IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 26, 2017, accepted January 24, 2018, date of publication January 29, 2018, date of current version April 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2799298

Efficient Leakage-Resilient Authenticated Key Agreement Protocol in the Continual Leakage eCK Model

JUI-DI WU, YUH-MIN TSENG, (Member, IEEE), AND SEN-SHAN HUANG
Department of Mathematics, National Changhua University of Education, Chang-Hua 500, Taiwan
Corresponding author: Yuh-Min Tseng (ymtseng@cc.ncue.edu.tw)

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST106-2221-E-018-007-MY2.

ABSTRACT Based on users' permanent private keys and ephemeral secret keys (randomness secret values), authenticated key agreement (AKA) protocols are used to construct a common session key between two session parties while authenticating each other. Recently, the design of leakage-resilient AKA (LR-AKA) resisting side-channel attacks has received significant attention from researchers. By side-channel attacks, an adversary is allowed to obtain fractional leakage information of private (secret) keys during the computation rounds of LR-AKA protocols. However, most LR-AKA protocols have a restriction, namely, the overall fractional leakage information must be bounded. In this paper, we propose an efficient LR-AKA protocol with overall unbounded leakage property in the continual leakage extended Canetti-Krawczyk model. Security analysis is given to demonstrate that our LR-AKA protocol is provably secure in the generic bilinear group model. By comparisons, our protocol is better than the previously proposed LR-AKA protocols in terms of computation cost, security model, and leakage properties.

INDEX TERMS Cryptography, key agreement, authentication, leakage-resilience, generic bilinear group.

I. INTRODUCTION

Recently, side-channel attacks have received significant attention from researchers because most of the existing cryptographic schemes/protocols did not resist this kind of attacks. Side-channel attacks mean that, when users execute these cryptographic schemes/protocols, adversaries could obtain fractional leakage information of the permanent/ephemeral private (secret) keys of users by employing several particular properties, such as differential power consumption [1], fault/bug attacks [2], [3] and timing attacks [4], [5]. Indeed, most adversary models or security notions did not concern with side-channel attacks, where they have assumed that fractional leakage information of the private (secret) keys of users could not be leaked to adversaries. Therefore, these cryptographic schemes/protocols could be broken in an environment with side-channel attacks.

Leakage-resilient cryptography is an emerging approach of resisting side-channel attacks. Indeed, the cryptographic schemes/protocols based on leakage-resilient cryptography must tolerate the fractional leakage of private (secret) keys while retaining security. Recently, numerous leakage-resilient signature schemes [6]–[10], and leakage-resilient encryption schemes [11]–[15] have been proposed. Since authenticated key agreement (AKA) is an important cryptographic primitive, it is critical to study leakage-resilient AKA (LR-AKA) protocols.

Based on users' permanent private keys and ephemeral secret keys (randomness secret values), AKA protocols are used to construct a common session key between two communication parties while authenticating each other. For the security model, Bellare and Rogaway [16] presented the first adversary model of AKA protocols. Afterward, Canetti and Krawczyk [17] extended Bellare and Rogaway's model to present a new adversary model, called CK model. Since the CK model did not address several possible attacks, LaMacchia *et al.* [18] further presented the extended Canetti−Krawczyk (eCK) model by considering stronger adversaries, who can compromise either permanent private keys or ephemeral secret keys. The eCK model has widely been used to show the security of AKA protocols [19]–[22]. However, the aforementioned AKA models and protocols did not take into account side-channel attacks with fractional leakage of permanent/ephemeral private (secret) keys. In this article, we will aim at the design

of an efficient LR-AKA protocol with strong security properties, especially overall unbounded leakage property and capturing general leakage attacks.

A. RELATED WORK

Indeed, the construction of LR-AKA protocols can be constructed straightforwardly by employing leakage-resilient encryption or signature schemes. Several LR-AKA protocols have been proposed. In 2009, Alwen *et al.* [23] employed a secure signature scheme to propose a leakage-resilient AKA protocol. Afterward, Dodis *et al.* [24] proposed two LR-AKA protocols by, respectively, employing leakage-resilient encryption and signature schemes. Meanwhile, they also proved that both protocols are secure in the leakage-resilient CK model of LR-AKA protocols. Moreover, by combing a secure AKA protocol under CK model and a random message unforgeable signature scheme, Yang *et al.* [25] proposed a secure LR-AKA protocol in the leakage-resilient CK model and the auxiliary input model. It is obvious that the leakage-resilient CK model did not address the compromise of ephemeral secret keys.

Moreover, Moriyama and Okamoto [26] introduced a leakage-resilient eCK model of LR-AKA protocols while proposing a concrete LR-AKA protocol to concern with the compromise of ephemeral secret keys. However, their LR-AKA protocol concerned with only the fractional leakage of the permanent private key, but not the ephemeral secret key. Very recently, to address the security incompleteness of the aforementioned LR-AKA protocols, Chen *et al.* [27] proposed a new adversary model, termed leakage-resilient eCK model with auxiliary inputs. Their model allows the fractional leakage of both the permanent private key and ephemeral secret key while enabling an adversary to issue leakage queries during the challenge session of LR-AKA protocols. Nevertheless, the LR-AKA protocols mentioned above have the restriction that the total amount of fractional leakage information must be bounded, called the bounded leakage model.

Alawatugoda *et al.* [28] and [29] presented a generic leakage-resilient eCK model of LR-AKA protocols, called after-the-fact leakage eCK model. In this model, adversaries are also allowed to obtain fractional leakage information of the permanent/ephemeral private (secret) keys even after the session key is established during the test/challenge session. In addition, the after-the-fact leakage eCK model has two variants, namely, bounded leakage and continual leakage. The former bounds the total amount of fractional leakage information of each user's permanent/ephemeral private (secret) keys for the entire protocol execution, whereas the continual leakage variant allows adversaries to reveal a fixed amount of leakage for each protocol session while possessing overall unbounded leakage property during protocol execution. Alawatugoda *et al.* [29] also proposed a concrete construction of LR-AKA protocol. However, the proposed LR-AKA protocol is only secure in the bounded leakage variant of after-the-fact leakage eCK model. Afterwards, Alawatugoda *et al.* [30] proposed the first concrete and secure LR-AKA protocol in the continual leakage variant of after-the-fact leakage eCK model (abbreviated "continual leakage eCK model"). They employed Dziembowski and Faust's key refreshing technique [31] to update the permanent private keys after each protocol session. However, the key refreshing procedure adopts the inner-product extractor method to compute the next permanent private key so that the required computational cost is heavy for achieving secure key refreshing procedure. The secure key refreshing procedure is a protocol which requires $O(n^2)$ computation operation, where n is a security parameter and depends on the leakage amount in each round of the secure key refreshing procedure. Indeed, Alawatugoda *et al.* [30] pointed out that it is worthwhile to investigate other techniques to realize continual after-the-fact leakage resilience LR-AKA protocol without inner-product extractor method.

B. CONTRIBUTION AND ORGANIZATION

In this article, we propose a novel and efficient LR-AKA protocol with overall unbounded leakage property in the continual leakage eCK model. In the continual leakage eCK model, our LR-AKA protocol allows adversaries to obtain fractional leakage information of both user's permanent private keys and ephemeral secret keys involved in the session key after the test/challenge session. The point is that the security model possesses the overall unbounded leakage property [13]. As the splitting storage idea [31], we also partition a permanent private key into two components, and refresh these two components by employing the multiplicative blinding technique [13], [15], [32] instead of the time-consuming inner-product extractor method. In this case, two new components of the permanent private key can be re-used safely since the leakage would be restricted to two "current" components and therefore no adversary can learn the useful information about a permanent private key.

In the generic bilinear group model [33], we demonstrate that our LR-AKA protocol is provably secure in the continual leakage eCK model. Table 1 lists the property comparisons among the aforementioned protocols [23]–[27], [29], [30] and our LR-AKA protocol in terms of AKA model, admitted leakage of randomness, overall leakage amount and computational cost. Obviously, our protocol is better than the others. Finally, performance analysis is made to demonstrate that the proposed LR-AKA protocol is suitable for both PC and mobile devices.

The rest of this paper is structured as follows. Section II gives preliminaries that include the generic bilinear group model and the associated assumptions. In Section III, the framework and security notions of LR-AKA protocols in the continual leakage eCK model are presented. A novel and efficient LR-AKA protocol with overall unbounded leakage property is proposed in Section IV. Security analysis of our LR-AKA protocol is presented in Section V. Section VI

TABLE 1. Property comparisons among our protocol and the previously proposed protocols.

Protocols	AKA model	Admitted Leakage of Randomness	Overall leakage	Computational cost
Alwen *et al.*'s protocol [23]	CK	No	Bounded	$O(1)$
Dodis *et al.*'s protocol [24]	CK	No	Bounded	$O(1)$
Yang *et al.*'s protocol [25]	CK	No	Bounded (Auxiliary input)	$O(1)$
Moriyama and Okamoto's protocol [26]	**eCK**	No	Bounded	$O(1)$
Alawatugoda *et al.*'s protocol [29]	**eCK**	No	Bounded	$O(1)$
Chen *et al.*'s protocol [27]	**eCK**	**Yes**	Bounded (Auxiliary input)	$O(1)$
Alawatugoda *et al.*'s protocol [30]	**eCK**	**Yes**	**Unbounded**	$O(n^2)$
Our protocol	eCK	**Yes**	**Unbounded**	$O(1)$

demonstrates performance analysis. Finally, conclusions are given in Section VII.

II. PRELIMINARIES

Here, we compendiously present properties of bilinear groups [34], [35] and entropy, concepts of the generic bilinear group model [9], [32], [33], and several associated hard assumptions.

A. BILINEAR GROUPS

Let G be a multiplicative cyclic group generated by g and let its order be a large prime p. Let G_T be also a multiplicative cyclic group of the same order p. An admissible bilinear pairing is a map $\hat{e}$: $G \times G \rightarrow G_T$ which possesses the following properties:

1. *Bilinearity*: for all $a, b \in Z_p^*$, $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab} = \hat{e}(g^b, g^a)$.
2. *Non − degeneracy*: $\hat{e}(g, g) \neq 1$.
3. *Computability*: for all $g_1, g_2 \in G$, the operation $\hat{e}(g_1, g_2)$ can be computed efficiently.

For the admissible map $\hat{e}$, G and G_T are called a bilinear group and the target group of the map $\hat{e}$, respectively. In addition, $\hat{e}(g, g)$ may be viewed as a generator of G_T, denoted by g_T. Such groups appear in hyper-elliptic curves or supersingular elliptic curves over finite field. We refer the reader to [34] and [35] for further details.

B. GENERIC BILINEAR GROUP MODEL

The generic group model is an adversary model for cryptographic schemes and protocols, which was first proposed by Shoup [36]. Based on the generic group model, Boneh *et al.* [33] further present the generic bilinear group model by adding bilinear pairing operation. The generic bilinear group model involves three group operations, namely, a multiplication in a group G, a multiplication in a group G_T and a bilinear pairing $\hat{e}$ between G and G_T. When an adversary would like to perform a group operation, it just issues a group query (oracle) to a challenger. Upon receiving this group query, the challenger uses the two elements to generate third element in G, records it in a list and sends it to the adversary. Namely, the adversary may have access to a randomly chosen element (encoding) of a group, which is maintained and controlled by the challenger.

In the generic bilinear group model, the elements of a group are encoded with bit strings. In such a case, the elements of G and G_T are, respectively, encoded to bit strings by using two random injective maps ψG: $Z_p \rightarrow \Omega$ and ψT: $Z_p \rightarrow \Omega_T$, where both Ω and Ω_T are sets of bit strings while satisfying $\Omega \cap \Omega_T = \phi$ and $|\Omega| = |\Omega_T| = p$. Also, let Q_G, Q_T and Q_p denote, respectively, group queries (oracles) on the multiplication operation in G, the multiplication operation in G_T and the bilinear pairing operation $\hat{e}$. For any $a, b \in Z_p^*$, the following properties hold.

- $Q_G(\psi G(a), \psi G(b)) \rightarrow \psi G(a + b \bmod p)$.
- $Q_T(\psi T(a), \psi T(b)) \rightarrow \psi T(a + b \bmod p)$.
- $Q_p(\psi G(a), \psi G(b)) \rightarrow \psi T(ab \bmod p)$.

It is worth mentioning that $\psi G(1) = g$ and $\psi T(1) = \hat{e}(g, g) = g_T$ are generators of G and G_T, respectively. One main employment of the generic bilinear group model is to analyze computational hardness assumptions. In the generic bilinear group model, if an adversary with non-negligible probability can find a collision element for a group operation, we say that it solves the computational hardness assumption, i.e. solving the discrete logarithm problem in the multiplicative group G or G_T.

Definition 1: Discrete logarithm (DL) problem and its associated assumption: Given two group elements $P, Q \in G$, where G is a multiplicative cyclic group with order to be a large prime p. The DL problem in G is to compute an integer $c \in Z_p^*$ such that $Q = P^c$, where P is a generator of G. The advantage of any probabilistic polynomial time (PPT) algorithm A in solving the DL problem in G is defined by $Adv_A^{DL} = \text{pr}[A(P, Q \in G) = c | c \in Z_p^*]$. The DL assumption is that the advantage Adv_A^{CDH} is negligibly small for any probabilistic polynomial time algorithm A [33].

In addition, we also need another computational hardness assumption in our protocol.

Definition 2: Computational Diffie-Hellman (CDH) problem: Given $(g, g^a, g^b) \in G^3$ for unknown $a, b \in Z_p^*$, where G is a multiplicative cyclic group with order to be a large prime p and with a generator g. The *CDH* problem in G is to compute g^{ab}. The advantage of any PPT algorithm A in solving the CDH problem in G is defined by $Adv_A^{CDH} = \text{pr}[A(g, g^a, g^b) = g^{ab} | a, b \in Z_p^*]$. The CDH assumption is that the advantage Adv_A^{CDH} is negligibly small [37] for any PPT algorithm A.

C. ENTROPY

Entropy is used to measure the amount of all possible statuses (states) in thermodynamic equilibrium systems. Indeed, the statistical exposition of entropy is able to evaluate the probability estimation of uncertainty. Let X be a discrete random variable which takes on a finite set of values

$x_1, x_2, \ldots, x_n$ with probability $Pr[x_1]$, $Pr[x_2]$, $\ldots$, $Pr[x_n]$ such that $\sum_{x \in X} Pr[x] = 1$. Moreover, the min-entropy of a random variable denotes the measure of the largest probability (worst-case predictability). The average conditional min-entropy of a random variable denotes the measure of the worst-case predictability under a correlated discrete random variable with some events. We formally define the two kinds of min-entropies as below.

1. $H_\infty(X) = -log_2(\max_x Pr[X = x])$: the min-entropy of the discrete random variable X.
2. $\widetilde{H}_\infty(X|Y) = -log_2(E_{y \leftarrow Y}[\max_x Pr[X = x|Y = y]]$: the average conditional min-entropy of the discrete random variable X under the correlated discrete random variable Y with an event $Y = y$.

In the leakage circumstance, to measure the average conditional min-entropy of a discrete random variable (i.e. a private/secret key), Dodis *et al.* [38] presented the following consequence.

Lemma 1: Given a discrete random variable X, let $f(X)$ denote the fractional leakage information on X, where f :x$\rightarrow$ $\{0, 1\}^\lambda$ is a leakage function on X and the output bit-length of f is limited to λ bits. The average conditional min-entropy of the discrete random variable X under the fractional leakage information $f(X)$ satisfies the inequality $\widetilde{H}_\infty(X|f(X)) \geq H_\infty(X) - \lambda$.

Based on the Schwartz-Zippel lemma [39], [40], Galindo and Vivek [9] demonstrated the property of probability distributions of a non-zero polynomial under a leakage function with the maximal output bit-length λ as below.

Lemma 2: Let $G \in Z_p[W_1, W_2, \ldots, W_n]$ denote a non-zero polynomial of total degree d with the probability distributions P_i (for $i = 1, 2, \ldots, n$) on Z_p such that $H_\infty(P_i) \geq logp - \lambda$ and $0 \leq \lambda \leq logp$. If $w_i \xleftarrow{P_i} Z_p$ (for $i = 1, 2, \ldots, n$) are mutually independent, we have the consequence $Pr[G(W_1 = w_1, W_2 = w_2, \ldots, W_n = w_n) = 0] \leq \frac{d}{p} 2^\lambda$. Meanwhile, if $\lambda < log(p) - \omega(loglogp)$, the probability $Pr[W_1 = w_1, W_2 = w_2, \ldots, W_n = w_n) = 0]$ is negligible.

III. ADVERSARY MODEL AND SECURITY NOTIONS

In the section, we introduce the security notions of the continual leakage variant of after-the-fact leakage eCK model [30], i.e. the continual leakage eCK model. We first present properties of the continual leakage model.

A. CONTINUAL LEAKAGE MODEL

The continual leakage model is used to model the leakage abilities of an adversary, which allows an adversary to continually reveal a fixed amount of leakage for each protocol session while possessing overall unbounded leakage property during the whole system lifecycle [9], [13], [32]. To achieve the overall unbounded leakage property, private (secret) keys must provide the *stateful* property. To do so, each private (secret) key is partitioned into two components and stored in different places of the memory. Generally, a cryptographic protocol/scheme comprises several computation rounds. After (or before) executing a computation round in the cryptographic protocol/scheme, the system refreshes the involved private (secret) key while the associated public key remains unchanged. In the following, we summarize four properties of the continual leakage model.

- *Only computation leakage*: Only the fractional leakage information of permanent/temporary private (secret) keys involved or accessed in the current computation round could be revealed to a side-channel adversary.
- *Bounded leakage of single computation round*: The amount of fractional leakage information in a single computation round is limited to some λ bits. Namely, the leakage information of each computation round is bounded to a fraction of private (secret) keys.
- *Independent leakage between computation rounds*: The leakage information of the computation rounds is independent with each other.
- *Overall unbounded leakage*: The total amount of leakage information is unbounded, namely, it possesses overall unbounded leakage property during the whole system lifecycle. Thus, after (or before) executing a computation round, the system refreshes the involved or accessed private (secret) keys.

B. THE ADVERSARIAL MODEL OF THE LR-AKA PROTOCOL

As mentioned earlier, the eCK model of AKA protocols introduced by LaMacchia *et al.* [18] is an accredited adversary model and has been widely used to show the security of AKA protocols. Based on the eCK model, several leakage-resilient eCK models [26]–[30] have been presented. Alawatugoda *et al.*'s continual leakage eCK model [29], [30] is the most accredited model, which allows an adversary to ask all kinds of queries as the abilities of the adversary in the eCK model. In addition, the adversary may also issue leakage queries to obtain the fractional leakage information of permanent/ephemeral private (secret) keys. Typically, a AKA protocol consists of two phases, namely, initial setup phase and session key construction phase. In the continual leakage eCK model, the session key construction phase of leakage-resilient (LR)-AKA protocols is further divided to two sub-phases, namely, key refreshing and key agreement. To represent the fractional leakage information of LR-AKA protocol obtained by the adversary, we choose two leakage functions $f_{i,s}$ and $h_{i,s}$, respectively, to model the adversary's abilities in the key refreshing and key agreement sub-phases, where i and s denote the s-th session of the user with identity ID_i. The output length of $f_{i,s}$ and $h_{i,s}$ are bounded by λ, where λ is the leakage parameter. That is, $|f_{i,s}| \leq \lambda$ and $|h_{i,s}| \leq \lambda$, where $|f|$ denotes the output length of leakage function f. Moreover, two leakage functions $f_{i,s}$ and $h_{i,s}$ can be efficiently computed. We define the outputs of two leakage functions as follows.

- $\Lambda f_{i,s} = f_{i,s}$ (*private keys*).
- $\Lambda h_{i,s} = h_{i,s}$ (*private keys*, *ephemeral secret keys*).

Here, *private keys* denote the private keys involved in the computations of the key refreshing and key agreement sub-phases while *ephemeral secret keys* denote the ephemeral secret keys involved in the computation of the key agreement sub-phase.

Based on the continual leakage eCK model, we present the associated security game G_{CL-eCK} that is played by an adversary A and a challenger B. Let the adversary A be a q-query probabilistic polynomial-time (PPT) algorithm that can issue queries to the challenger B at most q times. Let the oracle Π_i^s is denote the s-th session of the user with identity ID_i.

GAME G_{CL-eCK}

In the game G_{CL-eCK}, there are two phases, namely, *Initial Setup* and *Query*. The adversary A may issue six kinds of queries in any order for totally at most q times in the Query phase. A wins the game if A can determine whether or not a bit string is the real session key at the end of the game. Two phases are described as below:

- *Initial Setup*: The challenger B generates the system parameters and then sends the public parameters to the adversary A.
- *Query*: In this phase, A can issue the following six queries adaptively for totally at most q times.
  - *Send*(Π_i^s, m): Upon receiving this query along with the communication message m to the oracle Π_i^s, B sends the corresponding results to A by running the protocol Π_i^s according to m.
  - *Reveal*(Π_i^s): A can issue this query to obtain the session key of the oracle Π_i^s if Π_i^s has accepted the session; otherwise, it returns a null value.
  - *Ephemeral-secret-leakage*(Π_i^s): A can issue this query to obtain the ephemeral secret keys of Π_i^s.
  - *Corrupt*(ID_i): A can issue this query to obtain the private key of the user with identity ID_i.
  - *Leak*($\Pi_i^s, f_{i,s}, h_{i,s}$): A can issue this query along with the target oracle Π_i^s and two leakage function $f_{i,s}$ and $h_{i,s}$ to obtain the fractional leakage information of the private keys and ephemeral secret keys in the key refreshing and key agreement sub-phases of Π_i^s.
  - *Test*(Π_i^s): When the adversary A issues this query, the challenger B flips an unbiased coin bit cb. The challenger B returns the session key if $cb = 1$; otherwise, it returns a random value. The query phase ends whenever the Test query has been issued by A. If A can return a coin bit $cb' = cb$, then A wins the game G_{CL-eCK}. We emphasize that the adversary A is allowed to issue the Test(Π_i^s) query only once.

In the following, we describe the relationships between the real world attack scenarios and different queries in the continual leakage eCK model.

- Passive adversarial capabilities: The *Send* query addresses the power of a passive adversary who may control and obtain message flows between two communication parties.
- Malware attacks: The *Corrupt* query addresses the malware attack in the situation that the adversary A may reveal the private keys of users. The *Ephemeral-secret-leakage* query addresses the malware attacks in the situation that the adversary A may reveal the ephemeral secret keys of any connections.
- Weak random number generators: Due to the weak random number generators, the adversary can determine the random number correctly. The *Ephemeral-secret-leakage* query addresses such kinds of ability of the adversary.
- Known-session-key security: The *Reveal* query addresses the known-session-key security in the sense that an adversary cannot reveal other session keys when it compromises a session key.
- Side-channel attacks: Whenever the leakage happens due to any kinds of side-channel attacks, the adversary may obtain the fractional leakage information of the parameters involved in the computation. The Leak query addresses the side-channel attacks.

Definition 3 (Partnership): Two oracles Π_i^s and Π_j^t are *partners* if they can authenticate mutually and accept a common session key.

Definition 4 (Freshness): In the security game G_{CL-eCK}, the partnership session between Π_i^s and Π_j^t is said to be *fresh* if none of the following conditions holds:

- The adversary has issued *Reveal*(Π_i^s) or *Reveal*(Π_j^t).
- The adversary has issued both *Corrupt*(ID_i) and *Ephemeral-secret-leakage*(Π_i^s).
- The adversary has issued both *Corrupt*(ID_j) and *Ephemeral-secret-leakage*(Π_j^t).

IV. THE PROPOSED LR-AKA PROTOCOL

In this section, we propose a novel and efficient LR-AKA protocol in the continual leakage eCK model. Our LR-AKA protocol consists of two phases, initial setup and session key construction. Moreover, the session key construction phase consists of two sub-phases including the key refreshing and key agreement. In the following, we demonstrate how Alice and Bob construct a common session key by using our LR-AKA protocol.

- *Initial setup phase*: Given the security parameter κ, the system first generates the bilinear groups $\{p, G, G_T, g, \hat{e}\}$ as defined in Section 2.1. Moreover, an additional generator $h \in G$ is randomly chosen. Let the public parameters PP be $\{G, G_T, \hat{e}, p, g, h\}$. Without loss of generality, Alice and Bob generate their own private key pairs and public key pair as follows. Also, see Fig. 1.
  - Alice first picks two random values $a, \alpha_0 \in Z_p^*$, and computes two initial private key pairs $(SA_{0,1}, SA_{0,2}) = (g^{\alpha_0}, g^{a-\alpha_0})$ and $(XA_{0,1}, XA_{0,2}) = (\hat{e}(SA_{0,1}, h), \hat{e}(SA_{0,2}, h))$. Obviously, we have $SA = SA_{0,1} \cdot SA_{0,2} = g^a$ and $XA = XA_{0,1} \cdot XA_{0,2} = \hat{e}(g^a, h)$. Alice computes the public key pair $(TA, PA) = (\hat{e}(g, SA), \hat{e}(g, g)^{XA})$.

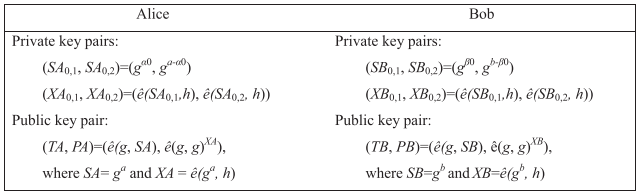

FIGURE 1. The Initial setup phase.

- Similarly, Bob picks two random values $b, \beta_0 \in Z_p^*$, and computes two private key pairs and the public key pair as Alice did. Bob's two initial private key pairs are $(SB_{0,1}, SB_{0,2})$ and $(XB_{0,1}, XB_{0,2})$ while the public key pair is $(TB, PB) = (\hat{e}(g, SB), \hat{e}(g, g)^{XB})$, where $SB = SB_{0,1} \cdot SB_{0,2} = g^b$ and $XB = XB_{0,1} \cdot XB_{0,2} = \hat{e}(g^b, h)$.

- *Session key construction phase*: To construct a session key between Alice's i-th and Bob's j-th session, they perform the key refreshing and key agreement sub-phases as follows. Also, see Fig. 2.
  - *Key refreshing*: Alice chooses a random number $\alpha_i \in Z_p^*$ and refreshes her private key pairs by $(SA_{i,1}, SA_{i,2}) = (SA_{i-1,1} \cdot g^{\alpha_i}, SA_{i-1,2} \cdot g^{-\alpha_i})$ and $(XA_{i,1}, XA_{i,2}) = (XA_{i-1,1} \cdot \hat{e}(g^{\alpha_i}, h), XA_{i-1,2} \cdot \hat{e}(g^{-\alpha_i}, h))$. By the same way, Bob also picks a random number $\beta_j \in Z_p^*$ and refreshes his private key pairs $(SB_{j,1}, SB_{j,2}) = (SB_{j-1,1} \cdot g^{\beta_j}, SB_{j-1,2} \cdot g^{-\beta_j})$ and $(XB_{j,1}, XB_{j,2}) = (XB_{j-1,1} \cdot \hat{e}(g^{\beta_j}, h), XB_{j-1,2} \cdot \hat{e}(g^{-\beta_j}, h))$. This key refreshing procedure adopts the so-called multiplicative blinding technique. Obviously, we have
    $SA_{i,1} \cdot SA_{i,2} = g^a = SA = SA_{0,1} \cdot SA_{0,2}$.
    $XA_{i,1} \cdot XA_{i,2} = \hat{e}(g^a, h) = XA = XA_{0,1} \cdot XA_{0,2}$.
    $SB_{j,1} \cdot SB_{j,2} = g^b = SB = SB_{0,1} \cdot SB_{0,2}$.
    $XB_{j,1} \cdot XB_{j,2} = \hat{e}(g^b, h) = XB = XB_{0,1} \cdot XB_{0,2}$.
  - *Key agreement*: Alice chooses an ephemeral secret key x, computes $X = g^x$ and sends the value X to Bob. Bob also chooses an ephemeral secret key y, computes $Y = g^y$ and sends the value Y to Alice. By using Bob's public key (TB, PB) and Y, and her own current private key pairs $(SA_{i,1}, SA_{i,2})$ and $(XA_{i,1}, XA_{i,2})$, Alice computes a session key $SK_{A,i}$ as below:

    (1) $KA_{i,1} = TB^x$.
    (2) $KA_{i,2} = \hat{e}(Y, SA_{i,1}) \cdot \hat{e}(Y, SA_{i,2})$.
    (3) $KA_{i,3} = Y^x$.
    (4) $KA_{i,4} = (PB^{XA_{i,1}})^{XA_{i,2}}$.
    (5) $SK_{A,i} = KA_{i,1} \oplus KA_{i,2} \oplus KA_{i,3} \oplus KA_{i,4}$.

    Similarly, Bob uses his current private key pairs $(SB_{j,1}, SB_{j,2})$ and $(XB_{j,1}, XB_{j,2})$ to compute a session key $SK_{B,j}$ as below:

    (1) $KB_{j,1} = TA^y$.
    (2) $KB_{j,2} = \hat{e}(X, SB_{j,1}) \cdot \hat{e}(X, SB_{j,2})$.
    (3) $KB_{j,3} = Y^x$.

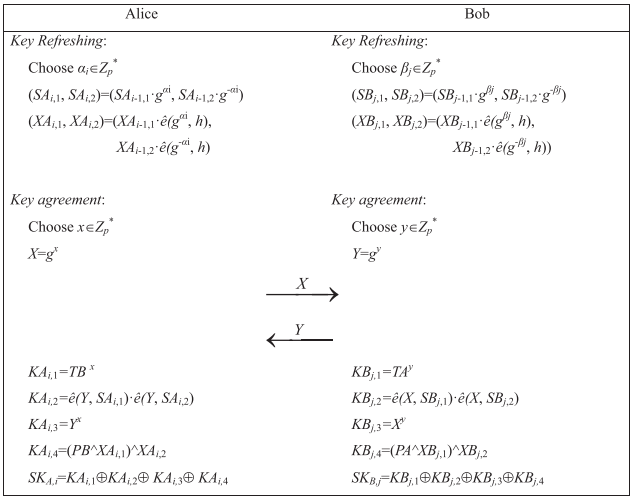

FIGURE 2. The session key construction phase.

(4) $KB_{j,4} = (PA^{XB_{j,1}})^{XB_{j,2}}$.
(5) $SK_{B,j} = KB_{j,1} \oplus KB_{j,2} \oplus KB_{j,3} \oplus KB_{j,4}$.

In the following, we show the correctness of the section keys KA_i and KB_j. Since $SK_{A,i} = KA_{i,1} \oplus KA_{i,2} \oplus KA_{i,3} \oplus KA_{i,4}$ and $SK_{B,j} = KB_{j,1} \oplus KB_{j,2} \oplus KB_{j,3} \oplus KB_{j,4}$, We show the equality $SK_{A,i} = SK_{B,j}$ by the following equalities:

$$\begin{aligned}
KA_{i,1} &= TB^x = \hat{e}(g, g)^{bx} = \hat{e}(g^x, g^b) \\
&= \hat{e}(X, g^b) = \hat{e}(X, SB_{j,1} \cdot SB_{j,2}) \\
&= \hat{e}(X, SB_{j,1}) \cdot \hat{e}(X, SB_{j,2}) = KB_{j,2}. \\
KA_{i,2} &= \hat{e}(Y, SA_{i,1}) \cdot \hat{e}(Y, SA_{i,2}) \\
&= \hat{e}(Y, SA_{i,1} \cdot SA_{i,2}) \\
&= \hat{e}(Y, g^a) = \hat{e}(g^y, g^a) \\
&= \hat{e}(g, g^a)^y = TA^y = KB_{j,1}. \\
KA_{i,3} &= Y^x = g^{yx} = g^{xy} = X^y = KB_{j,3}. \\
KA_{i,4} &= (PB^{XA_{i,1}})^{XA_{i,2}} \\
&= PB^{(}XA_{i,1} \cdot XA_{i,2}) = \hat{e}(g, g)^{XB \cdot XA} \\
&= PA^{XB} = PA^{XB_{j,1} \cdot XB_{j,2}} = KB_{j,4}.
\end{aligned}$$

Hence, we have $SK_{A,i} = SK_{B,j}$.

V. SECURITY ANALYSIS

In this section, we present the security analysis of our LR-AKA protocol in the continual leakage eCK model. Based on the generic bilinear group model [13], [32], [33], we demonstrate that our LR-AKA protocol is provably secure in the continual leakage eCK model.

Theorem 1: Assume that A is a q-query PPT adversary of the proposed LR-AKA protocol in the continual leakage eCK model. Based on the generic bilinear group model and CDH assumption, the proposed LR-AKA protocol is provably secure.

Proof: Let A be an adversary that can adaptively issue the queries at most q times in the security game G_{CL-eCK} which is played by a challenger B and the adversary A.

The advantage that A breaks the proposed LR-AKA protocol is bounded by the success probability of A in the game G_{CL-eCK}. In the adversary model defined in Section 3.2, A can issue the *Ephemeral-secret-leakage* and *Corrupt* queries to obtain the ephemeral secret key and the private key pairs of a participant, respectively. Without loss of generality, let Π_i^s and Π_j^t be the two oracles of a partnership session, where Π_i^s denotes the s-th session of the user with identity ID_i while Π_j^t represent the t-th session of the user with identity ID_j. The session key of a partnership session between Π_i^s and Π_j^t can be obtained by making use of both the private key pairs and the ephemeral secret key of one participant Π_i^s or Π_j^t. Without breaking the freshness of the partnership session between Π_i^s and Π_j^t, we discuss four situations as below:

- Situation 1: A may obtain the private key pairs of both Π_i^s and Π_j^t, but not the ephemeral secret key of Π_i^s or Π_j^t.
- Situation 2: A may obtain the ephemeral secret keys of both Π_i^s and Π_j^t, but not the private key pairs of Π_i^s or Π_j^t.
- Situation 3: A may obtain the ephemeral secret key of Π_i^s and the private key pairs of Π_j^t, but not the ephemeral secret key of Π_j^t or the private key pairs of Π_i^s.
- Situation 4: A obtains the ephemeral secret key of Π_j^t and the private key pairs of Π_i^s, but not the ephemeral secret key of Π_i^s or the private key pairs of Π_j^t.

We employ three lemmas to establish the security for each situation. The security of Situation 1 is given in Lemma 3. By Lemma 4, we show the security of Situations 3 and 4. Finally, in Lemma 5, we give the security of Situation 2. Hence, by applying Lemmas 3, 4 and 5, our LR-AKA protocol is provable secure in the continual leakage eCK model. Q.E.D.

Lemma 3: Assume that A is a q-query PPT adversary of the proposed LR-AKA protocol in the continual leakage eCK game G_{CL-eCK}. Assume that A is allowed to obtain the private key pairs of both Π_i^s and Π_j^t, but not the ephemeral secret keys of Π_i^s or Π_j^t. Under the CDH assumption, the probability that A wins the continual leakage eCK game G_{CL-eCK} is negligibly small.

Proof: Note that the adversary A can compute $KA_{i,1}$ $(= KB_{j,2})$, $KA_{i,2}(= KB_{j,1})$ and $KA_{i,4}(= KB_{j,4})$. Hence, to obtain the session key $SK_{A,i}(= KA_{i,1} \oplus KA_{i,2} \oplus KA_{i,3} \oplus KA_{i,4})$, A must be able to compute $KA_{i,3}(= Y^x = g^{xy} = X^y = KB_{j,3})$ from the instance $(G, p, g, X = g^x, Y = g^y)$, where x and y are the ephemeral secret key. However, this is the hard CDH problem and so the advantage that A can obtain g^{xy} is negligibly small. In the continual leakage eCK game G_{CL-eCK}, A can collect at most λ bits leakage information about x or y by issuing the Leak query with two leakage functions $h_{i,s}$ and $h_{j,t}$, respectively. Since the ephemeral secret key x and y are randomly selected for each session, A can obtain at most λ bits leakage information from each ephemeral secret key. Hence it is hard to determine $KA_{i,3}$ or $KB_{j,3}$. Based on the CDH assumption [37], the probability that A wins the game G_{CL-eCK} is negligibly small. Q.E.D.

Lemma 4: Assume that A is a q-query PPT adversary of the proposed LR-AKA protocol in the continual leakage eCK game G_{CL-eCK}. Assume that A is allowed to obtain the ephemeral secret key of Π_i^s and the private key pairs of Π_j^t, or the ephemeral secret key of Π_j^t and the private key pairs of Π_i^s. Based on the bilinear generic group model, the advantage that A wins the game is negligibly small.

Proof: By assumption (of Situation 3), A is allowed to obtain the ephemeral secret key x of Π_i^s and the private key pairs $(SB_{j,1}, SB_{j,2})$ and $(XB_{j,1}, XB_{j,2})$ of Π_j^t, but not the ephemeral secret key y of Π_j^t or the private key pairs $(SA_{i,1}, SA_{i,2})$ and $(XA_{i,1}, XA_{i,2})$ of Π_i^s. In this case, by using x, $(SB_{j,1}, SB_{j,2})$ and $(XB_{j,1}, XB_{j,2})$, the adversary A can compute $KA_{i,1}(= KB_{j,2})$, $KA_{i,3}(= KB_{j,3})$ and $KA_{i,4}(= KB_{j,4})$. Hence, to obtain the session key, A must be able to compute $KA_{i,2}(= \hat{e}(Y, SA_{i,1}) \cdot \hat{e}(Y, SA_{i,2}) = \hat{e}(Y, SA_{i,1} \cdot SA_{i,2}) = \hat{e}(Y, g^a) = \hat{e}(g^y, g^a) = TA^y = KB_{j,1})$. However, this is hard. Compute $KA_{i,2}$ can be viewed as a key encapsulation by employing leakage resilient ElGamal encryption scheme proposed by Kiltz and Pietrzak [13]. Kiltz and Pietrzak have shown the security of the key encapsulation for $\hat{e}(Y, SA_{i,1}) \cdot \hat{e}(Y, SA_{i,2}) = TA^y$ under the continual leakage model and based on the generic bilinear group model. Thus, the probability that A can obtain $KA_{i,2} = \hat{e}(Y, SA_{i,1}) \cdot \hat{e}(Y, SA_{i,2}) = TA^y = KB_{j,1}$ is negligibly small. Similarly, Situation 4 is also a key encapsulation from Alice to Bob, we omit the proof. Q.E.D.

Lemma 5: Assume that A is a q-query PPT adversary of the proposed LR-AKA protocol in the continual leakage eCK game G_{CL-eCK}. Assume that A is allowed to obtain the ephemeral secret keys of both Π_i^s and Π_j^t, but not the private key pairs of both Π_i^s and Π_j^t. Based on the generic bilinear group model, the probability that A wins the continual leakage eCK game G_{CL-eCK} is negligibly small.

Proof: The proof of the lemma is presented in Appendix.

VI. PERFORMANCE ANALYSIS

The performance analysis of our LR-AKA protocol is given here. We adopt the following notations to analyze the computational costs.

- TG_e: The computational cost of a bilinear pairing operation $\hat{e}$: $G \times G \rightarrow G_T$.
- TG_m: The computational cost of an exponentiation operation in G or G_T.

Note that the computational cost of the multiplication operation in G or G_T is trivial and negligible with compared to TG_e and TG_m [35]. The simulation results in [41]–[43] on mobile device and PC are treated as the benchmark of evaluating the running time of TG_e and TG_m. The mobile device is a Linux personal digital assistant (PDA) with a PXA270 624-MHz processor. The PC is equipped with a Pentium 3 GHz processor under Microsoft window system. In addition, for the equivalent level of 1024-bit RSA security, the employed bilinear pairings (i.e. Tate pairings) are defined over the elliptic curve E/F_p: $y^2 = x^3 + x$ with embedding

TABLE 2. Running time (in milliseconds) of various operations on mobile device and PC.

	TG_e	TG_m
PC (3 GHz processor)	20.1	6.38
Mobile device (624 MHz processor)	96.2	30.67

TABLE 3. Computational cost and running time (in milliseconds) of our LR-AKA protocol.

	Initial Setup	Key refreshing	Key agreement
Computational cost	$3TG_e$ $+3TG_m$	$2TG_e$ $+2TG_m$	$2TG_e$ $+4TG_m$
Running time on PC	79.44	52.96	65.72
Running time on mobile device	380.61	253.74	315.08

degree 2, where p is a 512-bit prime such that $p + 1 = 12qr$ while q is a 160-bit prime. Table 2 lists the running time (in milliseconds) of two operations on mobile device and PC, respectively. In Table 3, we list the computational cost and running time (in milliseconds) of three phases in the proposed LR-AKA protocol. By Table 3, the performance of our protocol is not only suiting for the platforms of PC but also for mobile devices.

VII. CONCLUSION

In the article, we proposed a novel and efficient LR-AKA protocol by using the multiplicative blinding technique instead of the time-consuming inner-product extractor method to achieve key refreshing. The idea of the multiplicative blinding technique in the continual leakage eCK model is to partition the private key into two components while the leakage of two components is independent each other. After two current components are involved to construct a session key, they must be refreshed to become two new components of the private key for reuse. By the key refreshing method, the proposed LR-AKA protocol possesses overall unbounded leakage property because an adversary can only learn fractional leakage information of two current components. In the generic bilinear group (GBG) model, we demonstrated that our LR-AKA protocol is provably secure in the continual leakage eCK model. In addition, performance analysis demonstrated that the proposed LR-AKA protocol is suitable for both mobile device and PC. A more challenging issue is to propose a LR-AKA protocol without random oracle model while possessing overall unbounded leakage property in the continual leakage eCK model.

APPENDIX

PROOF OF LEMMA 5

Proof: By assumption, A may obtain the ephemeral secret keys x and y of both Π_i^s and Π_j^t by issuing *Ephemeral-secret-leakage* query, but not the private key pairs $(SA_{i,1}, SA_{i,2})$ and $(XA_{i,1}, XA_{i,2})$ of Π_i^s, or the private key pairs $(SB_{j,1}, SB_{j,2})$ and $(XB_{j,1}, XB_{j,2})$ of Π_j^t. In this case, A can compute $KA_{i,1}(= KB_{j,2})$, $KA_{i,2}(= KB_{j,1})$ and $KA_{i,3}(= KB_{j,3})$. Hence, to obtain the session key, A must be able to compute $KA_{i,4} = (PB^{XA_{i,1}})^{XA_{i,2}}$ or $KB_{j,4} = (PA^{XB_{j,1}})^{XB_{j,2}}$ from the instance $(G, G_T, \hat{e}, p, g, PA = \hat{e}(g,g)^{XA}, PB = \hat{e}(g,g)^{XB})$. In the following, we prove that the advantage that A wins the game G_{CL-eCK} is negligibly small, based on the generic bilinear group model. In the generic bilinear group model, the elements of a group are encoded with bit strings. In order for A to perform the multiplication operation on G, the multiplication operation on G_T and the bilinear pairing operation $\hat{e}$, the adversary A should be able to issue associated group queries (oracles). Hence, in the following, we modify the game G_{CL-eCK} defined in Section 3.2 by adding these three group queries Q_G, Q_T and Q_P.

- *Initial Setup*: In this phase, the challenger B first builds several lists by performing the following steps:
  1. The challenger B builds two lists L_G and L_T to record pairs of elements in the groups G and G_T, respectively.
     - The list L_G consists of elements of the form $(PG_{m,n,r}, \psi G_{m,n,r})$. Each $PG_{m,n,r}$ is a multivariate polynomial that consists of a finite numbers of variates in G with coefficients in Z_p. For a given $PG_{m,n,r}$, B uses a bit string $\psi G_{m,n,r}$ to communicate with A. Here, the indices m, n and r indicate, respectively, the type of query, n-th query, and the r-th element in G/G_T. Two tuples $(g, \psi G_{I,1,1})$ and $(h, \psi G_{I,1,2})$ are initially added in the list L_G.
     - The list L_T consists of elements of the form $(PT_{m,n,r}, \psi T_{m,n,r})$ and records the elements of G_T. The indices m, n and r have the same meanings as above. $PT_{m,n,r}$ is a multivariate polynomial with coefficients in Z_p and variates in G or G_T. For a given $PT_{m,n,r}$, B uses the bit string $\psi T_{m,n,r}$ to communicate with A. It is worth mentioning that all the $PG_{m,n,r}$ in L_G and $PT_{m,n,r}$ in L_T are different multivariate polynomials. In addition, all the bit strings $\psi G_{m,n,r}$ and $\psi T_{m,n,r}$ are distinct bit strings. In the *Query* phase described later, the challenger B adopts two rules to update two lists L_G and L_T as below.

     (1) When recording a multivariate polynomial $PG_{m,n,r}/PT_{m,n,r}$ in L_G/L_T, B first checks whether or not $PG_{m,n,r}/PT_{m,n,r}$ has been recorded in L_G/L_T. If so, B obtains the corresponding bit string $\psi G_{m,n,r}/\psi T_{m,n,r}$. Otherwise, B randomly chooses a new bit string $\psi G_{m,n,r}/\psi T_{m,n,r}$ in L_G and L_T. B records $(PG_{m,n,r}, \psi G_{m,n,r})/(PT_{m,n,r}, \psi T_{m,n,r})$ in L_G/L_T.

     (2) When recording a bit string $\psi G_{m,n,r}/\psi T_{m,n,r}$ in L_G, B first checks whether or not $\psi G_{m,n,r}/\psi T_{m,n,r}$ has been recorded before. If so, B obtains the corresponding multivari-

ate polynomial $PG_{m,n,r}/PT_{m,n,r}$. Otherwise, B chooses a new variate $VG_{m,n,r}/VT_{m,n,r}$ and records $(VG_{m,n,r}, \psi G_{m,n,r})/(VT_{m,n,r}, \psi T_{m,n,r})$ in L_G/L_T.

2. The challenger B constructs a list L_K to record the user's private key pairs and public keys. More precisely, the elements in L_K are of the form (ID, SA, XA, TA, PA), where ID is in Z_p and SA, XA, TA, PA are multivariate polynomials recorded in L_G and L_T. And B generates and adds l honest users in L_K and sends these user's (ID, TA, PA) to the adversary A. The key generation of a user is described as below:
   - B randomly chooses the user's identity $ID_i \in Z_p^*$.
   - B picks a new variable T_i, and computes four multivariate polynomials:
   (1) $PSA_i = T_i \cdot g$.
   (2) $PXA_i = T_i \cdot g \cdot h$.
   (3) $PTA_i = T_i \cdot g \cdot g$.
   (4) $PPA_i = g \cdot g \cdot T_i \cdot g \cdot h$.
   - B records $(ID_i, PSA_i, PXA_i, PTA_i, PPA_i)$ in L_K.
   - B records T_i and PSA_i in L_G, and PXA_i, PTA_i and PPA_i in L_T.
3. The challenger B builds a list REC to record the details of the sessions. For each oracle Π_i^s, B records the session in the form $(\psi PNID_{i,s}, \psi PNPK_{i,s}, ES_{i,s}, SK_{i,s}, \psi SM_{i,s}, PSM_{i,s}, \psi RM_{i,s}, PRM_{i,s})$, where
   - $\psi PNID_{i,s}$: The partner's identity.
   - $ES_{i,s}$: The ephemeral key of Π_i^s.
   - $SK_{i,s}$: The session key of this session.
   - $\psi SM_{i,s}$: The transcript (bit string) of message sent by Π_i^s.
   - $PSM_{i,s}$: The multivariate polynomial of $\psi SM_{i,s}$.
   - $\psi RM_{i,s}$: The transcript (bit string) of message sent by partner.
   - $PRM_{i,s}$: The multivariate polynomial of $\psi RM_{i,s}$.
4. At the end of this phase, the challenger B sends the public parameters PP to A (using bit strings).

- *Query*: In this phase, A can issue the following queries adaptively for totally at most q times.
  - *Group query* Q_G $(\psi G_{Q,i,1}, \psi G_{Q,i,2},$ operation): When A issues the i-th group query Q_G with two bit strings $(\psi G_{Q,i,1}, \psi G_{Q,i,2})$ and an operation (multiplication or division), the challenger B performs the following steps.
  (1) B first records $\psi G_{Q,i,1}$ and $\psi G_{Q,i,2}$ in L_G and obtains the corresponding multivariate polynomials $PG_{Q,i,1}$ and $PG_{Q,i,2}$.
  (2) B computes and sets the polynomial $PG_{Q,i,3} = PG_{Q,i,1} + PG_{Q,i,2}$ if the operation is multiplication, or $PG_{Q,i,3} = PG_{Q,i,1} - PG_{Q,i,2}$ if the operation is division.
  (3) B records $PG_{Q,i,3}$ in L_G and obtain the corresponding bit string $\psi G_{Q,i,3}$. Finally, B returns $\psi G_{Q,i,3}$ to A.
  - *Group query* $Q_T(\psi T_{Q,i,1}, \psi T_{Q,i,2},$ operation): When A issues the i-th group query Q_T with two bit strings $(\psi T_{Q,i,1}, \psi T_{Q,i,2})$ and an operation (multiplication or division). Then, with respect to the group G_T and the list L_T, B performs similar steps in the Group query Q_G above. B finally returns $\psi T_{Q,i,3}$ to A.
  - *Pairing query* $Q_P(\psi G_{P,i,1}, \psi G_{P,i,2})$: For the i-th pairing query Q_P, by taking as input two bit strings $\psi G_{P,i,1}$ and $\psi G_{P,i,2}$, B performs the following steps:
  (1) B first records two bit strings $\psi G_{P,i,1}$ and $\psi G_{P,i,2}$ in L_G and obtains the corresponding multivariate polynomials $PG_{P,i,1}$ and $PG_{P,i,2}$
  (2) B computes the polynomial $PT_{P,i,1} = PG_{P,i,1} \cdot PG_{P,i,2}$.
  (3) B records $PT_{P,i,1}$ in L_T, obtains the corresponding bit string $\psi T_{P,i,1}$, and returns $\psi T_{P,i,1}$ to A.
  - *Send*$(\Pi_i^s, \psi RM_{i,s}, \psi PNID_{i,s}, \psi PNPK_{i,s})$: A can issue the *Send* query with an oracle Π_i^s, the partner's information $(\psi PNID_{i,s}, \psi PNPK_{i,s})$ and the message $\psi RM_{i,s}$. B first checks whether or not Π_i^s is recorded in the list REC. If so, then B returns $\psi SM_{i,s}$ to A provided that the other inputs are the same with Π_i^s in REC; otherwise, B returns "false" to A. If Π_i^s is not recorded in the list REC, B performs the following steps:
  (1) B creates a new record $(\Pi_i^s, \psi RM_{i,s}, \psi PNID_{i,s}, \psi PNPK_{i,s})$ of Π_i^s in REC.
  (2) B randomly chooses an ephemeral secret key $ES_{i,s} \in Z_p^*$ and sets $PSM_{i,s} = ES_{i,s} \cdot g$.
  (3) B stores $PSM_{i,s}$ in L_G and obtains the corresponding bit string $\psi SM_{i,s}$.
  (4) B stores $PSM_{i,s}$ and $\psi SM_{i,s}$ in the record of Π_i^s in the list REC.
  (5) B returns $\psi SM_{i,s}$ to A.
  - *Reveal*(Π_i^s): A can issue this query to obtain the session key $SK_{i,s}$ of Π_i^s. Upon receiving the *Reveal* query, B first checks whether or not Π_i^s has been recorded in the list REC. If Π_i^s is not recorded in REC, B returns "false" to A. Otherwise, B runs the following steps:
  (1) B first obtains $(\psi RM_{i,s}, \psi PNID_{i,s}, \psi PNPK_{i,s}, ES_{i,s}, PSM_{i,s})$ by searching Π_i^s in REC.
  (2) B searches $\psi PNID_{i,s}$ in L_K to obtains two multivariate polynomials $PPNTA$ and $PPNPA$.
  (3) B searches $\psi RM_{i,s}$ in L_G and obtains the multivariate polynomial $PRM_{i,s}$.
  (4) B obtains the multivariate polynomials of private key pair (PSA_i, PXA_i) by searching ID_i in L_K.
  (5) B sets a new variable $TES_{i,s}$ to represent $ES_{i,s}$.
  (6) B computes four multivariate polynomials:
    - $K_1 = TES_{i,s} \cdot PPNTA$.
    - $K_2 = PRM_{i,s} \cdot PSA_i$
    - $K_3 = TES_{i,s} \cdot PRM_{i,s}$.

- $K_4 = PPNPA \cdot PXA_i$.

(7) B records K_3 in L_G, and records K_1, K_2 and K_4 in L_T. Also, B obtains the corresponding bit strings ψK_1, ψK_2, ψK_3 and ψK_4.

(8) B sets the session key $SK_{i,s} = \psi K_1 \oplus \psi K_2 \oplus \psi K_3 \oplus \psi K_4$ in the tuple of Π_i^s of the list REC.

- *Ephemeral-secret-leakage*(Π_i^s): Upon receiving the Reveal query along with Π_i^s, B checks whether or not Π_i^s has been recorded in REC. If so, B returns $ES_{i,s}$ to A. Otherwise, B returns "false" to A.
- *Corrupt*(ID_i): Upon receiving the *Corrupt* query along with ID_i, B first checks whether or not ID_i has been recorded in L_K. If not, B returns "false" to A. Otherwise, B can obtain (ID_i, PSA_i, PXA_i, PTA_i, PPA_i) from L_K, and returns the corresponding bit strings of (PSA_i, PXA_i) to A.
- *Leak query*($f_{i,s}$, $h_{i,s}$, Π_i^s): Upon receiving the *Leak* query along with $f_{i,s}$, $h_{i,s}$ and Π_i^s, B computes and sends the fractional leakage information ($\Lambda f_{i,s}$, $\Lambda h_{i,s}$) to A, where $\Lambda f_{i,s} = f_{i,s}$ ($SA_{i-1,1}$, $SA_{i-1,2}$, $XA_{i-1,1}$, $XA_{i-1,2}$, α_i) and $\Lambda h_{i,s} = h_{i,s}$ ($SA_{i,2}$, $SA_{i,2}$, $XA_{i,2}$, $XA_{i,2}$, $ES_{i,s}$), where $ES_{i,s}$ denotes the random value x or y. Note that two leakage functions must satisfy $|f_{i,s}| \leq \lambda$ and $|h_{i,s}| \leq \lambda$.
- *Test*(Π_i^s): When A issues this query, the challenger B first checks whether or not Π_i^s has been record in REC. If not, B returns "false" to A. Also, if $SK_{i,s}$ has been set in REC, B also returns "false" to A. Otherwise, B performs the following steps:

(1) B issues the *Reveal*(Π_i^s) to obtain the session key $SK_{i,s}$.

(2) B flips an unbiased coin bit $cb \in 0, 1$. If $cb = 1$, B returns $SK_{i,s}$ to A; otherwise, B returns a random bit string to A.

Now we analyze the advantage of the adversary A winning the game.

(1) Let us discuss the numbers of elements added in of L_G and L_T after all kinds of queries.

- For each query of Q_G, Q_T or Q_P, at most 3 elements are recorded in L_G or L_T.
- For each *Send* query, at most 1 new element is recorded in L_G or L_T.
- For each *Reveal* query, at most 5 new elements are recorded in L_G or L_T.
- For each *Ephemeral-secret-leakage* or *Corrupt* query, no new element recorded.
- For each *Test* query, at most 5 new elements are recorded in L_G or L_T.

Let q_O denote the total number of group queries Q_G, Q_T and Q_P, and let q_S, q_R and q_T, denote the numbers of *Send*, *Reveal* and *Test* queries, respectively. As before, $|L_G|$ and $|L_T|$ denote, respectively, the numbers of elements in L_G and L_T. Then, we have $|L_G| + |L_T| \leq 2 + 3q_O + q_S + 5q_R + 5 \leq 5q$.

(2) Let us discuss the degrees of polynomials in L_G.

- Every polynomial of new variates $VG_{m,n,r}$, T_i and $TES_{i,s}$ in L_G has degree 1.
- Every polynomial of the private key PSA_i has degree 2.
- In the *Reveal* query, K_3 and $PrM_{i,s}$ have the same degree.
- In Q_G, the degree of $PG_{Q,i,3}$ is equal to the maximal degree of $PG_{Q,i,1}$ and $PG_{Q,i,2}$, since the polynomial $PG_{Q,i,3} = PG_{Q,i,1} + PG_{Q,i,2}$.

Hence, every polynomial in L_G has degree at most 2.

(3) Let us discuss the degrees of polynomials in L_T.

- Every polynomial of new variates $VT_{m,n,r}$ in L_T has degree 1.
- Every polynomial of the private key $PXA_i = T_i \cdot g \cdot h$ has degree 3.
- The polynomials of the private key pairs PTA_i and PPA_i have degrees 3 and 5, respectively.
- In Q_P, each polynomial $PT_{P,i,k}$ has degree at most 4 since each polynomial in L_G has degree at most 2.
- In the Reveal query, the degree of $K_1 = TES_{i,s} \cdot PPNTA$ is 4, the degree of $K_2 = PrM_{i,s} \cdot PSA_i$ is 4, and the degree of $K_4 = PPNPA \cdot PXA_i$ is 8.
- In Q_T, the degree of $PT_{Q,i,3}$ is equal to the maximal degree of $PT_{Q,i,1}$ and $PT_{Q,i,2}$ since the polynomial $PT_{Q,i,3} = PT_{Q,i,1} + PT_{Q,i,2}$.

Hence, every polynomial in L_T has degree at most 8.

Next, for each variable in L_G and L_T, the challenger B chooses a random value in Z_p^*, denoted by $v_1, v_2, \ldots, v_n$. The adversary A is said to win the game G_{CL-eCK} if one of the following two cases occurs:

- *Case 1:* The adversary A finds a collision in G or G_T:
  - ◇ There exist two polynomials PG_i and PG_j in L_G such that $PG_i(v_1, v_2, \ldots, v_n) = PG_j(v_1, v_2, \ldots, v_n)$.
  - ◇ There exist two polynomials PT_i and PT_j in L_T such that $PT_i(v_1, v_2, \ldots, v_n) = PT_j(v_1, v_2, \ldots, v_n)$.
- *Case 2:* The adversary A outputs $cb' = cb$ after the *Test* query.

In the real adversary game, the advantage of the adversary A in the simulated game G_{CL-eCK} is an upper bound of the success probability of A. For convenience, we first compute A's success probability under the situation that A cannot issue the *Leak* query in the query phase. Afterwards, we discuss the other situation.

- Without the *Leak* query: If A does not use the *Leak* query, the success probability can be computed by the following two cases:
  - *Case 1:* The adversary A can find a collision in G or G_T. In this case, A can resolve the discrete logarithm problem in G or G_T. Assume that PG_i and PG_j are two distinct polynomials in L_G. We compute the probability of the event when the polynomials $PG_C = PG_i - PG_j$ is a zero polynomial.

By applying Lemma 2 with $\lambda = 0$ in Section 2, the probability of $PG_C(v_1, v_2, \ldots, v_n) = 0$ in Z_p^* is at most $2/p$. Since there are $\binom{|L_G|}{2}$ different pairs (PG_i, PG_j) in L_G, the probability that Case 1 occurs is at most $(2/p)\binom{|L_G|}{2}$. Similarly, since every polynomial in L_T has degree at most 8, the collision probability in L_T is at most $(8/p)\binom{|L_T|}{2}$.

- *Case 2:* The adversary A cannot find any collision in G or G_T. In this case, A's view in the game G_{CL-eCK} is identical to that in the real game. This means that A does not obtain useful information for guessing a coin bit cb in the game G_{CL-eCK}. Hence, the success probability that A outputs a correct coin bit $cb' = cb$ is $1/2$ on average.

Now, let's evaluate the success probability Pr_{A-wol} that A wins the game G_{CL-eCK} without the *Leak* query. By the discussion above, the probability for Case 1 to occur satisfies the inequality

$$\begin{aligned} Pr[Case1] &\leq [(2/p)\binom{|L_G|}{2} + (8/p)\binom{|L_T|}{2}] \\ &\leq (8/p)(|L_G| + |L_T|)^2 \leq 200q^2/p. \end{aligned}$$

On the other hand, in Case 2, A has probability $1/2$ to output a correct coin bit. Therefore, the success probability Pr_{A-wol} satisfies

$$\begin{aligned} Pr_{A-wol} &\leq Pr[Case1] + Pr[Case2] \\ &\leq 200q2/p + (1/2). \end{aligned}$$

The advantage that A wins the game G_{CL-eCK} without issuing the Leak query is
$Adv_{A-wol} \leq |200q^2/p + (1/2) - (1/2)| = 200q^2/p$.
Obviously, Adv_{A-wol} is negligible if $q =$poly$(logp)$.

- With the *Leak* query: Note that, under Situation 2 described in the proof of Theorem 1, A is allowed to issue the *Ephemeral-secret-leakage*(Π_i^s) and *Ephemeral-secret-leakage*(Π_j^t), but not the *Corrupt*(ID_i) or *Corrupt*(ID_j) during the query phase. Two fractional leakage information $\Lambda f_{i,s}$ and $\Lambda h_{i,s}$, respectively, are used to represent the outputs of two leakage functions $f_{i,s}$ and $h_{i,s}$. By $\Lambda f_{i,s}$ and $\Lambda h_{i,s}$, the leaked information about $f_{i,s}(SA_{i-1,1}, SA_{i-1,2}, XA_{i-1,1}, XA_{i-1,2}, \alpha_i)$ and $h_{i,s}(SA_{i,2}, SA_{i,2}, XA_{i,2}, XA_{i,2}, ES_{i,s})$ are discussed below:
  - α_i: The random value α_i is used to generate the next private key pairs. Since the value α_i is randomly chosen for each key refreshing, the λ bits of α_i is useless for the next session.
  - $ES_{i,s}$: The ephemeral secret key $ES_{i,s}$ denotes the random value x or y in the oracle Π_i^s and can only be leaked once. Since A can obtain the ephemeral secret keys completely by issuing *Ephemeral-secret-leakage*(Π_i^s) and *Ephemeral-secret-leakage*(Π_j^t), the leakage information of $ES_{i,s}$ is useless.
  - $(SA_{i-1,1}, SA_{i,1}, SA_{i-1,2}, SA_{i,2})$: The user's first private key satisfies the equality $SA = SA_{i-1,1} \cdot SA_{i-1,2} = SA_{i,1} \cdot SA_{i,2}$. And the leakage information of $SA_{i,1}$ and $SA_{i,2}$ is independent of that of $SA_{i-1,1}$ and $SA_{i-1,2}$. Thus, A can learn at most λ bits of SA.
  - $(XA_{i-1,1}, XA_{i,1}, XA_{i-1,2}, XA_{i,2})$: The user's second private key satisfies the equality $XA = XA_{i-1,1} \cdot XA_{i-1,2} = XA_{i,1} \cdot XA_{i,2}$. And the leakage information of $XA_{i,1}$ and $XA_{i,2}$ is independent of that of $XA_{i-1,1}$ and $XA_{i-1,2}$. Thus, A can learn at most λ bits of XA.

Now, let us discuss the successful probability Pr_{A-wl} that A wins the game G_{CL-eCK} with the *Leak* query. Indeed, in Situation 2, A can obtain the ephemeral secret keys. Then, by getting a participant's private keys SA and XA, the adversary A can always output a correct coin bit. To evaluate the successful probability Pr_{A-wl}, we define two events as follows.

(1) The event EP denotes that A may obtain SA and XA completely from the fractional leakage information $\Lambda f_{i,s}$ and $\Lambda h_{i,s}$. In addition, the events $\overline{EP}$ denote the complement events of EP.

(2) The event EC denotes that A may output a correct coin bit $cb' = cb$. The successful probability Pr_{A-wl} that A wins the game G_{CL-eCK} with the *Leak* query is bounded as below.

$$\begin{aligned} Pr_{A-wl} &= Pr[EC] \\ &= Pr[EC \cap EP] + Pr[EC \cap \overline{EP}] \\ &= Pr[EC \cap EP] + Pr[EC|\overline{EP}] \cdot Pr[\overline{EP}]. \end{aligned}$$

Since $Pr[EC \cap EP] \leq Pr[EP]$ and $Pr[\overline{EP}] = 1 - Pr[EP]$, we obtain $Pr_{A-wl} \leq Pr[EP] + Pr[EC|\overline{EP}] \cdot (1 - Pr[EP])$. Under the condition, A can't obtain any useful information to output a correct coin bit. Hence, $Pr[EC|\overline{EP}]$ is $1/2$ on average. Thus,

$$\begin{aligned} Pr_{A-wl} &\leq Pr[EP] + (1/2) \cdot (1 - Pr[EP]) \\ &= 1/2(1 + Pr[EP]). \end{aligned}$$

Therefore, the advantage that A wins the game G_{CL-eCK} with the *Leak* query is $\text{Adv}_{A-wl} \leq |Pr_{A-wl} - 1/2| = (1/2)Pr[EP]$. Since the advantage Adv_{A-wol} that A wins the game G_{CL-eCK} without the *Leak* query is $Adv_{A-wol} \leq 100q^2/p = O(q^2/p)$, the advantage that A wins the game G_{CL-eCK} with issuing the *Leak* query $Adv_{A-wl} \leq O((q^2/p) * 2^{\lambda})$ because A can learn at most λ bits of the private keys SA and XA. By Lemma 2, if $\lambda \leq logp - \omega(loglogp)$, we say that the proposed LR-AKA protocol is secure in the continual leakage model. Q.E.D.

REFERENCES

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology*, vol. 1666. Berlin, Germany: Springer-Verlag, 1999, pp. 388–397.

[2] D. Boneh, R.-A. Demillo, and R.-J. Lipton, "On the importance of checking cryptographic protocols for faults," in *Advances in Cryptology—EUROCRYPT*, vol. 1233. Heidelberg, Germany: Springer, 1997, pp. 37–51.

[3] E. Biham, Y. Carmeli, and A. Shamir, "Bug attacks," in *Advances in Cryptology—CRYPTO*, vol. 5157. Heidelberg, Germany: Springer, 2008, pp. 221–240.

[4] P. C. Kocher, "Timing attacks on implementations of Diffie–Hellman, RSA, DSS, and other systems," in *Advances in Cryptology—CRYPTO*, vol. 1109. Heidelberg, Germany: Springer, 1996, pp. 104–113.

[5] D. Brumley and D. Boneh, "Remote timing attacks are practical," *Comput. Netw.*, vol. 48, no. 5, pp. 701–716, 2005.

[6] J. Katz and V. Vaikuntanathan, "Signature schemes with bounded leakage resilience," in *Advances in Cryptology—ASIACRYPT*, vol. 5912. Heidelberg, Germany: Springer, 2009, pp. 703–720.

[7] S. Faust, E. Kiltz, K. Pietrzak, and G. Rothblum, "Leakage-resilient signatures," in *Proc. Theory Cryptogr. Conf. (TCC)*, vol. 5978. Heidelberg, Germany, 2010, pp. 343–360.

[8] S. Faust, C. Hazay, J.-B. Nielsen, P. S. Nordholt, and A. Zottarel, "Signature schemes secure against hard-to-invert leakage," in *Advances in Cryptology—ASIACRYPT*, vol. 7658. Heidelberg, Germany: Springer, 2012, pp. 98–115.

[9] D. Galindo and S. Vivek, "A practical leakage-resilient signature scheme in the generic group model," in *Proc. Int. Conf. Sel. Areas Cryptogr. (SAC)*, vol. 7707. Heidelberg, Germany, 2013, pp. 50–65.

[10] F. Tang, H. Li, Q. Niu, and B. Liang, "Efficient leakage-resilient signature schemes in the generic bilinear group model," in *Proc. Int. Conf. Inf. Secur. Pract. Exper. (ISPEC)*, vol. 8434. Heidelberg, Germany, 2014, pp. 418–432.

[11] M. Naor and G. Segev, "Public-key cryptosystems resilient to key leakage," in *Advances in Cryptology—CRYPTO*, vol. 5677. Heidelberg, Germany: Springer, 2009, pp. 18–35.

[12] J. Alwen, Y. Dodis, M. Naor, G. Segev, S. Walfish, and D. Wichs, "Public-key encryption in the bounded-retrieval model," in *Advances in Cryptology—EUROCRYPT*, vol. 6110. Heidelberg, Germany: Springer, 2010, pp. 113–134.

[13] E. Kiltz and K. Pietrzak, "Leakage resilient ElGamal encryption," in *Advances in Cryptology—ASIACRYPT*, vol. 6477. Heidelberg, Germany: Springer, 2010, pp. 595–612.

[14] S. Halevi and H. Lin, "After-the-fact leakage in public-key encryption," in *Proc. Theory Cryptogr. Conf. (TCC)*, vol. 6597. Heidelberg, Germany, 2011, pp. 107–124.

[15] D. Galindo, J. Großschädl, Z. Liu, P. K. Vadnala, and S. Vivek, "Implementation of a leakage-resilient ElGamal key encapsulation mechanism," *J. Cryptogr. Eng.*, vol. 6, no. 3, pp. 229–238, 2016.

[16] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Advances in Cryptology—EUROCRYPT*, vol. 1807. Heidelberg, Germany: Springer, 2000, pp. 139–155.

[17] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Advances in Cryptology—EUROCRYPT*, vol. 2045. Heidelberg, Germany: Springer, 2011, pp. 453–474.

[18] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *Proc. Int. Conf. Provable Secur. (PROVSEC)*, vol. 4784. Heidelberg, Germany, 2007, pp. 1–16.

[19] T. Okamoto, "Authenticated key exchange and key encapsulation in the standard model," in *Advances in Cryptology—ASIACRYPT*, vol. 4833. Heidelberg, Germany: Springer, 2007, pp. 474–484.

[20] M. Kim, A. Fujioka, and B. Ustaoğlu, "Strongly secure authenticated key exchange without NAXOS' approach," in *Proc. Int. Workshop Secur. (IWSEC)*, vol. 5824. Heidelberg, Germany, 2009, pp. 174–191.

[21] T.-Y. Wu and Y.-M. Tseng, "An ID-based mutual authentication and key exchange protocol for low-power mobile devices," *Comput. J.*, vol. 53, no. 7, pp. 1062–1070, 2010.

[22] Y.-M. Tseng, S.-S. Huang, T.-T. Tsai, and J.-H. Ke, "List-free ID-based mutual authentication and key agreement protocol for multiserver architectures," *IEEE Trans. Emerg. Topics Comput.*, vol. 4, no. 1, pp. 102–112, Jan./Mar. 2016.

[23] J. Alwen, Y. Dodis, and D. Wichs, "Leakage-resilient public-key cryptography in the bounded-retrieval model," in *Advances in Cryptology—CRYPTO*, vol. 5677. Heidelberg, Germany: Springer, 2009, pp. 36–54.

[24] Y. Dodis, K. Haralambiev, A. López-Alt, and D. Wichs, "Efficient public-key cryptography in the presence of key leakage," in *Advances in Cryptology—ASIACRYPT*, vol. 6477. Heidelberg, Germany: Springer, 2010, pp. 613–631.

[25] G. Yang, Y. Mu, W. Susilo, and D.-S. Wong, "Leakage resilient authenticated key exchange secure in the auxiliary input model," in *Proc. Int. Conf. Inf. Secur. Pract. Exper. (ISPEC)*, 2013, pp. 204–217.

[26] D. Moriyama and T. Okamoto, "Leakage resilient eCK-secure key exchange protocol without random oracles," in *Proc. 6th ACM Symp. Inf., Comput. Commun. Secur.*, 2011, pp. 441–447.

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

[28] J. Alawatugoda, C. Boyd, and D. Stebila, "Continuous after-the-fact leakage-resilient key exchange," in *Proc. Austral. Conf. Inf. Secur. Privacy (ACISP)*, vol. 8544. Heidelberg, Germany, 2014, pp. 258–273.

[29] J. Alawatugoda, D. Stebila, and C. Boyd, "Modelling after-the-fact leakage for key exchange," in *Proc. 9th ACM Symp. Inf., Comput. Commun. Secur.*, 2014, pp. 207–216.

[30] J. Alawatugoda, D. Stebila, and C. Boyd, "Continuous after-the-fact leakage-resilient eCK-secure key exchange," in *Proc. IMA Int. Conf. Cryptogr. Coding*, 2015, pp. 277–294.

[31] S. Dziembowski and S. Faust, "Leakage-resilient cryptography from the inner-product extractor," in *Advances in Cryptology—ASIACRYPT*, vol. 7073. Heidelberg, Germany: Springer, 2011, pp. 702–721.

[32] J.-D. Wu, Y.-M. Tseng, and S.-S. Huang, "Leakage-resilient ID-based signature scheme in the generic bilinear group model," *Secur. Commun. Netw.*, vol. 9, no. 17, pp. 3987–4001, 2016.

[33] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Advances in Cryptology—EUROCRYPT*, vol. 3494. Heidelberg, Germany: Springer, 2005, pp. 440–456.

[34] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology—CRYPTO*, vol. 2139. Heidelberg, Germany: Springer, 2001, pp. 213–229.

[35] M. Scott, "On the efficient implementation of pairing-based protocols," in *Cryptography and Coding*, vol. 7089. Heidelberg, Germany: Springer, 2011, pp. 296–308.

[36] V. Shoup, "Lower bounds for discrete logarithms and related problems," in *Advances in Cryptology—EUROCRYPT*, vol. 1233. Heidelberg, Germany: Springer, 1997, pp. 256–266.

[37] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.

[38] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.

[39] R. Zippel, "Probabilistic algorithms for sparse polynomials," in *Proc. Int. Symp. Symbolic Algebraic Manipulation (EUROSAM)*, vol. 72. Heidelberg, Germany, 1979, pp. 216–226.

[40] J. T. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *J. ACM*, vol. 27, no. 4, pp. 701–717, 1980.

[41] Y.-M. Tseng and T.-T. Tsai, "Efficient revocable ID-based encryption with a public channel," *Comput. J.*, vol. 55, no. 4, pp. 475–486, 2012.

[42] H. Xiong and Z. Qin, "Revocable and scalable certificateless remote authentication protocol with anonymity for wireless body area networks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 7, pp. 1442–1455, Jul. 2015.

[43] H. Xiong, S. Wu, J. Geng, E. Ahene, S. Wu, and Z. Qin, "A pairing-free key-insulated certificate-based signature scheme with provable security," *KSII Trans. Internet Inf. Syst.*, vol. 9, no. 3, pp. 1246–1259, 2015.

JUI-DI WU received the B.S. and M.S. degrees from the Department of Mathematics, National Changhua University of Education, Taiwan, in 2006 and 2008, respectively, where he is currently pursuing the Ph.D. degree. His research interests include leakage-resilient cryptography and pairing-based cryptography.

YUH-MIN TSENG received the B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, in 1988, the M.S. degree from National Taiwan University, Taipei, Taiwan, in 1990, and the Ph.D. degree from National Chung Hsing University, Taichung, Taiwan, in 1999. He is currently a Professor with the Department of Mathematics, National Changhua University of Education, Changhua, Taiwan. He has authored over 100 scientific journal and conference papers on various research areas of cryptography, security, and computer network. His research interests include cryptography, network security, computer network, and mobile communications. He is a member of the IEEE Computer Society, the IEEE Communications Society, and the Chinese Cryptology and Information Security Association. In 2006, he was a recipient of the Wilkes Award from the British Computer Society. He serves as an Editor of several international journals.

SEN-SHAN HUANG received the Ph.D. degree from the University of Illinois at Urbana–Champaign under the supervision of Professor Bruce C. Berndt. He is currently a Professor with the Department of Mathematics, National Changhua University of Education, Taiwan. His research interests include number theory, cryptography, and network security.