After-the-Fact Leakage-Resilient Identity-Based Authenticated Key Exchange

Ou Ruan, Yuanyuan Zhang, Mingwu Zhang, Jing Zhou, and Lein Harn

***Abstract*—Authenticated key exchange (AKE) scheme is one of the most widely used cryptographic primitives in practice, even in the Internet-of-Things (IoT) environments. In order to resist side-channel attacks, several works have been proposed for defining leakage-resilient (LR) security models and constructing LR-AKE protocols. However, all these LR-AKE schemes employed the traditional X.509 certificate-based public-key infrastructure authentication framework, where the online transmission and verification of the public-key certificate are the major drawbacks. In this paper, we first propose a general framework for constructing identity-based AKE protocols in the bounded after-the-fact LR extended-Canetti–Krawczyk security model, and show a formal proof in the standard model. Our proposed scheme offers a flexible approach to simplify the certificate management. Moreover, our result could be extended to the bounded-retrieval model, yielding the first LR-AKE protocol in this model.**

***Index Terms*—Authenticated key exchange, after-the-fact, bounded-retrieval model, eCK security, identity-based system, leakage-resilient (LR).**

I. Introduction

With the development of Internet-of-Things (IoT), there is a big success of smart homes, connected cars, Industrie 4.0 factories, and so on. In these environments, the IoT nodes will collect, handle, record, and transfer lots of sensitive data. However, many IoT nodes are exposed to the public insecure environments, and are very vulnerable to side-channel attacks [1]. Thus, an IoT attacker can overcome the security protecting by executing side-channel attacks. Compared to the traditional mathematical attacks, side-channel attacks have many advantages: 1) requiring much less time and trouble; 2) no need of expensive equipment; and 3) being very difficult to detect.

In order to resist side-channel attacks, leakage-resilient (LR) cryptography has been introduced and studied, such as LR encryption [2]–[8], signature [9]–[12], pseudo-random function [13], and multiparty secure computation [14], [15]. As one of the most widely used cryptographic primitives, it is important to design and analyze the LR authenticated key exchange (AKE) protocols. However, there are only a few works for defining LR security models and constructing LR-AKE protocols. With LR-AKE protocols, several parties can derive a common cryptographically strong session key over an insecure network, where the adversary may execute leakage attacks and learn some partial information of parties' holding secrets such as their private keys.

Earlier AKE security models include the Bellare–Rogaway (BR) [16], Canetti–Krawczyk (CK) [17], and extended Canetti–Krawczyk (eCK) [18] models. In these models, side-channel attacks are not allowed, and the adversary could not get any information of secret values. However, side-channel attacks [19] exist in many practical environments. For example, an attacker of IoT can get partial secret information of nodes' long-term secret keys by measuring their electromagnetic radiation. Thus, modeling and designing LR-AKE protocols are very important.

Moriyama and Okamoto (MO) [20] presented the first formalization of λ-LR eCK security model for AKE protocol. The central limitation of the MO model is that leakage attacks are not allowed after the test session is selected by the adversary. Alawatugoda *et al.* [21] first introduced after-the-fact LR (AFLR) security model and presented a continuous AFLR (CAFLR) AKE protocol. The proposed protocol was only secure in the CK security model, that is a weaker variant of the generic eCK model. Alawatugoda *et al.* [22] introduced a generic AFLR security model for AKE protocols in the eCK security model, and constructed a bounded AFLR (BAFLR) eCK-secure AKE protocol. Alawatugoda *et al.* [23] proposed the first concrete construction of CAFLR eCK-secure AKE protocol. Chen *et al.* [24] introduced a strong AFLR eCK security model, which not only captured the leakage attacks on a long-term secret private key, but also considered the leakage of ephemeral secret randomness.

All the above LR-AKE schemes employed the traditional X.509 certificate-based public-key infrastructure (PKI) authentication framework. In such schemes, a major drawback is that it requires a lot of bandwidth and authentication time, because of the online transmission and verification of public-key certificates. The identity (ID)-based public-key system gives a good solution to this problem, where public-key certificates are no longer needed, since public keys are derived directly from the ID information. Thus, an ID-based LR-AKE protocol will be preferred to that under the employment of the traditional PKI. In this paper, our goal is to propose an ID-based AKE protocol in the AFLR eCK security model.

Manuscript received August 8, 2016; revised November 24, 2016 and February 4, 2017; accepted March 17, 2017. This work was supported by the Educational Commission of Hubei Province of China under Grant D20151401. *(Corresponding author: Yuanyuan Zhang.)*

O. Ruan, Y. Zhang, M. Zhang, and J. Zhou are with the School of Computer Science and Technology, Hubei University of Technology, Wuhan 430068, China (e-mail: ruanou@163.com; circle0519@hotmail.com; mzhang@mail.hbut.edu.cn; zhoujinghbut@163.com).

L. Harn is with the School of Computer Science & Technology, Hubei University of Technology, Wuhan 430068, China, and also with the Department of Computer Science and Electrical Engineering, University of Missouri—Kansas City, Kansas City, MO 64110-2499 USA (e-mail: harnlein@gmail.com).

Digital Object Identifier 10.1109/JSYST.2017.2685524

Our Results: Our main contribution is the construction of the first ID-based AKE protocol in the BAFLR eCK security model. The following points are discussed in detail in this paper.

1) We first propose a protocol for ID-based LR-AKE, based on λ-leakage-smooth ID-based hash proof system (IB-HPS) [25] and key derivation function (KDF) [26], and formally prove its BAFLR eCK security in the standard model.
2) Our protocol could be extended to the bounded-retrieval model (BRM), yielding the first LR-AKE protocol in BRM. BRM is a generalization of the relative-leakage model (RLM), where the leakage-parameter λ is an independent parameter and is only related to the secret private key. Thus, λ could be set based on practical considerations about leakage tolerance and may be arbitrary large, while computation time and communication bandwidth still remain small, since they are both independent of λ.
3) We consider a more strong security model, AFLR eCK security model, while most of others only employ other weaker security models.
4) Our protocol only needs one round of communication, while most of others require two rounds of communications.

The rest of this paper is organized as follows. Section II addresses related works; Section III reviews the preliminaries; Section IV describes the security model of the BAFLR eCK-secure ID-based AKE protocol; Section V presents the proposed general framework and analyzes the provable security and the performance; Section VI gives an instantiation of the framework; and finally, Section VII discusses the conclusion and future works.

II. Related Works

Traditional AKE: The BR model [16] first introduced the formal security model for AKE by using indistinguishable games. Cao *et al.* [27] proposed an ID-based AKE protocol and formally proved the security in the modified BR model. Canetti *et al.* gave the CK security model [17] based on the BR model and Bellare–Canetti–Krawczyk model [28]. LaMacchia *et al.* [18] proposed the eCK security model that is an extension of the CK security model. In the eCK model, the adversary is much stronger and he could access either the long-term secret key or the ephemeral secret randomness of the test session. Now, CK/eCK security models are the common models for designing AKE protocols. Xie *et al.* [29] gave a one-round ID-based CK-secure AKE protocol with perfect forward security. Pandit *et al.* [30] and Ni *et al.* [31] designed eCK-secure ID-based AKE protocols. Elashry *et al.* [32] presented a new CK security model for AKE protocol, where if a shared session key was compromised or leaked, parties could generate another new shared session key without running a new AKE session.

LR Model: Micali and Reyzin [2] first introduced a LR security model of public-key cryptosystems, where they supposed that only computations may leak information and treated side-channel attacks in an abstract way. They called it only computation leakage (OCL) model. In 2009, Akavia *et al.* [3] gave a general LR security model of public-key cryptosystems. They modeled side-channel attacks as following: Based on all his knowledge, the adversary $\boldsymbol{A}$ chooses an arbitrary leakage function f, which could be computed efficiently; $\boldsymbol{A}$ sends f to a leakage oracle $\boldsymbol{O}$; $\boldsymbol{O}$ computes $f(sk)$ and sends it to $\boldsymbol{A}$, where sk is the long-term secret key. This model is called RLM, where each leakage attack can reveal at most some small fraction of the secret key, and the total leakage length that $\boldsymbol{A}$ can get should not exceed a relatively percentage of the secret key size. Alwen [4] introduced BRM that is a generalization of RLM. In BRM, the leakage-parameter λ is independent of the security parameters and is only related to the secret key size s. Then, BRM could set λ and s according to practical considerations of the leakage tolerance, and λ could be arbitrary large. At the same time, computation time and communication bandwidth is independent of λ. Thus, BRM still remains efficient.

In 2010, Brakerski *et al.* [5] introduced a more practical LR security model, where leakages could be happened from the entire secret memory and are allowed continuously. They called this model the continuous leakage model (CLM). In CLM, the overall leakage amount could be arbitrarily large, but for each leakage it still needs the amount of leakage is bounded.

AF Leakage: Leakage that happens after the adversary gets the challenge is called AF leakage. In the AKE security models, the challenge to the adversary is to identity the real session key of the test session from a random key, AF leakage is the leakage which happens after the test session is established. In the above leakage models [3]–[5], the leakages are only allowed before the adversary chooses the test session and gets the challenge. Halevi *et al.* [6] proposed the chosen plaintext BAFLR security model of public-key cryptosystems. Dziembowski *et al.* [7] introduced an adaptively chosen ciphertext AFLR security model of public-key cryptosystems.

LR AKE: Moriyama and Okamoto [20] first presented the formal security notion of LR-AKE protocols in the eCK security model, named λ-LR eCK security model. They gave a PKI-based λ-LR eCK-secure AKE protocol based on hash proof system without random oracles (RO). There is one central limitation in the MO model, in which the adversary could only get the leakage information before he selected the test session. Alawatugoda *et al*. [21] first presented an AFLR security model, and constructed a CAFLR-AKE protocol using existing LR cryptographic primitives. Then, they gave a formal proof in the CAFLR-CK security model. Alawatugoda *et al*. [22] proposed a generic AFLR-eCK security model for AKE protocols, and gave a concrete construction for BAFLR eCK-secure AKE protocols. Alawatugoda *et al*. [23] proposed the first concrete construction of CAFLR eCK-secure AKE protocol based on continuously refreshing LR storage scheme, and formally proved it in the RO model. Chen *et al*. [24] introduced a strong AFLR eCK security model, which not only captured leakage attacks on long-term secret private key, but also considered leakage of ephemeral secret randomness. Based on smooth projective hash functions, they constructed a one-round AKE protocol in the CAFLR eCK security model. Recently, Toorani [33] showed that Alawatugoda *et al*.'s AKE protocol [21] was insecure by giving an ephemeral key compromise

impersonation(KCI) attack in the CK-secure model; Yang *et al*. [34] also presented a KCI attack against Alawatugoda *et al*.'s AKE protocol [22], and pointed out that their proofs of Case 2 (the adversary is active) were incorrectly reduced to decision Diffie–Hellman (DDH) assumption, and reproved them in the RO model under Gap Diffie–Hellman (GDH) assumption; Chakaraborty *et al*. [35] showed the proofs of Chen *et al*.'s AKE protocol [24] had the same problem as [22], and gave the correct proofs in the RO model under GDH assumption.

III. Preliminaries

This section addresses the used primitives, such as DDH assumption, IB-HPS, and KDF.

Notations: : Assume $\mathrm{s} \xleftarrow{\$} S$ represents that *s* is a random value, chosen uniformly from a finite set *S*, κ denote the system security parameter, and λ be the leakage parameter.

Definition 3.1: [Negligible function]

A function $\varepsilon(\kappa)$ is negligible, if for every positive integer $c \geq 0$, there exists an integer k_c, such that $\varepsilon(\kappa) < \kappa^{-c}$ for all $\kappa \geq k_c$.

Definition 3.2: [Statistical Indistinguishability]

Suppose *S* and *T* be two random variables over a finite set Ω, the statistical distance between *S* and *T* is defined as

$$\mathrm{SD}(S,T) = 1/2\Sigma_{\omega\in\Omega}|\Pr[S=\omega] - \Pr[T=\omega]|.$$

S and *T* are ε-statistically indistinguishable, if $\mathrm{SD}(S,T) \leq \varepsilon$, denoted as $S \overset{s}{\underset{\varepsilon}{\equiv}} T$ for simplicity; and *S* and *T* are perfectly indistinguishable, if $\varepsilon = 0$.

Definition 3.3: [Decision Diffie–Hellman Assumption]

Suppose G denote a cyclic multiplicative group with a large prime order *p,* and *g* be a random generator for G. The challenger ***C*** runs the following distinguishing game with an adversary ***A***:

1) ***A*** is provided with G and *g*.
2) ***C*** picks a random bit $b \xleftarrow{\$} (0,1)$. If $b = 0$, ***C*** sends (g^x, g^y, g^{xy}) to ***A***, else ***A*** is given (g^x, g^y, g^z), where $x, y, z \xleftarrow{\$} Z_p^*$.
3) ***A*** outputs a bit $b' \in (0,1)$. ***A*** wins, if $b' = b$.

Suppose $Adv_{DDH}(\mathrm{A})$ represent the advantage that ***A*** wins the above distinguishing game, and $\varepsilon(\kappa)$ be a negligible function, then the DDH assumption is that

$$Adv_{DDH}(A) = |\Pr[b'=b] - 1/2| = \varepsilon(\kappa).$$

Definition 3.4: [Identity-Based Hash Proof System]

An IB-HPS includes the following five probabilistic polynomial time (PPT) algorithms:

Setup: $(mpk, msk) \leftarrow Setup(1^\kappa)$, where κ is an input security parameter, *mpk* and *msk* denote the master public key and the master secret key, respectively. *mpk is* the public inputs for all other algorithms, and could be viewed as common reference string. At the same time, *mpk* identifies an identity set *ID* and an encapsulated-key set *K*.

KeyGen:$sk_{id} \leftarrow KeyGen(id, msk)$: ***KeyGen*** generates an ID secret private key sk_{id} for each identity $id \in ID$ by using the master secret key *msk*.

Encap: $(c,k) \leftarrow Encap(id)$: ***Encap*** is the valid encapsulation algorithm that produces a valid ciphertext c, and an encapsulated-key, $k \in K$.

Encap*:$c \leftarrow Encap^*(id)$: ***Encap**** is the alternative invalid encapsulation algorithm that creates an invalid ciphertext *c* for an identity *id*.

Decap:$k \leftarrow Decap(c, sk_{id})$: ***Decap*** is the deterministic decapsulation algorithm, which takes as inputs a ciphertext *c* and an identity secret private key sk_{id}, and outputs the encapsulated key *k*.

An IB-HPS should meet the following two properties:

1) Correctness of Decapsulation. For every *mpk*, *msk* generated by $Setup(1^\kappa)$and every $id \in ID$, there is

$$\Pr\left[k \neq k' \;\middle|\; \begin{array}{l} sk_{id} \leftarrow KeyGen(id, msk) \\ (c,k) \leftarrow Encap(id) \\ k' \leftarrow Decap(c, sk_{id}) \end{array}\right] \leq \varepsilon(\kappa)$$

where $\varepsilon(\kappa)$ is a negligible function.

2) Valid/Invalid Ciphertext Indistinguishability. The valid ciphertext produced by ***Encap*** and the invalid ciphertext created by ***Encap**** should be indistinguishable even given the ID secret private key. This property is captured by the following distinguishing game, which the challenger ***C*** runs with an adversary ***A***.
   a) ***C*** generates (mpk, msk) by running $Setup(1^\kappa)$, then sends *mpk* to ***A***.
   b) Test Stage 1: For any $id \in ID$ queried by ***A***, ***C*** replies with sk_{id}.
   c) ***A*** chooses a random challenge identity $id^* \in ID$, then ***C*** picks $b \xleftarrow{\$} (0,1)$, computes *c* and gives it to ***A***, where $(c,k) \leftarrow \mathrm{Encap}(id^*)$ if $b = 0$, and $c \leftarrow \mathrm{Encap}^*(id^*)$ if $b = 1$. In this stage, ***A*** could select any identity id^*, even one that he has queried in the Test Stage 1, and he could query this id^* in the Test Stage 2.
   d) Test Stage 2: For any $id \in ID$ queried by ***A***, ***C*** replies with sk_{id}.
   e) ***A*** outputs a bit $b' \in (0,1)$. ***A*** wins if $b' = b$.

Suppose $Adv^{\mathrm{V/I}-CI}_{\mathrm{IB}-HPS}(\mathrm{A})$ denote the advantage of ***A*** in distinguishing the above security game and $\varepsilon(\kappa)$ be a negligible function, then valid/invalid ciphertext indistinguishability means that

$$Adv^{\mathrm{V/I}-CI}_{\mathrm{IB}-HPS}(\mathrm{A}) = \varepsilon(\kappa).$$

Note: In both test stages, if *id* is the first query, ***C*** generates sk_{id} using $KeyGen(id, msk)$ and sends it to ***A***, otherwise ***C*** replies with the samesk_{id} for all future same *id* queries.

Definition 3.5: [Smooth IB-HPS]

An **IB-HPS** is smooth if, for every fixed values of (*mpk*,*msk*) generated by $\mathrm{Setup}(1^\kappa)$, every $id \in ID$, there is

$$Adv^{\mathrm{S}}_{\mathrm{IB}-HPS}(\mathrm{A}) = SD((c,k),(c,k')) \leq \varepsilon(\kappa)$$

where $c \leftarrow Encap^*(id)$, $k \leftarrow Decap(c, KeyGen(id, msk))$, $k' \xleftarrow{\$} K$, $\varepsilon(\kappa)$ denotes a negligible function and $Adv^{\mathrm{S}}_{\mathrm{IB}-HPS}(\mathrm{A})$ is the advantage of the adversary ***A*** in distinguishing the de-

capsulation of an invalid ciphertext and a random value picked from *K*.

Definition 3.6: [λ-Leakage-Smooth IB-HPS]

An **IB-HPS** is λ-leakage-smooth if, for every leakage function *f* with λ-bit output, there is

$$Adv^{\mathrm{L-S}}_{\mathrm{IB}-HPS}(\mathrm{A}) = SD((c, f(sk_{id}), k), (c, f(sk_{id}), k')) \leq \varepsilon(\kappa)$$

where (*mpk*, *msk*),sk_{id}, *c*, *k*, *k*′, $\varepsilon(\kappa)$ are same as above, and $Adv^{\mathrm{L-S}}_{\mathrm{IB}-HPS}(\mathrm{A})$ is the advantage of the adversary ***A*** with λ-bit leakage output in distinguishing the decapsulation of an invalid ciphertext and a random value picked from *K*.

Definition 3.7: [Source of Key Material]

A source of key material Σ is a two-valued (σ, ℓ) probability distribution generated by a PPT algorithm, where σ represents the source material of the secret keys and ℓ denotes some public knowledge about σ, such as its length.

Definition 3.8: [Key Derivation Function]

A KDF is an efficient algorithm that takes as inputs (σ, ℓ, *r*, *c*) and generates a cryptographically strong secret key, where (σ, ℓ) are selected from a source of keying material Σ, and (*r*, *c*) are two optional arguments that *r* is a salt value and *c* is a context variable.

Definition 3.9: [Security of KDF]

Security of KDF with a source of key material Σ is defined by the following distinguishing game, which the challenger ***C*** runs with an adversary ***A***.

1) ***C*** generates $(\sigma, \ell) \leftarrow \Sigma$ and a random salt value *r,* and sends (ℓ, r) to ***A***.
2) ***A*** selects an arbitrary value *c* and sends it to ***C***.
3) ***C*** picks $b \xleftarrow{\$} (0, 1)$ at random. If $b = 0$, ***C*** computes $k =$ KDF(σ, ℓ, *r*, *c*) and sends it to ***A***, else ***C*** chooses a random string *s* with the same length as *k* and gives it to ***A***.
4) ***A*** outputs a bit $b' \in (0, 1)$. ***A*** wins if $b' = b$.

Suppose $Adv_{KDF}(\mathrm{A})$denote the advantage of the adversary ***A*** in distinguishing the above security game and $\varepsilon(\kappa)$ be a negligible function, then the security of KDF means that

$$Adv_{KDF}(A) = \varepsilon(\kappa).$$

IV. Bounded After-the-Fact LR eCK Security Model

This section illustrates the BAFLR eCK security model that is the bounded instantiation of Alawatugoda *et al*.'s generic AFLR eCK security model [22] for AKE protocols. The BAFLR eCK model follows the OCL model, and assume that leakage occurs only in computations associated with the long-term secret key sk. The adversary can adaptively choose arbitrary PPT leakage functions $f = (f_1, \ldots, f_n)$to obtain leakage of the secret keys of the protocol principals. We require that the total leakage size is bounded, i.e., $\sum |f_i(sk)| \leq \lambda$. After issuing a ***Send*** query with f_i, the adversary will be given a normal protocol message and the leakage $f_i(sk)$.

A. Adversarial Powers

Let *U*, *V* identify two parties, the term "principal" represent a party involved in a protocol instance, and the term "session" denote a protocol instance with principals. Each principal may have multiple sessions that maybe run concurrently. We denote the *s*th session at the owner principal *U,* interacting with the intended partner principal *V* as the oracle $\Pi^s_{U,V}$, and denote the principal, who activates a session as the initiator of the session, and the principal who responds to the initiator as the responder.

The adversary ***A*** is a PPT algorithm that controls all communications over the whole network and interacts with a set of oracles. In fact, ***A*** can do anything as he wants. The following queries model the capabilities of the adversary ***A***.

Send(U, V, *s*, *m*, *f*) query: After issuing this ***Send*** query in the *s*th session with a protocol message *m* and a leakage function *f*, ***A*** will be given a normal next protocol message and the leakage $f(sk_U)$ produced by the oracle $\Pi^s_{U,V}$. ***A*** can run the protocol by this query, and can also start a new protocol instance as an initiator by using this query with blank *m* and *f*.

RevealSessionKey (U, V, *s*) query: $\Pi^s_{U,V}$ sends the *s*th session key to ***A***. This query models ***A***'s ability to compromise the certain session key.

RevealEphemeralKey (U, V, *s*) query: $\Pi^s_{U,V}$ sends the *s*th session ephemeral keys to ***A***. This query models ***A***'s ability to compromise the certain ephemeral keys.

Corrupt (*U*) query: The principal *U* sends his long-term secret key to ***A***. This query models ***A***'s ability to get the certain principal's secret key.

Test (*U*, *s*) query: After receiving a ***Test*** query, the challenger ***C*** picks a bit $b \xleftarrow{\$} (0, 1)$ at random; if $b = 1$, then ***C*** sends the actual key of the *s*th session to ***A***, while ***C*** chooses a random key and sends it to ***A***. This query is used to formalize the security notion of a BAFLR-AKE protocol, and could be activated only once across all sessions.

B. λ-BAFLR eCK Security Model

In the λ-BAFLR eCK security model, the overall leakage size of long-term secret keys are bounded with the leakage parameter λ, *i.e.*, $\sum |f_i(sk)| \leq \lambda$.

Definition 4.1: [Partner sessions in BAFLR eCK security model]

Two oracles $\Pi^s_{U,V}$ and $\Pi^{s'}_{U',V'}$ are called partners, if the following hold.

1) Both $\Pi^s_{U,V}$ and $\Pi^{s'}_{U',V'}$ have generated session keys.
2) Messages sent from $\Pi^s_{U,V}$ are same as those received by $\Pi^{s'}_{U',V'}$.
3) Messages sent from $\Pi^{s'}_{U',V'}$ are same as those received by $\Pi^s_{U,V}$.
4) $U = V'$ and $V = U'$.
5) Exactly one of *U* and *V* is the initiator, and the other is the responder.
6) Correctness of an AKE protocol means that two partners generate same session keys.

Definition 4.2: [λ-BAFLR-eCK-freshness]

Let $f = (f_1, \ldots, f_n)$ be *n* arbitrary PPT leakage functions chosen by the adversary. An oracle $\Pi^s_{U,V}$ is λ-BAFLR-eCK-fresh, if the following hold.

1) RevealSessionKey(U,V,s) or RevealSessionKey(V,U, s') (if $\Pi^s_{U,V}$'s partner, $\Pi^{s'}_{V,U}$, exists) has not been asked.

User U ($\mathrm{ID_U}$, Initiator)		User V ($\mathrm{ID_V}$, Responder)
Initial Setup		
$(mpk, msk) \leftarrow Setup(1^{\kappa})$		
$sk_U \leftarrow KeyGen(\mathrm{ID}_U, msk)$		$sk_V \leftarrow KeyGen(\mathrm{ID}_V, msk)$
Protocol Execution		
$(c_U, k_U) = Encap(ID_V)$		
$x_U \xleftarrow{\$} Z_p^*, Y_U = g^{x_U}$	$(\mathrm{ID}_U, c_U, Y_U)$ →	$(c_V, k_V) = Encap(ID_U)$
	← $(\mathrm{ID}_V, c_V, Y_V)$	$x_V \xleftarrow{\$} Z_p^*, Y_V = g^{x_V}$
$k_V = \underline{Decap}(c_V, sk_U)$		$k_U = \underline{Decap}(c_U, sk_V)$
$k_{UV} = KDF(ID_U, ID_V, Y_V^{x_U}, k_U \,\|\| k_V)$		$k_{UV} = KDF(ID_U, ID_V, Y_U^{x_V}, k_U \| k_V)$
k_{UV} is the session key		

Fig. 1. General framework of BAFLR eCK-secure ID-based AKE protocol.

2) If the partner $\Pi_{V,U}^{s'}$ exists, none of the following combinations has been queried.
   a) Corrupt(U) and RevealEphemeralKey(U, V, s).
   b) Corrupt(V) and RevealEphemeralKey(V,U, s').
3) If the partner $\Pi_{V,U}^{s'}$ does not exist, none of the following combinations has been queried.
   a) Corrupt(V).
   b) Corrupt(U) and RevealEphemeralKey(U, V, s).
4) For all Send(.,U,.,.,f_i) queries, $\sum |f_i(sk_U)| \leq \lambda$.
5) For all Send(.,V,.,.,f_i) queries,$\sum |f_i(sk_V)| \leq \lambda$.

C. Security Definition

This section formalizes the distinguishing game and the security definition of the ..-BAFLR eCK model.

Definition 4.3: [λ-BAFLR eCK distinguishing game]

In the model, the protocol challenger ***C*** will run the following distinguishing game with a PPT adversary ***A***.

1) ***A*** queries any of ***Send***, ***RevealSessionKey***, ***RevealEphemeralKey,*** and ***Corrupt*** to any oracle as he wants.
2) ***A*** selects a λ-BAFLR-eCK-fresh oracle and issues a ***Test*** query. After receiving a ***Test*** query, ***C*** picks a random bit $b \xleftarrow{\$} (0,1)$ if $b = 1$, then sends the actual session key to ***A***, while a random session key is sent to ***A***.
3) ***A*** continues querying ***Send***, ***RevealSessionKey***, ***RevealEphemeralKey,*** and ***Corrupt***. All these queries should not violate the λ-BAFLR-eCK-freshness of the test session.
4) At last ***A*** outputs a bit $b' \in (0,1)$. ***A*** wins if $b' = b$.

Definition 4.4: [λ-BAFLR eCK security]

λ-BAFLR eCK security means that

$$Adv_{\mathrm{AKE}}^{\lambda-BAFLReCK} = |\Pr[b' = b] - 1/2| = \varepsilon(\kappa)$$

where $Adv_{\mathrm{AKE}}^{\lambda-BAFLReCK}$ is the advantage of ***A*** winning the λ-BAFLR eCK distinguishing game in Definition 4.3, and $\varepsilon(\kappa)$ is a negligible function. In other words, an AKE protocol is λ-BAFLR eCK-secure, if there doesn't exist any PPT adversary ***A*** that can win the above distinguishing game with non-negligible advantage.

V. General Framework of λ-BAFLR eCK-Secure ID-Based AKE Protocol

This section gives our proposed general framework for the λ-BAFLR AKE protocol and a formal security proof in the eCK security model.

A. General Framework

Fig. 1 shows a general framework for the λ-BAFLR eCK-secure AKE protocol. In the proposed scheme, parties U and V can establish a strong secure session key *sk* over a public unreliable network. Following the OCL model, we assume that leakage occurs only in computations associated with a long-term secret key sk. Therefore, the computations using these long-term secret keys may leak some partial secret information of them, and we require that the total size of all these leakage should be bounded to λ. To overcome this challenge, we introduce the λ-leakage-smooth IB-HPS. We underline the computations that may leak information about long-term secret keys. Let G denote a cyclic multiplicative group with a large prime order p and g represent a random generator for G.

In the initial setup stage, the dealer generates each user's ID secret private key using the ***Setup*** and ***KeyGen*** algorithm of IB-HPS and sends them to him/her secretly. In the protocol execution stage, each of the principals computes a valid ciphertext and an encapsulated-key using the valid encapsulation ***Encap*** algorithm of IB-HPS, and picks his/her ephemeral secret key at random, encrypts it by computing its exponentiation, then sends all these messages to the intended partner principal. Then, both principals obtain the encapsulated-key by computing the decapsulation ***Decap*** algorithm of IB-HPS, and use KDF with the two identities, the exchanging ephemeral secrets and the encapsulated-keys to generate the session key. In the construction of the framework, the invalid encapsulation algorithm **Encap*** will not be used, and it will be used for proving the security of the proposed framework.

B. Security Proof

This section formally proves the security of the proposed protocol in the standard model.

Theorem 5.1: The proposed protocol is λ-BAFLR eCK-secure, if the DDH assumption holds, the IB-HPS is λ-leakage-smooth and the KDF with a source of uniformly random key material Σ is secure. Let $Adv_{\mathrm{AKE}}^{\lambda-BAFLReCK}$ denote the advantage of a PPT adversary $\boldsymbol{A}$ against λ-BAFLR eCK-security of the AKE protocol, there is

$$Adv_{\mathrm{AKE}}^{\lambda-BAFLReCK} \leq N_P^2 N_S^2 (Adv_{DDH} + Adv_{IB-HPS}^{V/I-CI} + Adv_{\mathrm{IB}-HPS}^{\mathrm{L-S}} + Adv_{KDF})$$

where Adv_{DDH}, Adv_{KDF}, $Adv_{IB-HPS}^{V/I-CI}$, $Adv_{\mathrm{IB}-HPS}^{\mathrm{L-S}}$ are advantages of $\boldsymbol{A}$ against the security of DDH problem, KDF and λ-leakage-smooth IB-HPS, respectively, and N_P denotes the number of protocol principals, N_S represents the number of sessions on a principal.

We use the game hopping technique to formally prove the BAFLR eCK security of the proposed AKE protocol in the standard model. The proof structure is first defining a sequence of games, then proving indistinguishability of each game and its previous game, which is similar to Alawatugoda *et al.* [22].

Proof: Our proof can be split into two main cases: When the partner to the test session exists, and when it does not.

Case 1: A partner session to the test session exists:

In this case, $\boldsymbol{A}$ is a static adversary, who may get the long-term secret private keys of principals by the ***Corrupt*** query or ephemeral keys by the ***RevealEphemeralKey*** query. However, these queries should not violate the λ-BAFLR-eCK-freshness of the test session. Let $Adv_{\mathrm{AKE}}^{\lambda-BAFLReCK}$ denote the advantage that $\boldsymbol{A}$ wins the BAFLR eCK challenge against λ-BAFLR eCK distinguishing game. We divide this case into the following four sub cases.

1) A corrupts both the owner and partner principals to the test session. In this case, A could get both principals' secret private keys using Corrupt queries.
2) A corrupts the owner to the test session, but does not corrupt the partner. In this case, A could get the owner principal's secret private key.
3) A corrupts the partner to the test session, but does not corrupt the owner. In this case, A could get the partner principal's secret private key.
4) A corrupts neither owner nor partner principal to the test session. In this case, A could not get both principals' secret private keys.

Case 1.1 A: corrupts both the owner and partner principals to the test session:

In this case, $\boldsymbol{A}$ could learn k_{U^*}, k_{V^*}, because he can get both principals' secret private keys using ***Corrupt*** queries.

Game 1: This game is the original λ-BAFLR eCK security game. After receiving a ***Test*** query, Game 1 challenger picks a bit $b \overset{\$}{\leftarrow} (0,1)$ at random, if $b = 1$, then sends the actual session key to $\boldsymbol{A}$, while a random session key is sent to $\boldsymbol{A}$.

Game 2: Game 2 and Game 1 are same, except for the following: $\boldsymbol{A}$ first picks two distinct principals $U^*, V^* \overset{\$}{\leftarrow}$, $\{U_1, \ldots, U_{N_p}\}$and two numbers $s^*, t^* \overset{\$}{\leftarrow} \{1, \ldots, N_s\}$ at random, where N_P is the number of protocol principals and N_S is the number of sessions on a principal. Then, $\boldsymbol{A}$ begins to run the game and selects the oracle $\Pi_{U^*,V^*}^{s^*}$ as the target session and $\Pi_{V^*,U^*}^{t^*}$ as its partner session. If the test session is not $\Pi_{U^*,V^*}^{s^*}$ or its partner oracle is not $\Pi_{V^*,U^*}^{t^*}$, Game 2 challenger stops and terminates the game.

Game 3: Game 3 and Game 2 are same, except for the following: Game 3 challenger $\boldsymbol{C}$ picks $z \overset{\$}{\leftarrow} Z_p^*$ at random and computes $k_{U^*V^*} = KDF(ID_{U^*}, ID_{V^*}, g^z, k_{U^*}||k_{V^*})$. Upon receiving $Test(U^*, V^*, s^*)$ query from $\boldsymbol{A}$, $\boldsymbol{C}$ sends $k_{U^*V^*}$ to $\boldsymbol{A}$. Further, upon receiving $Test(V^*, U^*, t^*)$ query from $\boldsymbol{A}$, $\boldsymbol{C}$ also sends the same $k_{U^*V^*}$ to $\boldsymbol{A}$, since there is a partner session $\Pi_{V^*,U^*}^{t^*}$ in Game 3.

Game 4: Game 4 and Game 3 are same, except for the following: Game 4 challenger $\boldsymbol{C}$ picks $k_{U^*V^*} \overset{\$}{\leftarrow} \{0,1\}^k$ at random. Then, upon receiving $Test(U^*, V^*, s^*)$ query or $Test(V^*, U^*, t^*)$ query from $\boldsymbol{A}$, $\boldsymbol{C}$ sends $k_{U^*V^*}$ to $\boldsymbol{A}$.

Differences between games: This section investigates indistinguishability of each game *t* and its previous game *t*−1. Suppose $\mathrm{A}dv_{Game t}(A)$ represent the advantage of $\boldsymbol{A}$ in winning Game *t*.

Game 1: It is the original game. Therefore,

$$\mathrm{A}dv_{\mathrm{Game1}}(A) = Adv_{\mathrm{AKE}}^{\lambda-BAFLReCK}. \quad (1)$$

Game 1 and Game 2: Unless $\boldsymbol{A}$ chooses an incorrect test session or an incorrect partner to the test session, Game 2 and Game 1 are same. The probability that $\boldsymbol{A}$ chooses a correct test session and a correct partner to the test session is $1/N_P^2 N_S^2$. Therefore,

$$\mathrm{A}dv_{\mathrm{Game2}}(A) = 1/N_P^2 N_S^2 \mathrm{A}dv_{\mathrm{Game1}}(A). \quad (2)$$

Game 2 and Game 3:

In Game 2, $k_{U^*V^*} = KDF(ID_{U^*}, ID_{V^*}, g^{x_{U^*}x_{V^*}}, k_{U^*}||k_{V^*})$, while $k_{U^*V^*} = KDF(ID_{U^*}, ID_{V^*}, g^z, k_{U^*}||k_{V^*})$ in Game 3. $g^{x_{U^*}x_{V^*}}$ and g^z are indistinguishable from DDH assumption, thus $\boldsymbol{A}$ could not distinguish between Game 2 and Game 3. Therefore,

$$|\mathrm{A}dv_{\mathrm{Game2}}(A) - \mathrm{A}dv_{\mathrm{Game3}}(A)| \leq Adv_{DDH}. \quad (3)$$

Game 3 and Game 4:

In Game 3, $k_{U^*V^*} = KDF(ID_{U^*}, ID_{V^*}, g^z, k_{U^*}||k_{V^*})$, while in Game 4 $k_{U^*V^*} \overset{\$}{\leftarrow} \{0,1\}^k$. From the security of KDF, $\boldsymbol{A}$ could not distinguish between Game 3 and Game 4. Therefore,

$$|\mathrm{A}dv_{\mathrm{Game3}}(A) - \mathrm{A}dv_{\mathrm{Game4}}(A)| \leq Adv_{KDF}. \quad (4)$$

Game 4: $\boldsymbol{A}$ has not any advantage in winning Game 4 because the session key $k_{U^*V^*}$ of $\Pi_{U^*,V^*}^{s^*}$ is picked at random and does not depend on any other values. Therefore,

$$\mathrm{A}dv_{\mathrm{Game4}}(A) = 0. \quad (5)$$

Using (1)–(5) we get

$$Adv_{\mathrm{AKE}}^{\lambda-BAFLReCK} \leq N_P^2 N_S^2 (Adv_{DDH} + Adv_{KDF}).$$

Case 1.2 A: corrupts the owner to the test session, but does not corrupt the partner:

For simplify, suppose the test session be on the initiator.

In this case, $\boldsymbol{A}$ could learn $k_{\mathrm{V}*} = \mathrm{Decap}(c_{V*}, sk_{U*})$using ***Corrupt***(U∗) query, and get x_{V^*} using ***RevealEphemeralKey*** (V∗, U∗, s∗) query.

Game 1: Same as Game 1 in Case 1.1.

Game 2: Same as Game 2 in Case 1.1.

Game 3: Game 3 and Game 2 are same, except for the following: Game 3 challenger $\boldsymbol{C}$ computes

$$c_{V^*} = \text{Encap} * (ID_{U^*}),\ k_{U^*} = \text{Decap}(c_{V^*}, ID_{U^*})$$
$$k_{U^*V^*} = KDF(ID_{U^*}, ID_{V^*}, {Y_{U^*}}^{x_{V^*}}, k_{U^*}||k_{V^*}).$$

Upon receiving $Test(U^*, V^*, s^*)$ query from the adversary $\boldsymbol{A}$, $\boldsymbol{C}$ sends $k_{U^*V^*}$ to $\boldsymbol{A}$. Further, upon receiving $Test(V^*, U^*, t^*)$ query from $\boldsymbol{A}$, $\boldsymbol{C}$ also sends the same $k_{U^*V^*}$ to $\boldsymbol{A}$, since there is a partner session $\Pi^{t^*}_{V^*,U^*}$ in Game 3.

Game 4: Game 4 and Game 3 are same, except for the following: Game 4 challenger $\boldsymbol{C}$ computes

$$c_{V^*} = \text{Encap} * (ID_{U^*}), k_{U^*} \overset{\$}{\leftarrow} U_K$$
$$k_{U^*V^*} = KDF(ID_{U^*}, ID_{V^*}, {Y_{U^*}}^{x_{V^*}}, k_{U^*}||k_{V^*}).$$

Upon receiving $Test(U^*, V^*, s^*)$ query from the adversary $\boldsymbol{A}$, $\boldsymbol{C}$ sends $k_{U^*V^*}$ to $\boldsymbol{A}$. Further, upon receiving $Test(V^*, U^*, t^*)$ query from $\boldsymbol{A}$, $\boldsymbol{C}$ also sends the same $k_{U^*V^*}$ to $\boldsymbol{A}$, since there is a partner session $\Pi^{t^*}_{V^*,U^*}$ in Game 4.

Game 5: Same as Game 4 in Case 1.1.

Differences between games:

Game 1: It is the original game. Therefore,

$$\text{A}dv_{Game1}(A) = Adv^{\lambda-BAFLReCK}_{\text{AKE}} \tag{6}$$

Game 1 and Game 2: Same as Game 1 and Game 2 in Case 1.1.,

$$\text{A}dv_{Game2}(A) = 1/N_P^2 N_S^2 \text{A}dv_{Game1}(A). \tag{7}$$

Game 2 and Game 3: In Game 3, the challenger $\boldsymbol{C}$ computes $c_{V^*} = Encap * (ID_{U^*})$ using the invalid encapsulation algorithm ***Encap****. $\boldsymbol{A}$ could not distinguish between Game 2 and Game 3, because the valid ciphertext and invalid ciphertext of IB-HPS are indistinguishable. Notice that, leakage queries are not allowed in the valid/invalid ciphertext security game, however, $\boldsymbol{A}$ could get the entire secret key sk_{U*} of the challenge identity $U*$. Thus, Game 2 and Game 3 are indistinguishable for $\boldsymbol{A}$, who could get some bounded leakage $f(sk_{U*})$of the ID secret-key sk_{U*}. Therefore,

$$|\text{A}dv_{Game2}(A) - \text{A}dv_{Game3}(A)| \leq Adv^{\text{V/I-CI}}_{IB-HPS}(\text{A}). \tag{8}$$

Game 3 and Game 4: In Game 3 $k_{U^*V^*} = KDF(ID_{U^*}, ID_{V^*}, {Y_{U^*}}^{x_{V^*}}, k_{U^*}||k_{V^*})$, while in Game 4, the challenger $\boldsymbol{C}$ picks a random key, $k_{U^*} \overset{\$}{\leftarrow} U_K$. $\boldsymbol{A}$ could not distinguish between Game 3 and Game 4 by the λ-leakage-smoothness of IB-HPS. Therefore

$$|\text{A}dv_{Game3}(A) - \text{A}dv_{Game4}(A)| \leq Adv^{\text{L-S}}_{IB-HPS}(\text{A}). \tag{9}$$

Game 4 and Game 5: Same as Game 3 and Game 4 in Case 1.1.,

$$|\text{A}dv_{Game4}(A) - \text{A}dv_{Game5}(A)| \leq Adv_{KDF}. \tag{10}$$

Game 5:

$$\text{A}dv_{Game5}(A) = 0. \tag{11}$$

Using (6)–(11) we get

$$Adv^{\lambda-BAFLReCK}_{\text{AKE}} \leq N_P^2 N_S^2 (Adv^{V/I-CI}_{IB-HPS} + Adv^{\text{L-S}}_{\text{IB}-HPS} + Adv_{KDF}).$$

Case 1.3 A: corrupts the partner to the test session, but does not corrupt the owner:

For simplify, suppose the test session be on the initiator.

In this case, $\boldsymbol{A}$ could learn $k_{U*} = Decap(c_{U*}, sk_{V*})$using ***Corrupt***(V∗) query and get x_{U^*} using ***RevealEphemeral Key***(U∗,V∗, s∗) query.

Game 1: Same as Game 1 in Case 1.1.

Game 2: Same as Game 2 in Case 1.1.

Game 3: Game 3 and Game 2 are same, except for the following: Game 3 challenger $\boldsymbol{C}$ computes

$$c_{\text{U}^*} = Encap * (ID_{\text{V}^*}), k_{\text{V}^*} = Decap(c_{\text{U}^*}, ID_{V^*})$$
$$k_{U^*V^*} = KDF(ID_{U^*}, ID_{V^*}, {Y_{\text{V}^*}}^{x_{U^*}}, k_{U^*}||k_{V^*}).$$

Upon receiving $Test(U^*, V^*, s^*)$ query from the adversary $\boldsymbol{A}$, $\boldsymbol{C}$ sends $k_{U^*V^*}$ to $\boldsymbol{A}$. Further, upon receiving $Test(V^*, U^*, t^*)$ query from $\boldsymbol{A}$, $\boldsymbol{C}$ also sends the same $k_{U^*V^*}$ to $\boldsymbol{A}$, since there is a partner session $\Pi^{t^*}_{V^*,U^*}$ in Game 3.

Game 4: Game 4 and Game 3 are same, except for the following: Game 4 challenger $\boldsymbol{C}$ computes

$$c_{\text{U}^*} = Encap * (ID_{\text{V}^*}), k_{\text{V}^*} \overset{\$}{\leftarrow} U_K$$
$$k_{U^*V^*} = KDF(ID_{U^*}, ID_{V^*}, {Y_{\text{V}^*}}^{x_{\text{U}^*}}, k_{U^*}||k_{V^*}).$$

Upon receiving $Test(U^*, V^*, s^*)$ query from the adversary $\boldsymbol{A}$, $\boldsymbol{C}$ sends $k_{U^*V^*}$ to $\boldsymbol{A}$. Further, upon receiving $Test(V^*, U^*, t^*)$ query from $\boldsymbol{A}$, $\boldsymbol{C}$ also sends the same $k_{U^*V^*}$ to $\boldsymbol{A}$, since there is a partner session $\Pi^{t^*}_{V^*,U^*}$ in Game 4.

Game 5: Same as Game 4 in Case 1.1.

Differences between games: The analysis is same as Case 1.2.

Case 1.4 A corrupts neither owner nor partner principal to the test session:

In this case, $\boldsymbol{A}$ could get x_{U^*} and x_{V^*} using ***RevealEphemeralKey*** query.

Game 1: Same as Game 1 in Case 1.1.

Game 2: Same as Game 2 in Case 1.1.

Game 3: Game 3 and Game 2 are same, except for the following: Game 3 challenger $\boldsymbol{C}$ computes

$$c_{V^*} = Encap * (ID_{U^*}), k_{U^*} = Decap(c_{V^*}, ID_{U^*})$$
$$c_{U^*} = Encap * (ID_{V^*}), k_{V^*} = Decap(c_{U^*}, ID_{V^*})$$
$$k_{U^*V^*} = KDF(ID_{U^*}, ID_{V^*}, g^{x_{U^*}x_{V^*}},\ k_{U^*}||k_{V^*}).$$

Upon receiving the $Test(U^*, V^*, s^*)$ query from the adversary $\boldsymbol{A}$, $\boldsymbol{C}$ sends $k_{U^*V^*}$ to $\boldsymbol{A}$. Further, upon receiving $Test(V^*, U^*, t^*)$ query from $\boldsymbol{A}$, $\boldsymbol{C}$ also sends the same $k_{U^*V^*}$ to $\boldsymbol{A}$, since there is a partner session $\Pi^{t^*}_{V^*,U^*}$ in Game 3.

TABLE I
SECURITY AND EFFICIENCY COMPARISON OF AKE PROTOCOLS

Scheme	[18]	[20]	[21]	[22]	[23]	[24]	Ours
Rounds	2	2	1	2	1	1	**1**
Security model	eCK	eCK	CK	eCK	eCK	eCK	**eCK**
Leakage Feature	None	RLM	CLM	RLM	CLM	RLM	**BRM**
After-the-fact	Yes	No	Yes	Yes	Yes	Yes	**Yes**
Proof model	RO	Standard	Standard	RO	RO	RO	**Standard**
Key Infrastructure	PKI	PKI	PKI	PKI	PKI	PKI	**IB**

Game 4: Game 4 and Game 3 are same, except for the following: Game 4 challenger $\boldsymbol{C}$ computes

$$c_{V^*} = Encap*(ID_{U^*}), k_{U^*} \overset{\$}{\leftarrow} U_K,$$
$$c_{U^*} = Encap*(ID_{V^*}), k_{V^*} \overset{\$}{\leftarrow} U_K,$$
$$k_{U^*V^*} = KDF(ID_{U^*}, ID_{V^*}, g^{x_{U^*}x_{V^*}}, k_{U^*}||k_{V^*}).$$

Upon receiving the $Test(U^*, V^*, s^*)$ query from the adversary $\boldsymbol{A}$, $\boldsymbol{C}$ sends $k_{U^*V^*}$ to $\boldsymbol{A}$. Further, upon receiving $Test(V^*, U^*, t^*)$ query from $\boldsymbol{A}$, $\boldsymbol{C}$ also sends the same $k_{U^*V^*}$ to $\boldsymbol{A}$, since there is a partner session $\Pi^{t^*}_{V^*,U^*}$ in Game 4.

Game 5: Same as Game 4 in Case 1.1.

Differences between games: The analysis is same as Case 1.2.

Case 2 A partner session to the test session does not exist:

In this case, $\boldsymbol{A}$ is an active adversary who could run the protocol with the owner of the test session by masquerading as the intended partner principal. Therefore, $\boldsymbol{A}$ is not allowed to get the long-term secret key of the intended partner principal by asking a ***Corrupt*** query. We divide this case into the following two sub-cases.

Case 2.1 A corrupts the owner to the test session:

The proof is same as Case 1.2, except for Game 2 that is shown as following:

Game 2: It is same as the challenging game, except for the following: A first picks two distinct principals $U^*, V^* \overset{\$}{\leftarrow} \{U_1, \ldots, U_{N_p}\}$ and a number $s^* \overset{\$}{\leftarrow} \{1, \ldots, N_s\}$ at random, where N_P is the number of protocol principals and N_S is the number of sessions on a principal. Then, $\boldsymbol{A}$ begins to run the game and selects the oracle $\Pi^{s^*}_{U^*,V^*}$ as the target session. If the test session is not $\Pi^{s^*}_{U^*,V^*}$, Game 2 challenger stops and terminates the game.

Case 2.2 A does not corrupt the owner to the test session:

The proof is same as Case 1.4, except for Game 2 is same as Game 2 of Case 2.1.

From **Case 1** and **Case 2**, we get

$$Adv^{\lambda-BAFLReCK}_{\mathrm{AKE}} \leq N_P^2 N_S^2 (Adv_{DDH} + Adv^{V/I-CI}_{IB-HPS} + Adv^{\mathrm{L-S}}_{\mathrm{IB}-HPS} + Adv_{KDF}).$$

B. Performance and Security Comparison

We analyze the performance and security of our protocol by comparing with other representative AKE protocols. The comparison between our protocol and others is shown in Table I. From Table I, we should note the following.

1) Our new protocol is the first protocol for ID-based LR AKE.
2) Our protocol could be extended to the BRM, yielding the first LR-AKE protocol in BRM.
3) We consider a more strong security model, AFLR-eCK security model, while LaMacchia *et al.* [18] did not allowed leakage attacks, Moriyama and Okamoto [20] only addressed the leakage that happens before the test session is selected by the adversary, and Alawatugoda *et al.* [21] just gave the proof in the CK security model.
4) Our formal proof is given in the standard model, while the security of [18], [22]–[24] only could be proved in the RO model.
5) Our protocol only needs one-round of communication, while [18], [20], and [22] all required two-rounds of communication.

VI. INSTANTIATION OF THE FRAMEWORK

The primitives used in the framework are IB-HPS and KDF. For KDF, Krawczyk [26] proposed the secure and efficient KDFs based on HMAC. For IB-HPS, Chow *et al.* [36] and Alwen *et al.* [25] showed several efficient constructions of smooth IB-HPS based on different primitives, such as (1) bilinear groups, (2) lattices, and (3) quadratic residuosity.

A. Instantiation for Smooth-IB-HPS

Suppose G_1 represent an additive cyclic group with a large prime order p, G_2denote a cyclic multiplicative group with the same order p, and $\hat{e} : G_1 \times G_1 \rightarrow G_2$ be a bilinear map.

$Setup(1^\kappa)$: Compute $mpk = (p, G_1, G_2, \hat{e}, g, u, h, \hat{e}(g,g)^\alpha, \hat{e}(g,g)^\beta)$and $msk = (g^\alpha, g^\beta)$, where $u, h \overset{\$}{\leftarrow} G_1$ and $\alpha, \beta \overset{\$}{\leftarrow} \mathbb{Z}_p$.

$KeyGen(id, msk)$: $sk_{id} = (s_1, s_2, s_3) = (g^\alpha g^{-\beta t}(u^{id}h)^r, g^{-r}, t)$ for $id \in ID$, where $t, r \overset{\$}{\leftarrow} \mathbb{Z}_p$.

$Encap(id)$: Output $c = (c_1, c_2, c_3) = (g^z, (u^{id}h)^z, \hat{e}(g, g)^{\beta z})$ and $k = \hat{e}(g,g)^{\alpha z}$ where $z \overset{\$}{\leftarrow} \mathbb{Z}_p$.

$\mathrm{Encap}^*(id)$: Output $c = (c_1, c_2, c_3) = (g^z, (u^{id}h)^z, \hat{e}(g, g)^{\beta z'})$, where $z, z' \overset{\$}{\leftarrow} \mathbb{Z}_p$that $z \neq z'$.

$\mathrm{Decap}(c, sk_{id})$: Output $k = \hat{e}(c_1, s_1)\hat{e}(c_2, s_2)c_3^{s_3}$.

Theorem 6.1: (see [36]) If the Decisional Bilinear Diffie-Hellman assumption holds, the above protocol is a smooth IB-HPS.

B. Instantiation for λ-Leakage-Smooth IB-HPS

If there is a smooth IB-HPS (Setup, KeyGen, Encap, Encap*, Decap) and an extractor *Ext*: $k \rightarrow \{0,1\}^v$, a transformation is defined as follows.

$\mathrm{Encap}_2(id)$: Compute $(c, k) = Encap(id)$, $k' = Ext(k; r)$, output $((c, r), k')$.

$\mathrm{Encap}^*_2(id)$: Pick a seed r at random and compute $c = \mathrm{Encap}^*(id)$, output (c, r).

$\mathrm{Decap}_2(c, sk_{id})$: Compute $k = \mathrm{Decap}(c, sk_{id})$, $k' = Ext(k; r)$, output k'.

Theorem 6.2: (see [25]) Suppose that the IB-HPS is smooth, $|k| = 2^m$ and *Ext*: $k \to \{0,1\}^v$ is an $(m - \lambda, \varepsilon)$-extractor for some $\varepsilon = \varepsilon(\kappa)$, the above transformation generates a λ-leakage-smooth IB-HPS.

C. Extend to the BRM

Given an λ-leakage-smooth IB-HPS protocol, Alwen *et al.* [25] showed a transformation that could extend it to the BRM. First, in the transformed scheme, the leakage-parameter λ is an independent parameter of the system and is only related to the secret private key $\mathrm{sk}_{\mathrm{ID}}$, which has *n* components $(\mathrm{sk}_{\mathrm{ID}[1]}, \cdots, \mathrm{sk}_{\mathrm{ID}[\mathrm{n}]})$. Thus, λ could be set according to practical considerations of the leakage tolerance and may be arbitrary large. Second, the transformed scheme still satisfies the efficiency requirements. The encapsulation procedure ***Encap*** and decapsulation ***Decap*** use only a small subset of *t*-out-of-*n* of the identities, where *t* is independent of *n* and could be much smaller. Therefore, computation-time and communication-bandwidth still remain small, which are both independent of λ and *n*.

D. Leakage Tolerance of the Instantiation

The leakage tolerance of the instantiation of the proposed framework is same as the leakage tolerance of the IB-HPS, where leakage parameter, $\lambda = (1-\varepsilon)\mathrm{nm} - \nu - \kappa$, where *m* is the private key entropy, *n* is a key-size parameter, κ is the security parameter, *v* is the encapsulated-key size, and ε is a negligible constant. From the leakage formulas, we can get that the leakage tolerance of the proposed protocol could be arbitrarily large by just setting the key-size parameter *n* be a large positive integer.

VII. Conclusion

In this paper, we first propose a general framework for constructing one-round BAFLR eCK-secure ID-based AKE protocol based on IB-HPS and KDF, and give a formal security proof in the standard model. Moreover, our result could be extended to BRM, yielding the first LR-AKE protocol in BRM. Our future works include: 1) considering a strong security model that not only captures leakage attacks on long-term secret private key but also considers leakage of ephemeral secret randomness; 2) developing an efficient CAFLR eCK-secure ID-based AKE protocol; and 3) extending our result to the group setting.

References

[1] C. S. Chen, T. Wang, and J. Tian, "Improving timing attack on RSA-CRT via error detection and correction strategy," *Inform. Sci.*, vol. 232, pp. 464–474, 2013.

[2] S. Micali and L. Reyzin, "Physically observable cryptography," in *Proc. Theory Cryptography Conf.*, vol. 2951, Cambridge, MA, USA, 2004, pp. 278–296.

[3] A. Akavia, S. Goldwasser, and V. Vaikuntanathan, "Simultaneous hardcore bits and cryptography against memory attacks," in *Proc. Theory Cryptography Conf.*, vol. 5444, San Francisco, CA, USA, 2009, pp. 474–495.

[4] J. Alwen, Y. Dodis, and D. Wichs, "Leakage-resilient public-key cryptography in the bounded-retrieval model," in *Proc. Annu. Int. Cryptol. Conf.*, vol. 5677, Santa Barbara, CA, USA, 2009, pp. 36–54.

[5] Z. Brakerski, Y. T. Kalai, J. Katz, and V. Vaikuntanathan, "Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage," in *Proc. IEEE 51st Annu. Symp. Found. Comput. Sci.*, Las Vegas, NV, USA, 2010, pp. 501–510.

[6] S. Halevi and H. Lin, "After-the-fact leakage in public-key encryption," in *Proc. Theory Cryptography Conf.*, vol. 6597, Providence, RI, USA, 2011, pp. 107–124.

[7] S. Dziembowski and S. Faust, "Leakage-resilient cryptography from the inner-productextractor," in *Proc. 17th Int. Conf. Theory Appl. Cryptol. Informat. Security*, vol. 7073, 2011, pp. 702–721.

[8] J. Li, Y. Guo, Y. Lu, and Y. Zhang, "Continuous leakage-resilient certificate-based encryption," *Inform. Sci.*, vol. 355–356, pp. 1–14, 2016.

[9] E. Boyle, G. Segev, and D. Wichs, "Fully leakage-resilient signatures," *J. Cryptol.*, vol. 26, no. 3, pp. 89–108, 2011.

[10] A. Faonio, J. B. Nielsen, and D. Venturi, "Mind your coins: Fully leakage-resilient signatures with graceful degradation," in *Proc. 42nd Int. Colloq. Automata, Lang., Program.*, Kyoto, Japan, 2015, pp. 456–468.

[11] J. Nielsen, D. Venturi, and A. Zottarel, "Leakage-resilient signatures with graceful degradation," in *Proc. Public-Key Cryptography*, Buenos Aires, Argentina, 2014, pp. 362–379.

[12] Y. J. Pi, Q. L. Xu, P. Liu, and X. Y. Hu, "Leakage-resilient signature against related-key attacks," *J. Comput. Inform. Syst.*, vol.11, no.23, pp.8807–8817, 2015.

[13] Y. Ishai, E. Kushilevitz, X. Li, R. Ostrovsky, M. Prabhakaran, A. Sahai, and D. Zuckerman, "Robust pseudorandom generators," in *Proc. 40th Int. Colloq. Automata, Lang., Program.*, Riga, Latvia, 2013, pp. 576–588.

[14] E. Boyle, S. Goldwasser, A. Jainet, and Y. T. Kalai, "Multiparty computation secure against continual memory leakage," in *Proc. 44th Annu. ACM Symp. Theory Comput.*, New York, NY, USA, 2012, pp. 1235–1254.

[15] D. Dana, F. H. Liu, and H. S. Zhou, "Leakage-resilient circuit revisited-optimal number of computing components without leak-free hardware," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Tech.*, Sofia, Bulgaria, 2015, pp. 131–158.

[16] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Proc. Annu. Int. Cryptol. Conf.*, vol. 773, Santa Barbara, CA, USA, 1994, pp. 232–249.

[17] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Tech.*, vol. 2045, Innsbruck, Austria, 2001, pp. 453–474.

[18] B.A. LaMacchia, K. Lauter, . A Mityagin, "Stronger security of authenticated key exchange," in *Proc. Int. Conf. Provable Security*, vol. 4784, Wollongong, Australia, 2007, pp. 1–16.

[19] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, and W. Paul, "Lest we remember: Cold boot attacks on encryption keys," *Commun. ACM*, vol. 52, no. 5, pp. 91–98, 2009.

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

[21] J. Alawatugoda, C. Boyd, and D. Stebila, "Continuous After-the-Fact Leakage-Resilient Key Exchange," in *Proc. Australasian Conf. Informat. Security Privacy*, Wollongong, Australia, 2014, pp. 258–273.

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

[23] J. Alawatugoda, D. Stebila, and C. Boyd, "Continuous after-the-fact leakage-resilient eCK-secure key exchange," in *Proc. IMA Cryptography Coding*, Oxford, U.K., 2015, pp. 277–294.

[24] R. Chen, Y. Mu, G. Yang, W. Susilo, and F. C. Guo, "Strongly Leakage-Resilient Authenticated Key Exchange," in *Proc. Cryptographers' Track RSA Conf.*, San Francisco, CA, USA, 2016, pp. 19–36.

[25] J. Alwen, Y. Dodis, M. Naor, G. Segev, S. Walfish, and D. Wichs, "Public-key encryption in the bounded-retrieval model," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Tech.*, French Riviera, France, 2010, pp. 113–134.

[26] H. Krawczyk. (Apr. 2008). On extract-then-expand key derivation functions and an HMAC based KDF [Online]. Available: http://webee.technion.ac.il/∼hugo/kdf/ kdf.pdf

[27] X. Cao, W. Kou, and X. Du, "A pairing-free identity-based authenticated key agreement scheme with minimal message exchanges," *Inform. Sci.*, vol. 180, no. 15, pp. 2895–2903, 2010.

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

[29] M. Xie and L. Wang, "One-round identity-based key exchange with Perfect Forward Security," *Inform. Process. Lett.*, vol. 112, nos. 14/15, pp. 587–591, 2012.

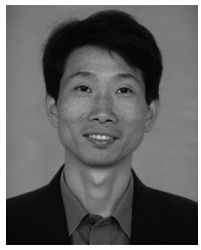
[30] T. Pandit, R. Barua, and S. Tripathy, "eCK secure single round ID-based authenticated key exchange protocols with master perfect forward secrecy," in *Proc. Network Syst. Security*, Xi'an, China, 2014, pp. 435–447.
[31] L. Ni, G. Chen, J. Li, and Y. Hao, "Strongly secure identity-based authenticated key agreement protocols without bilinear pairings," *Informat. Sci.*, vols. 367/368, pp. 176–193, 2016.
[32] I. Elashry, Y. Mu, and W. Susilo, "A resilient identity-based authenticated key exchange protocol," *Security Commun. Netw.*, vol. 8, no. 13, pp. 2279–2290, 2016.
[33] M. Toorani, "On continuous after-the-fact leakage-resilient key exchange," in *Proc. 2nd Workshop Cryptography Security Comput. Syst.*, Amsterdam, The Netherlands, 2015, pp. 31–35.
[34] Z. Yang and S. Q. Li, "On security analysis of an after-the-fact leakage resilient key exchange protocol," *Inform. Process. Lett.*, vol. 116, no. 1, pp. 33–40, 2016.
[35] S. Chakraborty, G. Paul, and C. P. Rangan. (Dec. 2016). Flaw in the security analysis of leakage-resilient authenticated key exchange protocol from CT-RSA 2016 and restoring the security proof [Online]. Available: http://eprint.iacr. org/2016/862.pdf
[36] S. S. M. Chow, Y. Dodis, Y. Rouselakis, and B. Waters, "Practical leakage-resilient identity-based encryption from simple assumptions," in *Proc. 17th ACM Conf. Comp. Commun. Security*, Chicago, IL, USA, 2010, pp. 152–161.

Ou Ruan received the Ph.D. degree from the College of Information Security, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2013.

He is an Associate Professor with the School of Computer Sciences, Hubei University of Technology, Wuhan, China. His research interests include leakage-resilient cryptography, secure computations, and network security.

Yuanyuan Zhang received the M.S. and Ph.D. degrees in applied mathematics from Wuhan University, Wuhan, China, in 2012 and 2015, respectively.

She is currently a Lecturer in the School of Computer Sciences, Hubei University of Technology, Wuhan. Her research interests include cloud computing security and cryptographic protocol.



Mingwu Zhang is a Professor with the School of Computer Sciences, Hubei University of Technology (HBUT), Wuhan, China. From August 2010 to August 2012, he was a JSPS Postdoctoral Fellow of the Japan Society of Promotion Sciences, Institute of Mathematics for Industry, Kyushu University, Fukuoka, Japan. He is the Director of Institute of Data Security and Privacy Preservation of HBUT. His research interests include cryptography technology for networks, secure computations, privacy preservations, etc.

Jing Zhou received the M.S. degree from the Wuhan University of Technology, Wuhan, China, in 2005.

She is currently a Lecturer with the School of Computer Sciences, Hubei University of Technology, Wuhan. Her research interests include network security and cryptography technology for networks.

Lein Harn received the Ph.D. degree in electrical engineering from the University of Minnesota, Minneapolis, MN, USA, in 1984.

He is a Professor with the Department of Computer Science Electrical Engineering, University of Missouri, Kansas City, MO, USA.His research interests include cryptography, network security, and wireless communication security.