol{A}$. An oracle $\Pi^t_{U_i}$ is λ-CAFLR-fresh if the followings satisfy:*

1) *The oracle $\Pi^t_{U_i}$ or any of its partners has not been queried a **RevealSessionKey**.*

2) *If the partners exists, $\boldsymbol{A}$ could not query any of the following combinations:*

   a. ***Corrupt**() and **RevealEphemeralKey**() to any principal.*

   b. ***RevealEphemeralKey**() to all principals and **Corrupt**().*

3) *If none of its partners exist, $\boldsymbol{A}$ could not queried **Corrupt** ().*

4) *For all **Send** $(\cdots, U_i, \cdots, f_i, \cdots)$ queries to any principal U_i, $\sum |f_i(pw)| \leq \lambda$.*

Definition 9 (λ-CAFLR security game)**.** *$\lambda-$CAFLR security game is as follows:*

1) *An adversary $\boldsymbol{A}$ asks any of **Send**, **RevealSessionKey**, **RevealEphemeralKey** and **Corrupt** to any oracle as he wants.*

2) *$\boldsymbol{A}$ chooses a λ-CAFLR-fresh oracle and asks a **Test** query. Upon getting a **Test** query, the challenger $\boldsymbol{C}$ randomly selects a bit $b \xleftarrow{\$} (0,1)$, if $b = 1$ then $\boldsymbol{A}$ is given the actual session key, while a random key is given to $\boldsymbol{A}$.*

3) *$\boldsymbol{A}$ continues asking **Send**, **RevealSessionKey**, **RevealEphemeralKey** and **Corrupt**. All these queries should not violate the λ-CAFLR-freshness of the test oracle.*

4) *$\boldsymbol{A}$ outputs his guessed bit b', and $\boldsymbol{A}$ wins if $b' = b$.*

Definition 10 (λ-CAFLR security)**.** *$\lambda-$CAFLR security means that:*

$$Adv^{\lambda-CAFLR}_{GPAKE} = |\Pr[b' = b] - 1/2| = N_S/N + \varepsilon(\cdot),$$

where $Adv^{\lambda-CAFLR}_{GPAKE}$ represents the advantage that $\boldsymbol{A}$ wins λ-CAFLR security game in Definition 9, N_S is the number of sessions on a client principal, N denotes the size of the password dictionary that is shared by all client, and $\varepsilon(\cdot)$ is a negligible function.

In GPAKE protocols, the on-line dictionary attack is unavoidable, and N_S/N is the success probability of the on-line dictionary attack. Thus, a λ-CAFLR secure GPAKE protocol means that there hasn't any PPT adversary that could win the above game with an advantage more than N_S/N. There are many ways to limit the on-line dictionary attack, one of the most common method is using a policy that blocks using a password if failed attempts have happened several times.

5 A New λ–CAFLR GPAKE Secure Protocol

5.1 The Proposed Protocol

Let $U_1, \cdots, U_n, n = poly(\kappa)$, be a group of parties that want to generate a group key.

Figure 2 shows the proposed protocol, which includes the following two stages:

The Initial Setup stage:

Each party U_i maps the password pw to an element s of the group G and runs a λ-secure DF-LRS scheme $\Phi^{n,1}_{Z^*_p}$, picks $(u_i)^0_L \xleftarrow{\$} (Z^*_p)^n \backslash \{(0^n)\}$ at random and generates $(u_i)^0_{\mathrm{R}} \in (Z^*_p)^{n\times 1}$, such that $(u_i)^0_L \cdot (u_i)^0_R = s$. We suppose that these calculations are secretly computed and there hasn't any leakage attack.

	User U$_i$
Initial setup stage:	s=H(pw), $(u_i)^0_L \xleftarrow{\$} (Z_p^*)^n\backslash\{(0^n)\}$, *computes* $(u_i)^0_R \in (Z_p^*)^{n\times 1}$ *such* that $(u_i)^0_L \cdot (u_i)^0_R$=s
Protocol Execution stage:	$r_i \xleftarrow{\$} Z_p^*, z_i = g^{r_i} \quad \xrightarrow{(U_i,z_i,t_i)}$ $t_i = g^{(u_i)^j_R}$ $X_i = (z_{i+1}/z_{i-1})^{r_i} \cdot (t_i)^{(u_i)^j_L} \quad \xrightarrow{(U_i,X_i)}$ $Y_i = (t_i)^{(u_i)^j_L}$ $K_i = (z_{i-1})^{nr_i} \cdot (X_i/Y_i)^{n-1} \cdot (X_{i+1}/Y_i)^{n-2} \cdot \cdots \cdot (X_{i-3}/Y_i)^2 \cdot (X_{i-2}/Y_i)^1$, $k_G = KDF(U_1\Vert \cdots \Vert U_n, Y_i, K_i)$ $((u_i)^{j+1}_L, (u_i)^{j+1}_R) \leftarrow \text{Refresh}^{n,1}_{Z_p^*}((u_i)^j_L, (u_i)^j_R)$

Figure 2: The LR PGAKE Protocol

The Protocol Execution stage:

Round 1. Each party $U_i, i = 1, \cdots, n$, chooses a random $r_i \in_R Z_q$, computes $z_i = g^{r_i} \bmod q$ and $t_i = g^{(u_i)^j_R}$, and broadcasts $(U_i,\ z_i,\ t_i)$.

Round 2. Each party $U_i, i = 1, \cdots, n$, computes $X_i = (z_{i+1}/z_{i-1})^{r_i} \cdot (t_i)^{(u_i)^j_L} \bmod q$ and broadcasts it, where the indices are taken in a cycle.

Key Computation: Each party $U_i, i = 1, \cdots, n$, computes

$$\begin{aligned} Y_i &= (t_i)^{(u_i)^j_L} \\ K_i &= (z_{i-1})^{nr_i} \cdot (X_i/Y_i)^{n-1} \cdot (X_{i+1}/Y_i)^{n-2} \cdots \\ &\quad \cdot (X_{i-3}/Y_i)^2 \cdot (X_{i-2}/Y_i)^1 \\ k_G &= KDF(U_1 || \cdots || U_n, Y_i, K_i) \end{aligned}$$

then refreshes the store pieces with

$$((u_i)^{j+1}_L, (u_i)^{j+1}_R) \leftarrow \text{Re fresh}^{n,1}_{Z_p^*}((u_i)^j_L, (u_i)^j_R).$$

Correctness of the proposed protocol.

First:

$$\begin{aligned} Y_i &= (t_i)^{(u_i)^j_L} = (g^{(u_i)^j_R})^{(u_i)^j_L} = g^s \\ X_i &= (z_{i+1}/z_{i-1})^{r_i} \cdot (t_i)^{(u_i)^j_L} \\ &= (z_{i+1}/z_{i-1})^{r_i} \cdot (g^{(u_i)^j_R})^{(u_i)^j_L} \\ &= (z_{i+1}/z_{i-1})^{r_i} \cdot g^s \end{aligned}$$

Second,

$$\begin{aligned} A_{i-1} &= (z_{i-1})^{r_i} = g^{r_{i-1}r_i} \\ A_i &= (z_{i-1})^{r_i} \cdot (X_i/Y_i) \\ &= (z_{i-1})^{r_i} \cdot ((z_{i+1}/z_{i-1})^{r_i} \cdot g^s/g^s) = g^{r_ir_{i+1}} \\ A_{i+1} &= (z_{i-1})^{r_i} \cdot (X_i/Y_i) \cdot (X_{i+1}/Y_i) = g^{r_{i+1}r_{i+2}} \\ &\cdots \\ K_i &= (z_{i-1})^{nr_i} \cdot (X_i/Y_i)^{n-1} \cdot (X_{i+1}/Y_i)^{n-2} \cdots \\ &\quad \cdot (X_{i-3}/Y_i)^2 \cdot (X_{i-2}/Y_i)^1 \\ &= A_{i-1} \cdot A_i \cdot A_{i+1} \cdot \cdots \cdot A_{i-2} \\ &= g^{r_1r_2+r_2r_3+\cdots+r_nr_1} \end{aligned}$$

Thus, the proposed protocol is correct.

5.2 Security Proof

Theorem 1. *If the leakage-resilient refreshing of LRS is $(\lambda, 2\lambda)$-secure, PDDH assumption is hold, and KDF is secure, the new GPAKE protocol is λ-CAFLR eCK-secure,* i.e., $Adv^{\lambda-CAFLR}_{GPAKE} \leq N_S/N + \frac{1}{(c^n_{N_P} \cdot c^2_{N_S})}(Adv_{Refresh-LRS} + Adv_{KDF} + Adv_{PDDH})$, *where $Adv^{\lambda-CAFLR}_{GPAKE}$ denotes the advantage of an adversary $\boldsymbol{A}$ in winning the λ-CAFLR security game of the proposed protocol, $Adv_{PDDH}, Adv_{KDF}, Adv_{Refresh-LRS}$ represent advantages of $\boldsymbol{A}$ in winning the security game of PDDH, KDF and leakage-resilient refreshing of LRS, respectively, and N_P is the number of protocol principals, N_S denotes the number of sessions on a principal, N is the password dictionary's size, $c^n_{N_P}$ is the number of choosing n elements from a set of N_P elements.*

Our formal proof is based on the game hopping technique. First, we give a sequence of games, in which Game 1 is the original λ-CAFLR security game and the advantages of the last Game is negligible; Second, we show that each game is not distinguished from its previous game. Thus, we get that the advantages of the original λ-CAFLR security game is negligible.

Proof. The proof could be divided into two main cases: (1) a partner to the test oracle exists, and (2) it does not exist.

Case 1. A partner to the test oracle exists.

In this case, the adversary $\boldsymbol{A}$ is a passive adversary who only collect the protocol messages. We split its proofs into two sub cases as follows:

1) $\boldsymbol{A}$ asks ***corrupt()*** query. In this case, $\boldsymbol{A}$ could get the long-term group password *pw*.

2) $\boldsymbol{A}$ does not ask ***corrupt()*** query. In this case, $\boldsymbol{A}$ could not get the long-term group password *pw*.

Case 1.1. *A* asks *corrupt*() query.

In this case, leakage attacks don't need to consider because ***A*** could get the long-term group password *pw* by ***corrupt*()** query and map it to the element s of the group G. However, ***A*** could not query ***RevealEphemeralKey*()** to any oracle in order not to violate λ-CAFLR-freshness of ***Test*** oracle.

Game 1: This is the original λ-CAFLR security game.

Game 2: Game 2 and Game 1 only have the following differences: ***A*** selects a group n different client principals $\{\mathrm{U}_1, \cdots, \mathrm{U}_n\} \xleftarrow{\$} \{\mathrm{u}_1, \cdots, \mathrm{u}_{N_p}\}$ and two numbers $t^*, r^* \xleftarrow{\$} \{1, \cdots, N_s\}$ at random, Then, ***A*** begin to activate Game 2 and chooses the oracle $\Pi^{t^*}_{\mathrm{U}_i} (i \in \{1, \cdots, \mathrm{n}\})$ as the target oracle and $\Pi^{r^*}_{U_j} (i \neq j)$ as the partner oracles. If the test oracle is not $\Pi^{t^*}_{\mathrm{U}_i}$ or the partner oracles are not $\Pi^{r^*}_{U_j}$, Game 2 challenger ***C*** exists and terminates Game 2.

Game 3: Game 3 and Game 2 only have the following differences: ***C*** calculates $k_G = KDF(U_1||\cdots||U_n, g^s, g^{r'_1+\cdots+r'_n})$ where $r'_1, \cdots, r'_n \xleftarrow{\$} Z^*_p$. Then, upon receiving a ***Test***$(\Pi^{t^*}_{\mathrm{U}_i})$ or ***Test***$(\Pi^{r^*}_{U_j})$ query, ***C*** gives k_G to ***A***.

Game 4: Game 4 and Game 3 only have the following differences: ***C*** selects a random key $k_G \xleftarrow{\$} \{0,1\}^k$. Then, upon getting a ***Test***$(\Pi^{t^*}_{\mathrm{U}_i})$ or ***Test***$(\Pi^{r^*}_{U_j})$ query, ***C*** gives k_G to ***A***.

Differences between games: The followings show that each game t is not distinguished from its previous game t-1. Let $\mathrm{Adv}_{Game\ t}(A)$ be the advantage that ***A*** wins Game t.

Game 1: In the original game, there has

$$\mathrm{Adv}_{Game\ 1}(A) = Adv^{\lambda-CAFLR}_{\mathrm{GPAKE}} \quad (1)$$

Game 1 and Game 2: If the test oracle is $\Pi^{t^*}_{\mathrm{U}_i}$ and the partner oracles are $\Pi^{r^*}_{U_j} (i \neq j)$, Game 2 is consistent with Game 1. The probability that ***A*** correctly selects a test session and a partner is $1/(c^n_{N_P} \cdot c^2_{N_S})$. Therefore,

$$\mathrm{Adv}_{Game\ 2}(A) = \frac{1}{(c^n_{N_P} \cdot c^2_{N_S})} \mathrm{Adv}_{Game\ 1}(A) \quad (2)$$

Game 2 and Game 3: In Game 2 $k_G = KDF(U_1||\cdots||U_n, g^s, g^{r_1r_2+r_2r_3+\cdots+r_nr_1})$, while in Game 3 $k_G = KDF(U_1||\cdots||U_n, g^s, g^{r'_1+\cdots+r'_n})$. From PDDH assumption, there has

$$|Adv_{Game\ 2}(A) - \mathrm{Adv}_{Game\ 3}(A)| \leq Adv_{PDDH} \quad (3)$$

Game 3 and Game 4: In Game 3 $k_G = KDF(U_1||\cdots||U_n, g^s, g^{r'_1+\cdots+r'_n})$, while $k_G \xleftarrow{\$} \{0,1\}^k$ in Game 4. Because KDF is secure, there has

$$|Adv_{Game\ 3}(A) - \mathrm{Adv}_{Game\ 4}(A)| \leq Adv_{KDF} \quad (4)$$

Game 4: In Game 4, the session key k_G is a random string that doesn't depends on any information. Therefore,

$$\mathrm{Adv}_{Game\ 4}(A) = 0 \quad (5)$$

Using Equations (1)-(5) we get,

$$Adv^{\lambda-CAFLR}_{\mathrm{GPAKE}} \leq \frac{1}{(c^n_{N_P} \cdot c^2_{N_S})}(Adv_{PDDH} + Adv_{KDF}).$$

Case 1.2. *A* does not ask *corrupt*() query.

In this case, ***A*** could get all the random keys $r_1, \cdots, r_n$ by ***RevealEphemeralKey*()**.

Game 1: It is the original game.

Game 2: Consistent with Game 2 in Case 1.1.

Game 3: Game 3 and Game 2 only have the following differences: ***C*** picks $s' \xleftarrow{\$} Z^*_p$ and encodes $((U_i)^0_L, (U_i)^0_R) = Encode(s')$, and continues refreshing the two encodings, then uses them to simulate the answers to ***A***'s leakage function.

Game 4: Game 4 and Game 3 only have the following differences: ***C*** generates

$$k_G = KDF(U_1||\cdots||U_n, g^{t'}, g^{r_1r_2+r_2r_3+\cdots+r_nr_1})$$

where $t' \xleftarrow{\$} Z^*_p$. Upon receiving a ***Test***$(\Pi^{t^*}_{\mathrm{U}_i})$ or ***Test***$(\Pi^{r^*}_{U_j})$ query, ***C*** gives k_G to ***A***.

Game 5: Consistent with Game 4 in Case 1.1.

Differences between games:

Game 1:

$$\mathrm{Adv}_{Game\ 1}(A) = Adv^{\lambda-CAFLR}_{\mathrm{GPAKE}} \quad (6)$$

Game 1 and Game 2: From Game 1 and Game 2 in Case 1.1., we get,

$$\mathrm{Adv}_{Game\ 2}(A) = \frac{1}{(c^n_{N_P} \cdot c^2_{N_S})} \mathrm{Adv}_{Game\ 1}(A) \quad (7)$$

Game 2 and Game 3: In Game 2 the leakage of the shared password is the real leakage of $s = H(\text{pw})$, while the leakage is a leakage of a random value s' in Game 3. Assume $\boldsymbol{A}$ will output a bit b to distinguish between Game 2 and Game 3, $b = 1$ if running Game 2 and otherwise $b = 0$. We design an algorithm $\boldsymbol{B}$ against the leakage-resilient refreshing security distinguishing game, which uses $\boldsymbol{A}$ as a subroutine and runs as following: (1) upon receiving s or $s' \xleftarrow{\$} Z_p^*$ from the leakage-resilient refreshing challenger, $\boldsymbol{B}$ transfers it to $\boldsymbol{A}$'s challenger $\boldsymbol{C}$. $\boldsymbol{C}$ uses it as the mapping group element of the shared secret password, encodes it and continues refreshing two encodings, then uses these encodings to simulate the answers to $\boldsymbol{A}$'s ***Send*** queries with $f_{Refresh} = (f^L_{Refresh}, f^R_{Refresh})$ of the principal U_i. If the received message is s in the first step, the simulation is same as Game 2, otherwise it's same as Game 3. (2) $\boldsymbol{B}$ outputs the bit that $\boldsymbol{A}$ outputs.

If $\boldsymbol{A}$ could distinguish between Game 2 and Game 3, $\boldsymbol{B}$ wins the leakage-resilient refreshing security distinguishing game. Therefore,

$$\begin{aligned} & |Adv_{Game\ 2}(A) - Adv_{Game\ 3}(A)| \\ \leq\ & Adv_{Refresh-LRS}. \end{aligned} \qquad (8)$$

Game 3 and Game 4:

In Game 3 $k_G = KDF(U_1 \parallel \cdots \parallel U_n,\ g^s,\ g^{r_1r_2+r_2r_3+\cdots+r_nr_1})$, while $k_G = KDF(U_1 \parallel \cdots \parallel U_n,\ g^{t'},\ g^{r_1r_2+r_2r_3+\cdots+r_nr_1})$ in Game 4. Because t' is chosen at random and independent on s, g^s and $g^{t'}$ are perfectly indistinguishable. Therefore,

$$|Adv_{Game\ 3}(A) - Adv_{Game\ 4}(A)| = 0. \quad (9)$$

Game 4 and Game 5: From Game 3 and Game 4 in Case 1.1., we get,

$$|Adv_{Game\ 4}(A) - Adv_{Game\ 5}(A)| \leq Adv_{KDF}. \quad (10)$$

Game 5: In Game 5, the leakage is computed using a random value s', and the session key k_G is picked at random. Therefore,

$$Adv_{Game\ 5}(A) = 0 \quad (11)$$

Using Equations (6)-(11) we get,

$$Adv^{\lambda-CAFLR}_{\text{GPAKE}} \leq \frac{1}{(c^n_{N_P} \cdot c^2_{N_S})}(Adv_{Refresh-LRS} + Adv_{KDF}).$$

Case 2. A partner oracle to the test oracle does not exist.

In this case, $\boldsymbol{A}$ is an active adversary. He may masquerade as one of the intended partners and run the protocol with the test oracle Π^t_U. Therefore, $\boldsymbol{A}$ could not ask a ***corrupt*** **()** query to get the password.

In this case, $\boldsymbol{A}$ could get all the random keys $r_1, \cdots, r_n$ by *RevealEphemeralKey*().

Game 1: It is the original game.

Game 2: Game 2 and Game 1 only have the following differences: $\boldsymbol{A}$ selects a password pw', computes $s' = H(pw')$, encodes it, then uses the encodings of s' to generate the message based on the protocol specifications.

Game 3: Consistent with Game 2 in Case 1.1.

Game 4: Consistent with Game 3 in Case 1.2.

Game 5: Consistent with Game 4 in Case 1.2.

Game 6: Consistent with Game 4 in Case 1.1.

Differences between games:

Game 1:

$$Adv_{Game\ 1}(A) = Adv^{\lambda-CAFLR}_{\text{GPAKE}} \quad (12)$$

Game 1 and Game 2: If $pw' = pw$, Game 2 is consistent with Game 1, otherwise Game 2 is independent on Game 1. The probability that $pw' = pw$ is N_s/N. Therefore,

$$|Adv_{Game\ 2}(A)\text{-}Adv_{Game\ 1}(A)| = \frac{N_s}{N} \quad (13)$$

Game 2 and Game 3: The analysis is consistent with Game 1 and Game 2 in Case 1.1.

$$Adv_{Game\ 3}(A) = \frac{1}{(c^n_{N_P} \cdot c^2_{N_S})} Adv_{Game\ 2}(A) \quad (14)$$

Game 3 and Game 4: The analysis is consistent with Game 2 and Game 3 in Case 1.2.

$$\begin{aligned} & |Adv_{Game\ 2}(A) - Adv_{Game\ 3}(A)| \\ \leq\ & Adv_{Refresh-LRS}. \end{aligned} \qquad (15)$$

Game 4 and Game 5: The analysis is consistent with Game 3 and Game 4 in Case 1.2.

$$|Adv_{Game\ 4}(A) - Adv_{Game\ 5}(A)| = 0 \quad (16)$$

Game 5 and Game 6: The analysis is consistent with Game 4 and Game 5 in Case 1.2.

$$|Adv_{Game\ 5}(A) - Adv_{Game\ 6}(A)| \leq Adv_{KDF} \quad (17)$$

Game 6: The analysis is consistent with Game 5 in Case 1.2.

$$Adv_{Game\ 6}(A) = 0 \quad (18)$$

Using Equations (12)-(18) we get, we get:

$$\begin{aligned}&Adv^{\lambda-CAFLR}_{\text{GPAKE}}\\ \leq\ & \frac{N_S}{N} + \frac{1}{(c^n_{N_P}\cdot c^2_{N_S})}(Adv_{Refresh-LRS}\\ &\qquad\qquad +Adv_{KDF}).\end{aligned}$$

From Case 1 and Case 2, we get:

$$\begin{aligned}&Adv^{\lambda-CAFLR}_{\text{GPAKE}}\\ \leq\ & \frac{N_S}{N} + \frac{1}{(c^n_{N_P}\cdot c^2_{N_S})}(Adv_{Refresh-LRS}\\ &\qquad +Adv_{KDF}+Adv_{PDDH}).\end{aligned}$$

□

5.3 Protocol Analysis

In this section, we discuss our GPAKE protocol and compare it with other protocols [1,36,38] by the five properties: communication rounds, authentication, provability, security model and leakage-resilience. The result is shown in Table 1, which shows our protocol has the following advantages:

1) Our protocol is the first LR GPAKE protocol;

2) We give a formal security proof in the standard model, while [36] did not provide a formal security proof and [1] only gave the security proof in the RO model;

3) Our protocol is much efficient with only two rounds communications, while [36] requires $n+1$ rounds communications and [1] need 4 rounds communications.

Table 1: Comparisons of other related protocols and the proposed protocol

Scheme	[36]	[1]	[38]	**Ours**
Rounds	$n+1$	4	2	**2**
Authenticated	No	Yes	Yes	**Yes**
Provably	No	Yes	Yes	**Yes**
Security model		RO	Standard	**Standard**
LR	No	No	No	**Yes**

6 Conclusion

For traditional GPAKE protocol, it's very vulnerable to side-channel attacks, because a very small leakage may be completely exposed the whole password. In the paper, we first defined a CAFLR security model for GPAKE protocol and proposed a LR GPAKE protocol that it is suitable to securely generate a group key in the leakage environments. The proposed LR GPAKE protocol is provably secure in the standard model based on the new CAFLR security model.

References

[1] M. Abdalla, E. Bresson, O. Chevassut, and D. Pointcheval, "Password-based group key exchange in a constant number of rounds," in *Proceedings of the 9th International Conference on Theory and Practice in Public-Key Cryptography (PKC'06)*, pp. 427–442, Apr. 2006.

[2] J. Alawatugoda, C. Boyd, and D. Stebila, "Continuous after-the-fact leakage-resilient key exchange," in *Proceedings of the 19th Australasian Conference on Information Security and Privacy (ACISP'14)*, pp. 258–273, July 2014.

[3] J. Alawatugoda, D. Stebila, and C. Boyd, "Modelling after-the-fact leakage for key exchange," in *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security (ASIACCS'14)*, pp. 207–216, June 2014.

[4] J. Alawatugoda, D. Stebila, and C. Boyd, "Continuous after-the-fact leakage-resilient eck-secure key exchange," in *Proceedings of the 15th IMA International Conference Cryptography and Coding (IMACC'15)*, pp. 277–294, Dec. 2015.

[5] N. Asokan and P. Ginzboorg, "Key agreement in ad hoc networks," *Computer Communications*, vol. 23, no. 17, pp. 1627–1637, 2000.

[6] G. Ateniese, A. Faonio, and S. Kamara, "Leakage-resilient identification schemes from zero-knowledge proofs of storage," in *Proceedings of the 10th 15th IMA International Conference (IMACC'15)*, pp. 311–328, Dec. 2015.

[7] C. Boyd and J. M. G. Nieto, "Round-optimal contributory conference key agreement," in *Proceedings of the 6th International Workshop on Practice and Theory in Public Key Cryptography (PKC'03)*, pp. 161–174, Jan. 2003.

[8] E. Bresson, O. Chevassut, and D. Pointcheval, "Diffie-hellman key distribution extended to group communication," in *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'01)*, pp. 290–309, Dec. 2001.

[9] E. Bresson, O. Chevassut, and D. Pointcheval, "Group diffie-hellman key exchange secure against dictionary attacks," in *Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'02)*, pp. 497–514, Dec. 2002.

[10] M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT'94)*, pp. 275–286, May 1994.

[11] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, pp. 453–474, May 2001.

[12] S. Chakraborty, J. Alawatugoda, and C. Rangan, "Leakage-resilient non-interactive key exchange in the continuous-memory leakage setting," in *Proceedings of the International Conference on Provable Security (ProvSec'17)*, pp. 167–187, Oct. 2017.

[13] T. Chang, M. Hwang, and W. Yang, "A communication-efficient three-party password authenticated key exchange protocol," *Information Sciences*, vol. 181, no. 1, pp. 217 – 226, 2011.

[14] R. Chen, Y. Mu, G. Yang, W. Susilo, and F. Guo, "Strong authenticated key exchange with auxiliary inputs," *Designs, Codes and Cryptography*, vol. 85, no. 1, pp. 145–173, 2017.

[15] R. Chen, Y. Mu, G. Yang, W. Susilo, F. Guo, and Y. Zheng, "A note on the strong authenticated key exchange with auxiliary inputs," *Designs, Codes and Cryptography*, vol. 85, no. 1, pp. 175–178, 2017.

[16] Q. Dai, X. Zhao, Q. Xu, and H. Jiang, "A new cross-realm group password-based authenticated key exchange protocol," in *Proceedings of the 7th International Conference on Computational Intelligence and Security (CIS'11)*, pp. 856–860, Dec. 2011.

[17] S. G. Dai, J. F. Wei, and F. G. Hang, "Memory leakage-resilient secret sharing schemes," *Science China Information Sciences*, vol. 58, no. 11, pp. 1–9, 2015.

[18] F. Dav, S. Dziembowski, and D. Venturi, "Leakage-resilient storage," in *Proceedings of the International Conference on Security and Cryptography for Networks (SCN'10)*, pp. 121–137, Sep. 2010.

[19] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[20] R. Dutta and R. Barua, "Password-based encrypted group key agreement," *International Journal of Network Security*, vol. 3, no. 1, pp. 23–34, 2006.

[21] S. Dziembowski and S. Faust, "Leakage-resilient cryptography from the inner-product extractor," in *Proceedings of the 17th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'11)*, pp. 702–721, Dec. 2011.

[22] C. Guo, Z. J. Zhang, L. H. Zhu, Y. A. Tan, and Z. Yang, "Scalable protocol for cross-domain group password-based authenticated key exchange," *Frontiers of Computer Science*, vol. 9, no. 1, pp. 157–169, 2015.

[23] T. R. Halford, T. A. Courtade, and K. M. Chugg, "Energy-efficient, secure group key agreement for ad hoc networks," in *Proceedings of the IEEE Conference on Communications and Network Security (CNS'13)*, pp. 181–188, Oct. 2013.

[24] T. R. Halford, T. A. Courtade, K. M. Chugg, and X. Li, "Energy efficient group key agreement for wireless networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 10, pp. 5552–5564, 2015.

[25] M. Hedabou, "Efficient countermeasure for securing the eta pairing computation over binary fields," *International Journal of Network Security*, vol. 14, no. 1, pp. 47–52, 2012.

[26] I. Ingemarsson, D. Tang, and C. Wong, "A conference key distribution system," *IEEE Transactions on Information Theory*, vol. 28, no. 5, pp. 714–719, 1982.

[27] J. Katz and M. Yung, "Scalable protocols for authenticated group key exchange," in *Proceedings of the 23rd Annual International Cryptology Conference (CRYPTO'03)*, pp. 17–21, Aug. 2003.

[28] R. M. Kesavulu, "Elliptic curve cryptosystems and side-channel attacks," *International Journal of Network Security*, vol. 12, no. 3, pp. 151–158, 2011.

[29] H. Krawczyk and P. Eronen, *Hmac-based Extract-and-Expand Key Derivation Function*, RFC 5869, 2010.

[30] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *Proceedings of the First International Conference on Provable Security (ProvSec'07)*, pp. 1–16, Nov. 2007.

[31] S. Li, Y. Mu, and M. Zhang, "Certificate-based smooth projective hashing and its applications," *Information Sciences*, vol. 20, no. 2, pp. 266–277, 2018.

[32] D. Moriyama and T. Okamoto, "Leakage resilient eck-secure key exchange protocol without random oracles," in *Proceedings of the 6th International Symposium on Information, Computer and Communications Security (ASIACCS'11)*, pp. 441–447, Mar. 2011.

[33] O. Ruan, J. Chen, and M. Zhang, "Provably leakage-resilient password-based authenticated key exchange in the standard model," *IEEE Access*, vol. 5, pp. 26832-26841, Nov. 2017.

[34] O. Ruan, Q. Wang, and Z. Wang, "Provably leakage-resilient three-party password-based authenticated key exchange," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-11, Nov. 2017.

[35] O. Ruan, Y. Y. Zhang, J. Zhou, and L. Harn, "After-the-fact leakage-resilient identity-based authenticated key exchange," *IEEE Systems Journal*, vol. 12, no. 2, pp. 2017–2026, 2018.

[36] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-hellman key distribution extended to group communication," in *Proceedings of the ACM Conference on Computer and Communications Security (ACM CCS'96)*, pp. 31–37, Mar. 1996.

[37] F. Tang, H. Li, Q. Niu, and B. Liang, "Efficient leakage-resilient signature schemes in the generic bilinear group model," in *Proceedings of the 10th International Conference on Information Security Practice and Experience (ISPEC'14)*, pp. 418–432, May 2014.

[38] J. Teng and C. Wu, "Efficient group key agreement for wireless mobile networks," in *Proceedings of the IET International Conference on Wireless Sensor Network (IET-WSN'10)*, pp. 323–330, Nov. 2010.

[39] S. Wu and Y. Zhu, "Efficient hybrid password-based authenticated group key exchange," in *Proceedings of the Advances in Data and Web Management Joint International Conferences (AP-Web/WAIM'09)*, pp. 562–567, Apr. 2009.

[40] H. Xiong, C. Zhang, T. H. Yuen, E. P. Zhang, S. M. Yiu, and S. Qing, "Continual leakage-resilient dynamic secret sharing in the split-state model," in *Proceedings of the 14th International Conference (ICICS'12)*, pp. 119–130, Oct. 2012.

[41] C. C. Yang, T. Y. Chang, M. S. Hwang, "A new anonymous conference key distribution system based on the elliptic curve discrete logarithm problem", *Computer Standards and Interfaces*, vol. 25, no. 2, pp. 141-145, 2003.

[42] F. C. Zhou, E. G. Zhou, H. Yan, and X. X. Su, "Cross-realm group pake protocol using different passwords," *Computer Science*, vol. 36, no. 3, pp. 74–77, 2009.

[43] Y. Zhou and B. Yang, "Leakage-resilient cca2-secure certificateless public-key encryption scheme without bilinear pairing," *Information Processing Letters*, vol. 130, no. 2, pp. 16–24, 2018.

[44] L. Zhu, C. Guo, Z. Zhang, W. Fu, and R. Xu, "A novel contributory cross-domain group password-based authenticated key exchange protocol with adaptive security," in *Proceedings of the Second International Conference on Data Science in Cyberspace (DSC'17)*, pp. 213–222, June 2017.

Biography

Ou Ruan is a professor at School of Computer Sciences, Hubei University of Technology. In 2013, He received his Ph.D. at College of Information Security, School of Computer Science & Technology, Huazhong University of Science & Technology of China. His research interests include leakage-resilient cryptography, secure computations, and network security.

Zihao Wang is pursuing his Master degree from the School of Computer Science, Hubei University of Technology, Wuhan, China. His research interests include secure computations and network security.

Qingping Wang is pursuing his Master degree from the School of Computer Science, Hubei University of Technology, Wuhan, China. His research interests include leakage-resilient cryptography and information security.

Mingwu Zhang is a professor at School of Computer Sciences, Hubei University of Technology. From August 2010 to August 2012, he has been a JSPS postdoctoral fellow of Japan Society of Promotion Sciences at Institute of Mathematics for Industry in Kyushu University. His research interests include cryptography technology for networks, secure computations, and privacy preservations etc. Dr. Zhang is the director of Institute of Data Security and Privacy Preservation of HBUT.