Adaptively-Secure Authenticated Key Exchange Protocol in Standard Model

Mojahed Mohamed[1,2], Xiaofen Wang[1], and Xiaosong Zhang[1]
(Corresponding author: Mojahed Ismail Mohamed)
School of Computer Science and Engineering, University of Electronic Science and Technology of China[1]
4 Jianshe North Rd 2nd Section, Chenghua Qu, Chengdu Shi, Sichuan Sheng 610051, China
Department of Electronic Engineering, Karary University[2]
(Email: mmmmoj@hotmail.com)
(Received Nov. 17, 2016; revised and accepted Feb. 21, 2017)

Abstract

Design a Secure Authenticated Key Exchange (AKE) protocol is a wide research area. Many works have been done in this field and remain few open problems. Design an AKE-secure without NAXOS approach is remaining as an open problem. NAXOS approach [18] is used to hide the ephemeral secret key from an adversary even if the adversary in somehow may obtain the ephemeral secret key. Using NAXOS approach will cause two main drawbacks, (1) leaking of the static secret key which will be used in computing the exponent of the ephemeral public key. (2) Maximize of using random oracle when applying to the exponent of the ephemeral public key and session key derivation. Another open problem is designing an AKE-secure in the standard model without relying on Pseudo-Random Function with Pairwise-Independent Random Sources. In this paper, we present a general construction for AKE-secure protocol from the projective hash family secures under hard subset membership problem in the standard model. We also give an instantiation of our protocol from DDH with a novel security proof from games sequences tool introduced by [24]. We show the efficiency of our protocol compares to other similar AKE protocol.

Keywords: AKE; Decision Diffie-Hellman Assumption; eCK Model; Hash Proof System; NAXOS' Approach; Smooth Projective Hash Function

1 Introduction

An Authenticated Key Exchange Protocol is a primitive cryptographic notion which enables two parties after exchanging individual messages to agree on a symmetric shared key used later to secure the channel used between them. The authentication problem deals with restraining adversary that actively controls the communication links used by legitimated parties. They may modify and delete messages in transit, and even inject false one or may control the delays of messages.

Bellare nad Rogaway [1] formalized the security of KE protocols in the realistic setting of synchronal sessions running in a network controlled by the adversary. Their work focused on the shared-key model of authentication while other works [2, 3] expand the techniques to include public-key setting.

Canetti-Krawczyk [5] provides Adopted model for [25] with extraction of construction of secure sessions.

LaMacchia, Lauter and Mityagin [18] Presents significant security model for Authenticated Key Exchange (AKE) protocols which it is extending to Canetti-Krawczyk model. This model capture attacks resulting from leakage of ephemeral and long-term secret keys defined by an experiment in which the adversary is given many corruption power for various key exchange sessions and most solve a challenge on a test session. Moreover, This model doesn't give an adversary capability to break an AKE protocol trivially.

Recently a variant of eCK model used in literatures (e.g., [8, 9, 17, 26]). The difference is those models using StateReveal query instead of EphemeralKeyReveal query in the eCK model, which models maximum exposure.

Bresson *et al.* [4] used a secure device together with an untrusted host machine to attain the existing gap between formal models and effective security. A secure device may use to store long-term keys and, at least, be able to perform some mathematical functions (such as addition, modulo, and exponentiation) which are necessary to achieve cryptographic operations. In such way, we could assume that ephemeral keys and intermediate states generated on host machine are liable to StateReveal attacks to model the maximum state leakage attack (MSL). Although there might exist some side-channel attacks (such as [16]) against the secure device, we assume it works as a black-box to avoid the leakage of internal states [26].

When using the secure device then the security model would equal to a model without StateReveal query. How-

ever, the security result of a protocol analyzed with such implementation scenario must be weaker than that in a case allowing leakage of states. In contrast, our goal is to define the maximum states that can be leaked. The secure devices have limited resources which may cause performance bottleneck of systems.

In this paper, we will use eCK for several security attributes, as resistance to key compromise impersonation attack (KCI), leakage of secret states and chosen identity and public key attack (CIDPK) and provide of weak perfect forward secrecy (wPFS). Also, we will use StateReveal query instead of EphemeralKeyReveal query to models maximum disclosure.

Motivating Problem

(1) In AKE, still remind few open problems. Is it essentially to use NAXOS trick [18] in designing the AKE protocols. This method is used to hide the ephemeral secret key from an adversary even if the adversary in somehow may obtain the ephemeral secret key. Design AKE-secure protocol without NAXOS trick will achieve two goals: (i) To reduce the risk of leaking the static private key, since the derivation of the ephemeral public key is independent of the static private key. This method in contrast to protocols that use the NAXOS' approach. (ii) Minimize the utilization of the random oracle, by applying it only to the session key derivation. Kim, Minkyu, Atsushi Fujioka, and Berkant Ustaolu [15, 19] proposed a two strongly secure authenticated key exchange protocols without NAXOS approach, one of their protocol supposed to be secure under the GDH assumption and the other under the CDH assumption in random oracle model. Mohamed *et al.* [20] designed a protocol without NAXOS approach but secure in RO model. Recently, Daisuke *et al.* [12] presents an eCK secure AKE protocol without using the NAXOS trick, but they still rely on Pseudo-Random Function with Pairwise-Independent Random Sources. In another hand, we can find several protocols designed with NAXOS trick and supposed to be secure in a different manner of definition. Those protocols should answer the question how to implement the NAXOS trick securely. In the original implementation, the hash function will be used as in original NAXOS protocol [18]. In some design, we can found the exponent of the ephemeral secret key hidden with a particular kind of PRF [9, 22]. In some scenario, secure device may use to cover up the ephemeral secret key. The remain challenges are the limitation of computational capability of those devices and limitation of resources. (2) Design AKE-secure without NAXOS trick in the standard model. As mentioned above that secure device might not be the ideal solution because of its short in storage capacity and computational resources. (3) Design adaptive AKE-secure using a hash proof system. Cramer and Shop [7] invented the universal hash proof system. It is a particular kind of non-interactive zero-knowledge proof system for the language. They show how to construct an efficient public-key encryption schemes secure against adaptive chosen ciphertext attack in the standard model given an efficient universal hash proof system. [10] presented a general framework for password-based authenticated key exchange protocols using modified smooth projective hash function.Katz *et al.* [13] introduced password-based authenticated key exchange protocol. Their protocol uses a CCA-secure labeled public-key encryption scheme (Gen, Enc, Dec), and a smooth projective hash function. That protocol does not consider the attack of StaticKeyReveal, SessionKeyReveal or EstablishParty, which causes static secret keys leakage and session keys leakage. The protocol follows plan security model which not consider the session freshness definition and needed to be shared password with the public keys.

Contributions

In this paper, we present a concrete and practical AKE-secure protocol which is eCK secure under Decisional Diffie-Hellman assumption in the standard model. Our protocol does not rely on NAXOS trick or Pseudo-Random Function with Pairwise-Independent Random Sources as [12, 22]. We give a generic construction for AKE-secure protocol from the projective hash family secure under hard subset membership problem in the standard model. We also provide an instantiation of our protocol from DDH with a novel security proof from games sequences tool introduced by [24]. We show the efficiency of our protocol compares to other similar AKE protocol.

Organization

Section 2 reviews security definitions, general assumptions and states the hard problem. Section 3 gives brief for the eCK model. Section 4 proposes a Generic adaptively-secure AKE Construction from HPS. Section 5 presents an instantiation from the DDH Assumption for paradigm designed in Section 4. And finally we draws the conclusion in Section 6.

2 Preliminaries

Notation

Let [n] denote the set $\{1, ..., n\}$. Let $k \in \mathbb{N}$ denote the security parameter and 1^k denote the string of k ones. $s \leftarrow_{\$} S$ denotes picking an element s uniformly random from S. $y \leftarrow \mathcal{A}(x)$ denotes runnung $\mathcal{A}$ with input x and assigning y as the result. *logs* denotes logarithm s for base 2. Let $\Delta(X;Y)$ be the *statistical distance* between two random variables X and Y having a common domain $\mathcal{X}$.

2.1 Randomness Extractor

Entropy

is a measurement of unpredictable of information content.

Definition 1 (Entropy). *entropy* $\mathrm{H}(.)$ *of a discrete random variable* X *with possible values* $\{x_1, ..., x_n\}$ *and probability mass function* $Pr[X]$ *defined as:*

$$\mathrm{H}(X) = \mathbb{E}[-ln(Pr[X])] = -\sum Pr[x_i] \log Pr[x_i]$$

Min-entropy

The min-entropy of a distribution X (denoted $\mathrm{H}_\infty(X)$), is the largest real number k such that $\Pr[X = x] \leq 2^{-k}$ for every x in the range of X. In essence, this measures how likely X is to take its most likely value, giving a worst-case bound on how random X appears. Letting U_ℓ denote the uniform distribution over $\{0,1\}^\ell$, clearly $\mathrm{H}_\infty(U_\ell) = \ell$.

For an $n - bit$ distribution X with min-entropy k, we say that X is an (n, k) distribution.

Definition 2 (Randomness Extractor). *Let* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a function that takes as input a sample from an* (n, k) *distribution* X *and a* d*-bit seed from* U_d*, and outputs an* m*-bit string.* Ext *is a* (k, ε)*-extractor, if for all* (n, k) *distributions* X*, the output distribution of* Ext *is* ε*-close to* U_m*.*

Definition 3 ((k, ϵ)-Strong Extractor). *Let* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k, ϵ)*-strong extractor such that for any* x *distributed over* X *that has min-entropy* k *and for any seed* $s \leftarrow_\$ \{0,1\}^d$ *which is chosen uniformly at random from* (d, k) *distribution and for any value* $r \leftarrow_\$ \{0,1\}^m$ *which is chosen uniformly at random from* (m, k) *distribution, the two distributions* $\langle s, Ext(s, x)\rangle$ *and* $\langle s, r\rangle$ *have statistical distance at most* ϵ

$$\frac{1}{2} \sum_{y \in (m,k)} |Pr[Ext(s, x) = y] - Pr[r = y]| = \epsilon$$

Some good results on key derivation and randomness extraction can be also found in [6].

2.2 Pseudo-Random Functions

A pseudo-random function PRF is a deterministic functions introduced by Goldreich, Goldwasser and Micali [11]. Let $\mathsf{PRF} : \mathcal{K}_{\mathsf{PRF}} \times \mathcal{D}_{\mathsf{PRF}} \to \mathcal{R}_prf$ denote a family of deterministic functions, where $\mathcal{K}_{\mathsf{PRF}}$ is the key space, $\mathcal{D}_{\mathsf{PRF}}$ is the domain and $\mathcal{R}_prf$ is the range of PRF for security parameter λ. Let $RL = \{(x_1, y_1), ..., (x_q, y_q)\}$ be a list which is used to record bit strings formed as tuple $(x_i, y_i) \in (\mathcal{D}_{\mathsf{PRF}}, \mathcal{R}_prf)$ where $1 \leq i \leq q$ and $q \in \mathbb{N}$. In RL each x is associated with a y. Let $RF : \mathcal{D}_{\mathsf{PRF}} \to \mathcal{R}_prf$ be a stateful uniform random function, which can be executed at most a polynomial number of q times and keeps a list RL for recording each invocation. On input a message $x \in \mathcal{D}_{\mathsf{PRF}}$, the function $RF(x)$ is executed as follows:

- If $x \in RL$, then return corresponding $y \in RL$.
- Otherwise return $y \leftarrow_\$ \mathcal{R}_{\mathsf{PRF}}$ and record (x, y) into RL.

Definition 4. *We say that* PRF*is a* $(q, t, \epsilon_{\mathsf{PRF}})$*-secure pseudo-random function family, if it holds that*

$$\left|\Pr\left[EXP^{ind-cma}_{\mathsf{PRF},\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2}\right| \leq \epsilon_{\mathsf{PRF}}$$

for all adversaries $\mathcal{A}$ *that makes a polynomial number of oracle queries* q *while running in time at most* t *in the following experiment:*

$EXP^{ind-cma}_{\mathsf{PRF},\mathcal{A}}(\lambda)$	$\mathcal{F}(b, x)$
$k \leftarrow_\$ \{0,1\}^\lambda$,	**for** $i = 1..q, q \in \mathbb{N}$ **do**
$b \leftarrow_\$ \{0,1\}$	**if** $x \notin \mathcal{D}_{\mathsf{PRF}}$**return** $\perp$
$b' \leftarrow \mathcal{A}^{\mathcal{F}(b,.)}(\lambda)$	$z_0 = \mathsf{PRF}(k, x), z_1 = RF(x)$
return $b = b'$	**return** z_b

2.3 Hash Proof System

Cramer and Shoup [7] introduced a novel security notion called universal hash proof system. They showed that given this system how to construct efficient public-key encryption schemes secure against adaptive chosen ciphertext attack in the standard model. In another hand, we can describe the hash proof system as a key encapsulation mechanism (KEM) [14, 23] with special algebraic properties.

Universal Hashing

Let $\mathcal{SK}, \mathcal{PK}, \mathcal{K}, \mathcal{C}$ and $\mathcal{V}$ be non-empty finite sets, represents secret keys, public keys, encapsulated keys, ciphertext set and valid ciphertext set respectively. Where $\mathcal{V} \subset \mathcal{C}$. Let $\Lambda_{sk} : \mathcal{C} \to \mathcal{K}$ be an indexed hash function indexed by sk. We call Λ_{sk} *projective* if there exists a projection $\mu : \mathcal{SK} \to \mathcal{PK}$ where $\mu(sk) \in \mathcal{PK}$ defines the action of Λ_{sk} over the subset $\mathcal{V}$. That is, for every $C \in \mathcal{V}$, the value $K = \Lambda_{sk}(C)$ is uniquely determined by $\mu(sk)$ and C. In contrast, nothing is guaranteed for $C \in \mathcal{C} \setminus \mathcal{V}$, and it may not be possible to compute $\Lambda_{sk}(C)$ from $\mu(sk)$ and C.

Definition 5 (Universal). *A projective hash function* Λ_{sk} *is* ϵ*-universal, if for all* $pk, C \in \mathcal{C} \setminus \mathcal{V}$*, and all* $K \in \mathcal{K}$*, it holds that* $\Pr[\Lambda_{sk}(C) = K|(pk, C)] \leq \epsilon$*, in other word we can say*

$$\Delta[(pk, \Lambda_{sk}(C)); (pk, K)] \leq \epsilon$$

where in the above $pk = \mu(sk)$ *for* $sk \leftarrow_\$ \mathcal{SK}$ *and* $K \leftarrow_\$ \mathcal{K}$*.*

Lemma 1. *Assume that* $\Lambda_{sk} : \mathcal{C} \to \mathcal{K}$ *is an* ϵ*-universal projective hash function. Then, for all* pk *and* $C \in \mathcal{C} \setminus \mathcal{V}$*, it holds that* $H_\infty(\Lambda_{sk}(C)|(pk, C)) \geq \log 1/\epsilon$*, where* $sk \leftarrow \mathcal{SK}$ *with* $pk = \mu(sk)$*.*

Hash Proof System

A hash proof system **HPS = (Gen, Pub, Priv)** consists of three algorithms. The parameter generation algorithm

$HPS.Gen(1^k)$ generates parameterized instances of the form $params = (group, \mathcal{K}, \mathcal{C}, \mathcal{V}, \mathcal{SK}, \mathcal{PK}, \Lambda_{(.)} : \mathcal{C} \rightarrow \mathcal{K}, \mu : \mathcal{SK} \rightarrow \mathcal{PK})$, where group may contain additional structural parameters. The public evaluation algorithm $HPS.Pub(pk, C, w)$ takes as input a projective public key $pk = \mu(sk)$, a valid ciphertext $C \in \mathcal{V}$ and a witness w of the fact that $C \in \mathcal{V}$, and computes the encapsulated key $K = \Lambda_{sk}(C)$. The private evaluation algorithm $HPS.Priv(sk, C)$ takes a secret key sk and a ciphertext $C \in \mathcal{V}$ as input, and returns the encapsulated key $K = \Lambda_{sk}(C)$ without knowing a witness. We assume that μ and $\Lambda_{()}$ are efficiently computable.

We say that a hash proof system is universal if for all possible outcomes of $HPS.Gen(1^k)$ the underlying projective hash function is ϵ-almost universal for negligible $\epsilon(k)$. Furthermore, we say that a hash proof system is perfectly universal if $\epsilon(k) = 0$.

Subset Membership Problems

The subset membership problem associated with a HPS suggests that a random valid ciphertext $C_0 \leftarrow_{\$} \mathcal{V}$ and a random invalid ciphertext $C_1 \leftarrow_{\$} \mathcal{C} \setminus \mathcal{V}$ are computationally indistinguishable. This is formally captured by a negligible advantage function $\mathsf{Adv}^{\mathrm{smp}}_{HPS,\mathcal{A}}(k)$ for all PPT adversary $\mathcal{A}$, where

$$\begin{aligned}\mathsf{Adv}^{\mathrm{smp}}_{HPS,\mathcal{A}}(k) \quad = \quad & |\Pr[\mathcal{A}(\mathcal{C}, \mathcal{V}, C_0) = 1 | C_0 \leftarrow_{\$} \mathcal{V}] \\ & - \Pr[\mathcal{A}(\mathcal{C}, \mathcal{V}, C_1) = 1 | C_1 \leftarrow_{\$} \mathcal{C} \setminus \mathcal{V}]\,|.\end{aligned}$$

Definition 6. *A hash proof system* ***HPS = (HPS.Gen, HPS.Pub, HPS.Priv)*** *is ϵ-universal if: (i) for all sufficiently large $k \in \mathbb{N}$ and for all possible outcomes of $HPS.Gen(1^k)$, the underlying projective hash function is $\epsilon(k)$-universal for negligible $\epsilon(k)$; (ii) the underlying subset membership problem is hard. Furthermore, a hash proof system is called perfectly universal if $\epsilon(k) = 1/|\mathcal{K}|$.*

2.4 The DDH Assumption

We assume a PPTalgorithm $\mathcal{G}(1^k)$ that takes as input 1^k and outputs a tuple of $\mathbb{G} = \langle q, G, g \rangle$, where G is a cyclic group of prime order q and g is a generator of G. The Decisional Diffie-Hellman (DDH) assumption holds iff

$$\mathsf{Adv}^{\mathrm{ddh}}_{G,\mathcal{D}}(k) = \left| \Pr[\mathcal{D}(g_1, g_2, g_1^r, g_2^r) = 1] - \Pr\left[\mathcal{D}(g_1, g_2, g_1^r, g_2^{r'}) = 1\right] \right|$$

is negligible in k for any PPT adversary $\mathcal{D}$, where $g_1 \leftarrow_{\$} G$, $g_2 \leftarrow_{\$} G$, $r \leftarrow_{\$} \mathbb{Z}_q$ and $r' \leftarrow_{\$} \mathbb{Z}_q \setminus \{r\}$.

3 Security Model

In this section, eCK model is outlined. In eCK model there are n different parties $P = P_1, , P_n$ running the KE protocol Π. Each party possesses a pair of long-term static (private/public) keys together with a corresponding certificate issued by a certificate authority. The protocol Π is executed between two parties say $\mathcal{A}$ and $\mathcal{B}$, whose static public key are A and B respectively. These two parties exchange their ephemeral public keys X and Y and obtain the same final session key.

Sessions

A party is activated by an outside call or an incoming message to execute the protocol Π. Each program of executing Π is modeled as an interactive probabilistic polynomial-time machine. We call a session an invocation of an instance of Π within a party. We assume that $\mathcal{A}$ is the session initiator and $\mathcal{B}$ is the session responder. Then $\mathcal{A}$ is activated by the outside call $(\mathcal{A}, \mathcal{B})$ or the incoming message $(\mathcal{A}, \mathcal{B}, Y)$. When activated by $(\mathcal{A}, \mathcal{B})$, $\mathcal{A}$ prepares an ephemeral public key X and stores a separate session state which includes all session-specific ephemeral information. The session identifier (denoted by sid) in $\mathcal{A}*$ is initialized with $(\mathcal{A}, \mathcal{B}, X, -, \mathcal{I})$. After $\mathcal{A}$ is activated by $(\mathcal{A}, \mathcal{B}, Y)$ (receiving an appropriate message from responder), the session identifier is updated to $(\mathcal{A}, \mathcal{B}, X, Y, \mathcal{I})$. Similarly, the responder $\mathcal{B}$ is activated by the incoming message $(\mathcal{B}, \mathcal{A}, X)$. When activated, $\mathcal{B}$ also prepares an ephemeral public key Y and stores a separate session state, and the corresponding session identifier is $(\mathcal{B}, \mathcal{A}, Y, X, \mathcal{R})$. A $(\mathcal{B}, \mathcal{A}, Y, X, \mathcal{R})$ (if it exists) is said to be matching to the session $(\mathcal{A}, \mathcal{B}, X, Y, \mathcal{I})$ or $(\mathcal{A}, \mathcal{B}, X, -, \mathcal{I})$. For a session $(\mathcal{A}, \mathcal{B}, *, *, role)$, $\mathcal{A}$ is called the owner of the session while $\mathcal{B}$ is called the peer of the session. We say sid is complete if there is no symbol "" in sid.

Adversaries

The adversary $\mathcal{M}$ is also modeled as a probabilistic polynomial-time machine. $\mathcal{M}$ controls the whole communications between parties by sending arbitrary messages to the intended party on behalf of another party and receiving the outgoing message from the communicating parties. To capture the possible attacks, $\mathcal{M}$ is allowed to make the following queries.

EstablishParty($\mathcal{U}$): $\mathcal{M}$ Registers an arbitrary party $\mathcal{U}$ not in P, whose static public key is on $\mathcal{M}$s own choice. We call this kind of new registered parties dishonest ($\mathcal{M}$ totally controls the dishonest parties), while the parties in P are honest. We require that when $\mathcal{M}$ makes such query, the certificate authority(CA) should verify that the submitted static public key is in the appropriate group (to avoid small subgroup attack) and the proof that $\mathcal{M}$ knows the corresponding static private key.

Send($\mathcal{A}$,m): $\mathcal{M}$ sends the message m to party $\mathcal{A}$. Upon invocation $\mathcal{A}$ by m, the adversary obtains the outgoing message of $\mathcal{A}$.

StateReveal(sid): $\mathcal{M}$ obtains the secret state stored in the session state of the session sid.

StaticKeyReveal(P_i): $\mathcal{M}$ learns the long-term static private key of an honest party P_i. In this case, P_i no longer seems honest.

SessionKeyReveal(sid): $\mathcal{M}$ obtains the session key for the session sid if the it has been accepted, otherwise $\mathcal{M}$ obtains nothing.

Experiment

$\mathcal{M}$ is given the set P of honest parties, and makes whichever queries he wants. The final aim of the adversary is to distinguish a session key from a random string of the same length. Thus $\mathcal{M}$ selects a complete and fresh session sid, and makes a special query *Test(sid)*. This query can be queried only once, and the session sid is called test session. On this query, a coin b is flipped, if $b = 1$ $\mathcal{M}$ is given the real session key held by sid, otherwise $\mathcal{M}$ is given a random key drawn from the key space at random. $\mathcal{M}$ wins the experiment if he guesses the correct value of b. Of course, $\mathcal{M}$ can continue to make the above queries after the *Test* query; however the test session should remain fresh throughout the whole experiment.

Definition 7 (Fresh session)**.** *Let sid be a complete session, owned by honest $\mathcal{A}$ with honest peer $\mathcal{B}$. If the matching session of sid exists, we let $\overline{sid}$ denote the session identifier of its matching session. sid is said to be fresh if none of the following events occurs:*

1) *$\mathcal{M}$ makes a **SessionKeyReveal(**sid**)** query or a **SessionKeyReveal(**$\overline{sid}$**)** query if $\overline{sid}$ exists.*
2) *If $\overline{sid}$ exists, $\mathcal{M}$ makes either of the following queries:*
   a. *Both **StaticKeyReveal(**$\mathcal{A}$**)** and **StateReveal(**sid**)**, or*
   b. *Both **StaticKeyReveal(**$\mathcal{B}$**)** and **StateReveal(**$\overline{sid}$**)**.*
3) *If $\overline{sid}$ does not exist, $\mathcal{M}$ makes either of the following queries:*
   a. *Both **StaticKeyReveal(**$\mathcal{A}$**)** and **StateReveal(**sid**)**, or*
   b. ***StaticKeyReveal(**$\mathcal{B}$**)**.*

The eCK security notion can be described now.

Definition 8 (eCK security)**.** *The advantage of the adversary $\mathcal{M}$ in the above experiment with respect to the protocol Π is defined as (b is the guessed value of coin by $\mathcal{M}$):*

$$Adv_{\Pi}^{AKE}(\mathcal{M}) = |2\Pr[b' = b] - 1| \quad (1)$$

The protocol Π is said to be secure if the following conditions hold:

1) *If two honest parties complete matching sessions, then they will both compute the same session key, except with a negligible probability.*
2) *The advantage of the adversary $\mathcal{M}$ is negligible.*

4 A Generic Adaptively-secure AKE Construction from HPS

In this section, we present a generic authenticated key exchange protocol from HPS.This protocol can be implemented to ensure eCK adaptive security.

4.1 Protocol Description

Parameters

Let $HPS = (HPS.Gen, HPS.Pub, HPS.Priv)$ be an ϵ_{hps}-universal hash proof system, where $HPS.Gen(1^{\lambda})$ generates instances of $params = (group, \mathcal{K}, \mathcal{C}, \mathcal{V}, \mathcal{SK}, \mathcal{PK}, \Lambda_{(.)}: \mathcal{C} \to \mathcal{K}, \mu : \mathcal{SK} \to \mathcal{PK})$. Let $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a $(\nu, \epsilon_{\mathsf{Ext}})$-Strong Extractor. Let $\mathsf{PRF} : \mathcal{K}_{\mathsf{PRF}} \times \mathcal{D}_{\mathsf{PRF}} \to \mathcal{R}_{\mathsf{PRF}}$ be a $(q, t, \epsilon_{\mathsf{PRF}})$-secure pseudo-random function family. We assume that $\epsilon_{\mathsf{PRF}}, \epsilon_{\mathsf{Ext}}$ and ϵ are negligible λ.

Key Generation

At beginning of the protocol, $HPS.Gen(1^{\lambda})$ will be run for once to generate the public parameter $(param)$. A party $\hat{A}$ picks $sk_{\hat{A}} \leftarrow_{\$} \mathcal{SK}$ and sets $pk_{\hat{A}} = \mu(sk)$. The public/secret key for the party $\hat{A}$ is $(sk_{\hat{A}}, pk_{\hat{A}})$. Similarly, a party $\hat{B}$ will set his public/secret keys as $(sk_{\hat{B}}, pk_{\hat{B}})$. We assume this protocol executed between party $\hat{A}$ and party $\hat{B}$, where party $\hat{A}$ is the initiator and party $\hat{B}$ is the responder.

Messages Exchange

The party $\hat{A}$ chooses $C_{\hat{A}} \leftarrow \mathcal{V}$ with witness $\omega_{\hat{A}}$, a random seed $s_{\hat{A}} \leftarrow_{\$} \{0,1\}^d$ and then computes

$$k_{\hat{A}} \leftarrow HPS.Pub(pk_{\hat{B}}, C_{\hat{A}}, \omega_{\hat{A}}), \Phi_{\hat{A}} \leftarrow \mathsf{Ext}(k_{\hat{A}}, s_{\hat{A}})$$

Then sends $(\hat{B}, \hat{A}, C_{\hat{A}}, \Phi_{\hat{A}}, s_{\hat{A}})$ to party $\hat{B}$. Simulatly, the part $\hat{B}$ will follow the same steps and sends $(\hat{A}, \hat{B}, C_{\hat{B}}, \Phi_{\hat{B}}, s_{\hat{B}})$ to party $\hat{A}$.

Upon receiving $(\hat{B}, \hat{A}, C_{\hat{A}}, \Phi_{\hat{A}}, s_{\hat{A}})$, party $\hat{B}$ uses his secret key to get $k'_{\hat{A}} \leftarrow HPS.Priv(sk_{\hat{B}}, C_{\hat{A}})$, then computes $\Phi'_{\hat{A}} \leftarrow \mathsf{Ext}(k'_{\hat{A}}, s_{\hat{A}})$. The party $\hat{B}$ checks $\Phi_{\hat{A}} = \Phi'_{\hat{A}}$, if not then halt. Otherwise, the party $\hat{B}$ computes the session key.

Session Key

The party $\hat{B}$ compute his session key as following:

1) $K \leftarrow k'_{\hat{A}} \oplus k_{\hat{B}}$.
2) $seed \leftarrow \hat{A} \parallel \hat{B} \parallel pk_{\hat{A}} \parallel pk_{\hat{B}} \parallel C_{\hat{A}} \parallel C_{\hat{B}} \parallel k'_{\hat{A}} \parallel k_{\hat{B}}$.
3) $k_s \leftarrow \mathsf{PRF}(K, seed)$.

The correctness of the above protocol follows from the correctness of the underlying hash proof system.

$\hat{A}$		$\hat{B}$
$sk_{\hat{A}} \leftarrow_{\$} \mathcal{SK}; pk_{\hat{A}} \leftarrow \mu(sk)$		$sk_{\hat{B}} \leftarrow_{\$} \mathcal{SK}; pk_{\hat{B}} \leftarrow \mu(sk)$
$C_{\hat{A}} \leftarrow \mathcal{V}$ **with witness** $\omega_{\hat{A}}$		$C_{\hat{B}} \leftarrow \mathcal{V}$ **with witness** $\omega_{\hat{B}}$
$s_{\hat{A}} \leftarrow_{\$} \{0,1\}^d$		$s_{\hat{B}} \leftarrow_{\$} \{0,1\}^d$
$k_{\hat{A}} \leftarrow HPS.Pub(pk_{\hat{B}}, C_{\hat{A}}, \omega_{\hat{A}})$		$k_{\hat{B}} \leftarrow HPS.Pub(pk_{\hat{A}}, C_{\hat{B}}, \omega_{\hat{B}})$
$\Phi_{\hat{A}} \leftarrow \mathsf{Ext}(k_{\hat{A}}, s_{\hat{A}})$		$\Phi_{\hat{B}} \leftarrow \mathsf{Ext}(k_{\hat{B}}, s_{\hat{B}})$
$sid = (\hat{A}, \hat{B}, C_{\hat{A}}, -, \mathcal{I})$		$\overline{sid} = (\hat{B}, \hat{A}, C_{\hat{B}}, -, \mathcal{R})$
	$\xrightarrow{(\hat{B}, \hat{A}, C_{\hat{A}}, \Phi_{\hat{A}}, s_{\hat{A}})}$	
	$\xleftarrow{(\hat{A}, \hat{B}, C_{\hat{B}}, \Phi_{\hat{B}}, s_{\hat{B}})}$	
$k'_{\hat{B}} \leftarrow HPS.Priv(sk_{\hat{A}}, C_{\hat{B}})$		$k'_{\hat{A}} \leftarrow HPS.Priv(sk_{\hat{B}}, C_{\hat{A}})$
$\Phi'_{\hat{B}} \leftarrow \mathsf{Ext}(k'_{\hat{B}}, s_{\hat{B}})$		$\Phi'_{\hat{A}} \leftarrow \mathsf{Ext}(k'_{\hat{A}}, s_{\hat{A}})$
$\Phi_{\hat{B}} =^{?} \Phi'_{\hat{B}}$		$\Phi_{\hat{A}} =^{?} \Phi'_{\hat{A}}$
$sid = (\hat{A}, \hat{B}, C_{\hat{A}}, C_{\hat{B}}, \mathcal{I})$		$\overline{sid} = (\hat{B}, \hat{A}, C_{\hat{B}}, C_{\hat{A}}, \mathcal{R})$
$K \leftarrow k_{\hat{A}} \oplus k'_{\hat{B}}$		$K \leftarrow k'_{\hat{A}} \oplus k_{\hat{B}}$
$seed \leftarrow \hat{A} \parallel \hat{B} \parallel pk_{\hat{A}} \parallel pk_{\hat{B}} \parallel C_{\hat{A}} \parallel C_{\hat{B}} \parallel k_{\hat{A}} \parallel k'_{\hat{B}}$		
$k_s \leftarrow \mathsf{PRF}(K, seed)$		$k_s \leftarrow \mathsf{PRF}(K, seed)$

Figure 1: A generic adaptively-secure AKE construction from HPS

4.2 Security Analysis

Apparently, we use a Hash Proof System (HPS) to generate an encapsulated key k as an idea in [23], then we used that key to derive the PRF key to obtain the session key. We used the extractor to prevent the key k leakage.

Theorem 1. *Assuming that HPS is an ϵ_{hps}-universal hash proof system, $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a $(\nu, \epsilon_{\mathsf{Ext}})$-Strong Extractor, $\mathsf{PRF} : \mathcal{K}_{\mathsf{PRF}} \times \mathcal{D}_{\mathsf{PRF}} \to \mathcal{R}_{\mathsf{PRF}}$ is a $(q, t, \epsilon_{\mathsf{PRF}})$-secure pseudo-random function family. Then the proposed protocol is eCK-secure in the sense of Deffinition 8.*

The proof of above theorem could be found in Appendix A.

5 Instantiation from the DDH Assumption

We organized this section as follows. In Section 5.1, we show how to construct a hash proof system from the Decisional Diffie-Hellman (DDH) assumption. We follow [7, 23] in the instantiation of our protocol from DDH assumption. In Section 5.2, we apply the construction in Section 4 to a building block and obtain an adaptively DDH-based secure AKE scheme, depicted in Figure 2. In this section, we will show a comparison of our scheme with some existing AKE-secure scheme.

5.1 A DDH-Based HPS

Let $\langle q, G, g\rangle \leftarrow \mathcal{G}(1^\lambda)$ and let $g_1, g_2 \leftarrow_{\$} G$. Let $\Gamma : G \to \mathbb{Z}_q$ be an efficient injective mapping.

For any $u = (u_1, ..., u_n) \in G^n, n \in \mathbb{N}$ let $\hat{\Gamma}(u) = (\Gamma(u_1), ..., \Gamma(u_n)) \in \mathbb{Z}_q^n$. Obviously, $\hat{\Gamma}$ is also an injection. We will set up the a parameter *param* of the hash proof system mentioned in Section 2.3 as follows.

- $group = \langle q, G, g_1, g_2, n\rangle, \mathcal{C} = G \times G, \mathcal{V} = \{(g_1^r, g_2^r) : r \in \mathbb{Z}_q\}$ with witness $\omega = \mathbb{Z}_q$.
- $\mathcal{K} = \mathbb{Z}_q^n, \mathcal{SK} = (\mathbb{Z}_q \times \mathbb{Z}_q)^n, \mathcal{PK} = G^n$.
- For all $sk = (x_{i,1}, x_{i,2})_{i\in[n]} \in \mathcal{SK}$ we define $pk = (pk_i)_{i\in[n]} = \mu(sk) = (g_1^{x_{i,1}} g_2^{x_{i,2}})_{i\in[n]}$.
- For all $C = (u_1, u_2) \in \mathcal{C}$ we define $\Lambda_{sk}(C) = \hat{\Gamma}((u_1^{x_{i,1}} u_2^{x_{i,2}})_{i\in[n]})$.
- HPS.Gen(1^λ): Will generate sk and pk as mentioned above.
- HPS.Pub$(pk, C, r) = \hat{\Gamma}(pk_1^r, ..., pk_n^r)$ for all $C = (g_1^r, g_2^r) \in \mathcal{V}$ with witness $r \in \mathbb{Z}_q$.
- HPS.Priv$(sk, C) = \Lambda_{sk}(C) = \hat{\Gamma}((g_1^{rx_{i,1}} u_2^{rx_{i,2}})_{i\in[n]})$ for all $C = (g_1^r, g_2^r) \in G^2$.
- Correctness follow since

$$\begin{aligned}\hat{\Gamma}(pk_i^r)_{i\in[n]} &= \hat{\Gamma}(pk_1^r, ..., pk_n^r) = \hat{\Gamma}(g_1^{rx_{1,1}} g_2^{rx_{1,2}}, ..., g_1^{rx_{n,1}} g_2^{rx_{n,2}}) \\ &= \hat{\Gamma}(u_1^{x_{1,1}} u_2^{x_{1,2}}, ..., u_1^{x_{n,1}} u_2^{x_{n,2}}) \\ &= \Lambda_{sk}(C).\end{aligned}$$

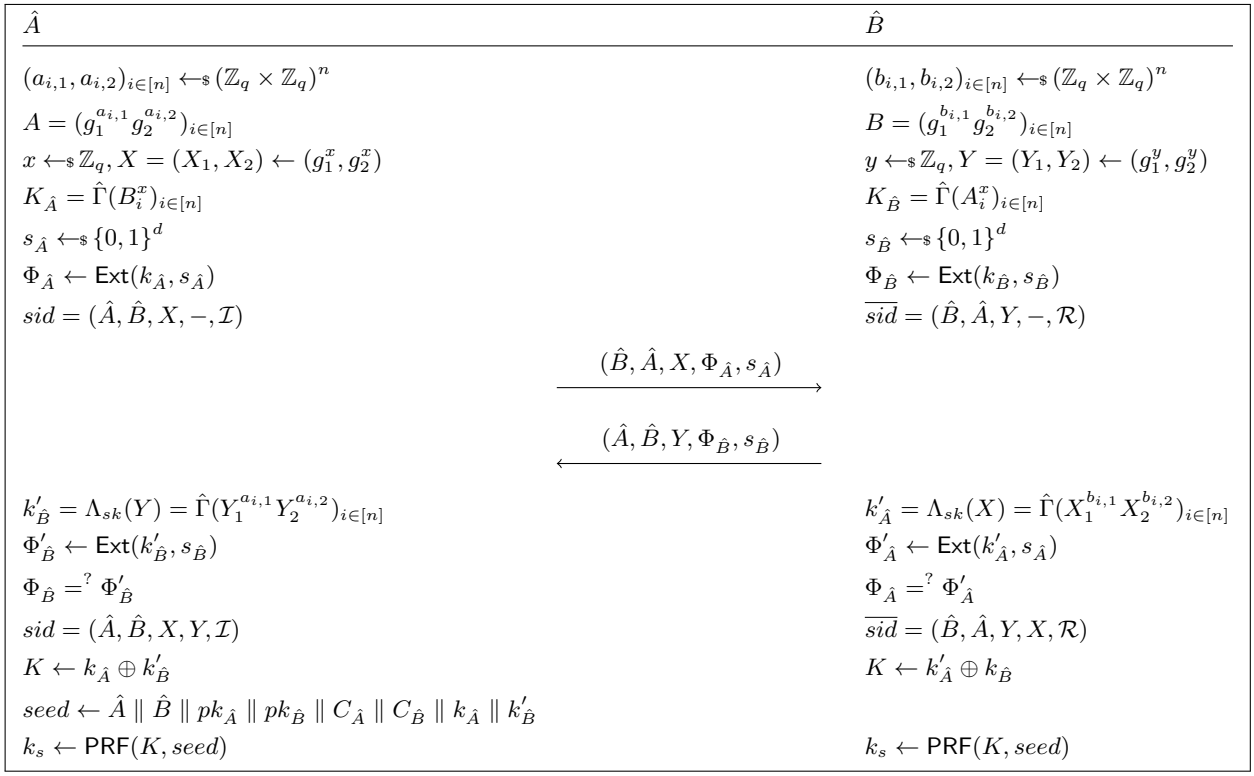

Figure 2: A DDH-based adaptively-secure AKE construction from HPS

Theorem 2. *The above system **HPS**, which contains of following algorithms (HPS.Gen,HPS.Pub,HPS.Priv) is a ϵ-universal HPS for $\mathcal{V}$.*

The proof of above theorem could be found in Appendix A.

5.2 The DDH-Based Instantiation AKE Scheme from Scheme in Section 4

Parameters

Let $\mathbb{G} = \langle q, G, q\rangle$. Let $n \in \mathbb{N}$. Let HPS is ϵ-universal hash proof system described in above. Let $\mathsf{Ext} : \mathbb{Z}_q^n \times \{0,1\}^d \to \{0,1\}^m$ be a $(\nu, \epsilon_{\mathsf{Ext}})$-Strong Extractor. Let $\mathsf{PRF} : \mathbb{Z}_q^n \times \mathcal{D}_{\mathsf{PRF}} \to \mathcal{R}_{\mathsf{PRF}}$ be a $(q, t, \epsilon_{\mathsf{PRF}})$-secure pseudo-random function family. We obtain a DDH-based scheme as follows.

Key Generation

A party $\hat{A}$ picks $(a_{i,1}, a_{i,2})_{i\in[n]} \leftarrow_{\$} (\mathbb{Z}_q \times \mathbb{Z}_q)^n$ and sets $A = (g_1^{a_{i,1}} g_2^{a_{i,2}})_{i\in[n]}$. The public/secret key for the party $\hat{A}$ is $((a_{i,1}, a_{i,2})_{i\in[n]}, A = (A_i)_{i\in[n]})$. Similarly, a party $\hat{B}$ will set his public/secret keys as $((b_{i,1}, b_{i,2})_{i\in[n]}, B = (B_i)_{i\in[n]})$. We assume this protocol executed between party $\hat{A}$ and party $\hat{B}$, where party $\hat{A}$ is the initiator and party $\hat{B}$ is the responder.

Messages Exchange

The party $\hat{A}$ chooses $x \leftarrow_{\$} \mathbb{Z}_q$ and compute X as following $(X_1, X_2) \leftarrow (g_1^x, g_2^x)$, a random seed $s_{\hat{A}} \leftarrow_{\$} \{0,1\}^d$ and then computes

$$K_{\hat{A}} = \hat{\Gamma}(B_i^x)_{i\in[n]}, \omega_{\hat{A}}), \Phi_{\hat{A}} \leftarrow \mathsf{Ext}(k_{\hat{A}}, s_{\hat{A}})$$

Then sends $(\hat{B}, \hat{A}, X, \Phi_{\hat{A}}, s_{\hat{A}})$ to party $\hat{B}$. Simulatly, the part $\hat{B}$ will follow the same steps and sends $(\hat{A}, \hat{B}, Y, \Phi_{\hat{B}}, s_{\hat{B}})$ to party $\hat{A}$.

Upon receiving $(\hat{B}, \hat{A}, X, \Phi_{\hat{A}}, s_{\hat{A}})$, party $\hat{B}$ uses his secret key to get $k'_{\hat{A}} = \Lambda_{sk}(X) = \hat{\Gamma}(X_1^{b_{i,1}} X_2^{b_{i,2}})_{i\in[n]}$, then computes $\Phi'_{\hat{A}} \leftarrow \mathsf{Ext}(k'_{\hat{A}}, s_{\hat{A}})$. The party $\hat{B}$ checks $\Phi_{\hat{A}} = \Phi'_{\hat{A}}$, if not then halt. Otherwise, the party $\hat{B}$ computes the session key.

Session Key

The party $\hat{B}$ compute his session key as following:

1) $K \leftarrow k'_{\hat{A}} \oplus k_{\hat{B}}$.
2) $seed \leftarrow \hat{A} \parallel \hat{B} \parallel pk_{\hat{A}} \parallel pk_{\hat{B}} \parallel C_{\hat{A}} \parallel C_{\hat{B}} \parallel k'_{\hat{A}} \parallel k_{\hat{B}}$.
3) $k_s \leftarrow \mathsf{PRF}(K, seed)$.

The correctness of the above protocol follows from the correctness of the underlying hash proof system.

Theorem 3. *If the DDH assumptions hold in groups G and $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a $(\nu, \epsilon_{\mathsf{Ext}})$-Strong*

Table 1: Protocols comparison

Protocol	Computation	Model	Security	Assumption	NAXOS Approch	SPK/EPK
Okamoto [22]	8E	eCK	Standard	DDH & πPRF	Yes	6/6
Moriyama *et al.* [21]	18E	eCK	Standard	DDH & πPRF	No	6/3
Fujioka *et al.* [9]	9E	eCK	Standard	DDH & KEM	No	1/2
HMQV [17]	2.5E	CK, wPFS, KCI, LEP	RO	GDH, KEA1	Yes	1/1
NAXOS [18]	4E	eCK	RO	GDH	Yes	1/1
Mojahed *et al.* [20]	3E	eCK	RO	DLIN	No	1/1
Our	4E	eCK	Standard	DDH,Λ_{sk}	No	2/1

Extractor, $\mathsf{PRF} : \mathcal{K}_{\mathsf{PRF}} \times \mathcal{D}_{\mathsf{PRF}} \rightarrow \mathcal{R}_{\mathsf{PRF}}$ *is a* $(q, t, \epsilon_{\mathsf{PRF}})$*-secure pseudo-random function family. Then the proposed protocol is eCK-secure in the sense of Deffinition 8.*

Proof. From the proof of Theorem 2 and according to proof of Theorem 1, we concludes the proof of Theorem 3. □

5.3 Efficiency

We show the efficiency of our protocol compare to other related ones regarding based assumption, computational efficiency and security model will be discussed in this section. In Table 1, we show number of exponentiation in G (E), number of static public keys (SPK) and number of ephemeral public key (EPK).

From Table 1, we show that our paradigm is much efficient group exponentiations count comparing to a similar protocol that does not rely on NAXOS trick or proved in the standard model. Our protocol does not rely on πPRF or KEM; it uses an adaptive smooth projective hash function instead. Since our protocol is using standard assumption and preliminaries, thus, it is practical to design it using different language programs and various devices.

6 Conclusions

In this paper, we presented a general construction for AKE-secure protocol from the projective hash family secures under hard subset membership problem in the standard model. We gave a novel security proof from games sequences tool introduced by [24]. Our methodology in research was how to design an eCK-secure paradigm from a smooth projective hash function defined in [7]. In our study, we gave a literature about using NAXOS trick in developing an AKE-secure protocol and stated open problem related to that. We proved the security of our paradigm in the standard model which presents another challenge in our research.

Moreover, we also gave an instantiation of our protocol from DDH. We show the efficiency of our protocol compares to other similar AKE protocol.

References

[1] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Advances in Cryptology (CRYPTO'93)*, pp. 232–249, Springer, 1993.

[2] S. Blake-Wilson, D. Johnson, and A. Menezes, "Key agreement protocols and their security analysis," in *IMA International Conference on Cryptography and Coding*, LNCS 1355, pp. 30–45, Springer, 1997.

[3] S. Blake-Wilson and A. Menezes, "Entity authentication and authenticated key transport protocols employing asymmetric techniques," in *Security Protocols*, pp. 137–158, Springer, 1997.

[4] E. Bresson, O. Chevassut, and D. Pointcheval, "Dynamic group diffie-hellman key exchange under standard assumptions," in *Advances in Cryptology (EUROCRYPT'02)*, pp. 321–336, Springer, 2002.

[5] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Advances in Cryptology (EUROCRYPT'01)*, pp. 453–474, Springer, 2001.

[6] O. Chevassut, P.-A. Fouque, P. Gaudry, and D. Pointcheval, "Key derivation and randomness extraction," *IACR Cryptology ePrint Archive*, vol. 2005, pp. 61, 2005.

[7] R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption," in *Advances in Cryptology (EUROCRYPT'02)*, pp. 45–64, Springer, 2002.

[8] C. J. Cremers, "Session-state reveal is stronger than ephemeral key reveal: Attacking the naxos authenticated key exchange protocol," in *Applied Cryptography and Network Security*, pp. 20–33, Springer, 2009.

[9] A. Fujioka, K. Suzuki, K. Xagawa, and K. Yoneyama, "Strongly secure authenticated key exchange from factoring, codes, and lattices," *Designs, Codes and Cryptography*, vol. 76, no. 3, pp. 469–504, 2015.

[10] R. Gennaro and Y. Lindell, "A framework for password-based authenticated key exchange," in *Advances in Cryptology (EUROCRYPT'03)*, pp. 524–543, Springer, 2003.

[11] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions," *Journal of the ACM*, vol. 33, no. 4, pp. 792–807, 1986.

[12] H. Huang, "Authenticated key exchange protocol under computational diffie–hellman assumption from trapdoor test technique," *International Journal of Communication Systems*, vol. 28, no. 2, pp. 325–343, 2015.
[13] J. Katz and V. Vaikuntanathan, "Round-optimal password-based authenticated key exchange," *Journal of Cryptology*, vol. 26, no. 4, pp. 714–743, 2013.
[14] E. Kiltz, K. Pietrzak, M. Stam, and M. Yung, "A new randomness extraction paradigm for hybrid encryption," in *Advances in Cryptology (EUROCRYPT'09)*, pp. 590–609, Springer, 2009.
[15] M. Kim, A. Fujioka, and B. Ustaoğlu, "Strongly secure authenticated key exchange without naxosapproach," in *Advances in Information and Computer Security*, pp. 174–191, Springer, 2009.
[16] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology (CRYPTO'99)*, pp. 388–397, Springer, 1999.
[17] H. Krawczyk, "HMQV: A high-performance secure diffie-hellman protocol," in *Advances in Cryptology (CRYPTO'05)*, pp. 546–566, Springer, 2005.
[18] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *Provable Security*, pp. 1–16, Springer, 2007.
[19] K. Minkyu, A. Fujioka, B. USTAO *et al.*, "Strongly secure authenticated key exchange without naxos'approach under computational diffie-hellman assumption," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 95, no. 1, pp. 29–39, 2012.
[20] M. Mohamed, X. Wang, and X. Zhang, "Efficient secure authenticated key exchange without naxos approach based on decision linear problem," in *Collaborative Computing: Networking, Applications, and Worksharing*, pp. 243–256, Springer, 2015.
[21] D. Moriyama and T. Okamoto, "An eCK-secure authenticated key exchange protocol without random oracles," in *International Conference on Provable Security*, LNCS 5848, pp 154–167, Springer, 2009.
[22] T. Okamoto, "Authenticated key exchange and key encapsulation in the standard model," in *Advances in Cryptology (ASIACRYPT'07)*, pp. 474–484, Springer, 2007.
[23] B. Qin and S. Liu, "Leakage-resilient chosen-ciphertext secure public-key encryption from hash proof system and one-time lossy filter," in *Advances in Cryptology (ASIACRYPT'13)*, pp. 381–400, Springer, 2013.
[24] V. Shoup, "Sequences of games: A tool for taming complexity in security proofs." *IACR Cryptology ePrint Archive*, vol. 2004, pp. 332, 2004.
[25] V. Shoup, *On Formal Models for Secure Key Exchange*, Citeseer, 1999.
[26] Z. Yang, "Efficient eck-secure authenticated key exchange protocols in the standard model," in *Information and Communications Security*, pp. 185–193, Springer, 2013.

A Proof of Theorem 1

Let $\mathcal{M}$ be a polynomial bounded adversary against protocol Π.Let sid^* is the target session chosen by adversary $\mathcal{M}$. Without lose fo generality, assume $\hat{A}$ is the owner of the session sid^* and $\hat{B}$ is the peer. Let sid be $(\hat{A}, \hat{B}, \tilde{C}_{\hat{A}}, \tilde{C}_{\hat{A}}, I)$ where $\tilde{sk}_{\hat{A}} \leftarrow_{\$} \mathcal{SK}, \tilde{pk}_{\hat{A}} \leftarrow \mu(\tilde{sk}_{\hat{A}})$ is the public keys for $(\hat{A})$ and $\tilde{sk}_{\hat{B}} \leftarrow_{\$} \mathcal{SK}; \tilde{pk}_{\hat{B}} \leftarrow \mu(\tilde{sk}_{\hat{B}})$ is the public keys for $(\hat{B})$. Assume also that $\mathsf{Adv}^{\mathrm{ake}}_{\mathcal{M},\Pi}(k)$ is the adversary advantage which we want to evaluate in this proof. From fresh session definition we will have this two events:

Case 1: Existence of a matching session $\overline{sid^*}$ for the target session sid^* which will give us following sub-events:

- Case 1.1 : $\neg StateReveal(sid^*) \vee \neg StateReveal(\overline{sid^*})$.
- Case 1.2 : $\neg StaticKeyReveal(owner) \vee \neg StaticKeyReveal(peer)$.
- Case 1.3 : $\neg StateReveal(sid^*) \vee \neg StaticKeyReveal(peer)$.
- Case 1.4 : $\neg StaticKeyReveal(owner) \vee \neg StateReveal(\overline{sid^*})$.

Case 2: No existence of a matching session for the target session sid^* which will generate the following sub-events:

- Case 2.1 : $\neg StateReveal(sid^*)$.
- Case 2.2 : $\neg StaticKeyReveal(owner)$.

Obviously, those sub-events are independent events. We can describe Case 1.2 as Case 2.2, similarly, we can describe cases (Case 1.3,Case 1.4) as Case 2.1. We do that because: the existance adversary can breaks the protocol in cases (1.2, Case 1.3, Case 1.4) will let us to construct another adversary can breaks it in cases (Case 2.2, Case 2.1). Thus, we can construct three adversaries will break the protocol in the previous sub-events.

Case 1.1

To analyze this event, Adversary $\mathcal{M}$ will play next games:

- Game_{1-0}: This is eCK original game where adversary $\mathcal{M}$ try to distinguish the real session key from random string.

  Claim 1. *let* G0 *be the event that* $b = b'$ *in* Game_{1-0}. *we claim that*

  $$\Pr[\mathsf{G0}] = \frac{\mathsf{Adv}^{\mathrm{ake}}_{\mathcal{M},\Pi}(\lambda) + 1}{2} \tag{2}$$

  Proof. it's easy to derive the proof from Definition 8. □

- Game_{1-1}: This is reduced game from Game_{1-0}, In this game the adversary will choose only two parties $\hat{A}, \hat{B}$ and only two sessions, the target session and its matching session$(sid^*, \overline{sid^*})$ with identifiers $(\hat{A}, \hat{B}, \tilde{C}_{\hat{A}}, \tilde{C}_{\hat{B}}, \mathcal{I})$ and $(\hat{B}, \hat{A}, \tilde{C}_{\hat{B}}, \tilde{C}_{\hat{A}}, \mathcal{I})$ respectively. We suppose that $\mathcal{M}$ activates at most $s(\lambda)$ sessions for each $n(\lambda)$ party For game state, see Appendix **??**.

Claim 2. *let* $\overline{\mathsf{G1}}$ *be the event that* $\mathcal{A}$*success in guessing* sid^* *and* $\overline{sid^*}$ *in* Game_{1-1}. *we claim that*

$$\Pr[\mathsf{G0}] - \Pr[\mathsf{G1}] \leq \frac{2}{n(\lambda)^2 s(\lambda)} \tag{3}$$

Proof. In this game it's obvious that this game is similar to game Game_{1-1} except it required the adversary to guess target session and its matching session correctly to win this game. To select correct parties $\hat{A}$ nad $\hat{B}$, adversary should choose between $n(\lambda)$ parties the couple$(\hat{A}, \hat{B})$, Let $\Pr\left[\hat{A} \cap \hat{B}\right]$ denotes that event, thus:

$$\Pr\left[\hat{A} \cap \hat{B}\right] = \frac{1}{C_2^{n(\lambda)}} = \frac{1}{\frac{n(\lambda)!}{(n(\lambda)-2)!2!}}$$
$$= \frac{2}{n(\lambda)(n(\lambda)-1)} \leq \frac{2}{n^2(\lambda)}$$

In other hand, the adversary should success in guessing target session and its matching session.
Let $\Pr\left[sid^*_{\hat{A},\hat{B}} \cup sid^*_{\hat{A},\hat{B}}\right]$ denote the probability that adversary successfully guess the target session and its matching session thus:

$$\Pr\left[sid^*_{\hat{A},\hat{B}} \cup sid^*_{\hat{A},\hat{B}}\right] =$$
$$\Pr\left[sid^*_{\hat{A},\hat{B}}\right] + \Pr\left[sid^*_{\hat{A},\hat{B}}\right] - \Pr\left[sid^*_{\hat{A},\hat{B}} \cap sid^*_{\hat{A},\hat{B}}\right]$$

$$\Pr\left[sid^*_{\hat{A},\hat{B}} \cap sid^*_{\hat{A},\hat{B}}\right] = \frac{1}{P_2^{s(\lambda)}} = \frac{1}{\frac{s(\lambda)!}{(s(\lambda)-2)!}}$$
$$= \frac{1}{s(\lambda)(s(\lambda)-1)}$$

thus

$$\Pr\left[sid^*_{\hat{A},\hat{B}} \cup sid^*_{\hat{A},\hat{B}}\right] = \frac{1}{s(\lambda)} + \frac{1}{s(\lambda)} - \frac{1}{s(\lambda)(s(\lambda)-1)} =$$
$$\frac{s(\lambda)-2}{s(\lambda)(s(\lambda)-1)} \leq \frac{1}{s(\lambda)}$$

From these two probabilities, we can derive the whole probability that adversary success in guessing parties $\hat{A}$ and $\hat{A}$ with target session and its matching session with the form:

$$\Pr[\mathsf{G0}] - \Pr[\mathsf{G1}] \leq \Pr\left[\hat{A} \cap \hat{B}\right] \Pr\left[sid^*_{\hat{A},\hat{B}} \cup sid^*_{\hat{B},\hat{A}}\right]$$
$$= \frac{2}{n(\lambda)^2 s(\lambda)}$$

□

- Game_{1-2}: We transform Game_{1-1} into Game_{1-2}, the way of generation of k will be change.
In this game, *Sim* computes $k_{(.)}$ with HPS:Priv$(sk_{(.)}, C_{(.)})$ instead of HPS:Pub$(pk_{(.)}, C_{(.)}, \omega_{(.)})$.

Claim 3. *let* $\mathsf{G2}$ *be the event that Sim computes* $k_{(.)}$ *with HPS:Priv*$(sk_{(.)}, C_{(.)})$ *instead of HPS:Pub*$(pk_{(.)}, C_{(.)}, \omega_{(.)})$. *we claim that*

$$\Pr[\mathsf{G1}] = \Pr[\mathsf{G2}] \tag{4}$$

Proof. Since HPS is projective, this change is purely conceptual, and thus Pr[S3] = Pr[S2]. □

- Game_{1-3}: We transform Game_{1-2} into Game_{1-3}, the way of generation of $\tilde{C}_{(.)}$ will be change. In this game, *Sim* samples $\tilde{C}_{(.)}$ from $\mathcal{C} \setminus \mathcal{V}$.

Claim 4. *let* $\mathsf{G3}$ *be the event that* $\tilde{C}_{(.)} \leftarrow_{\$} \mathcal{C} \setminus \mathcal{V}$. *we claim that*

$$\Pr[\mathsf{G2}] - \Pr[\mathsf{G3}] \leq \mathsf{Adv}^{\mathrm{smp}}_{HPS,\mathcal{A}}(\lambda) \tag{5}$$

which $\mathsf{Adv}^{\mathrm{smp}}_{HPS,\mathcal{A}}(\lambda)$ *is advantage of some efficient adversary* $\mathcal{A}$ *to beaks HPS.*

Proof. A straightforward reduction to the indistinguishability of the subset membership problem yields we can conclude our proof. □

- Game_{1-4}: We transform Game_{1-3} into Game_{1-4}. This game is the same as Game_{1-3}, except that we choose $\Phi_{(.)}$ randomly instead of computing it from the Ext function.

Claim 5. *let* $\mathsf{G4}$ *be the event of evaluation of* $\Phi_{(.)} \leftarrow_{\$} \{0,1\}^m$ *randomly. we claim that*

$$\Pr[\mathsf{G3}] - \Pr[\mathsf{G4}] \leq 2\epsilon_{\mathsf{Ext}} \tag{6}$$

Proof. It shows clearly that if the adversary can distinguish between Game_{1-3} and Game_{1-4} then he can generate the same value of Ext function. In Game_{1-3}, $\Pr[G3]$ represents $\Pr\left[\mathsf{Ext}(k_{(.)}), s_{(.)}\right]$ where $s_{(.)}$ represent the seed of the extraction function and key generated from HPS. In Game_{1-4}, $\Pr[G4]$ represents $r \leftarrow_{\$} \{0,1\}^m$. From (k,$\epsilon$)-Strong Extractor definition we get

$$\frac{1}{2} \sum_{y \in (m,k)} \left| Pr[Ext(s,x) = \Phi_{(.)}] - Pr[r = \Phi_{(.)}] \right| = \epsilon$$

We can write the above equation in form

$$\frac{1}{2}\sum_{y\in(m,k)}|\Pr[G3]-\Pr[G4]|=\epsilon$$

which imply 6. □

- Game_{1-5}: We transform Game_{1-4} into Game_{1-5}. This game is the same as Game_{1-4}, except that we choose k_s randomly instead of computing it from the PRF function.

Claim 6. *let* G5 *be the event of computing* $k_s \leftarrow_{\$} \mathcal{R}_{\mathsf{PRF}}$ *randomly. we claim that*

$$\Pr[\mathsf{G4}]-\Pr[\mathsf{G5}]\leq\epsilon_{\mathsf{PRF}}+\frac{1}{2} \tag{7}$$

Proof. It shows clearly that if the adversary can distinguish between Game_{1-4} and Game_{1-5} then he can generate the same value of PRF function. $\Pr\left[EXP^{ind-cma}_{\mathsf{PRF},\mathcal{A}}(\lambda)=1\right]$ represents $\Pr[\mathsf{G4}]-\Pr[\mathsf{G5}]$. From Pseudo-Random function definition we get

$$\left|\Pr\left[EXP^{ind-cma}_{\mathsf{PRF},\mathcal{A}}(\lambda)=1\right]-\frac{1}{2}\right|\leq\epsilon_{\mathsf{PRF}}$$

We can write above equation in form

$$|\Pr[\mathsf{G4}]-\Pr[\mathsf{G5}]|\leq\epsilon_{\mathsf{PRF}}+\frac{1}{2}$$

which complete the proof.
Apparently, Pseudo-Random function behave as one time pad in game Game_{1-5}, which imply

$$\Pr[\mathsf{G5}]=\frac{1}{2} \tag{8}$$

□

Combining Equations (3), (4), (5), (6), (7) and (8), we obtain:

$$\mathsf{Adv}^{\mathrm{ake}}_{\mathcal{M},\Pi}(\lambda)\leq\frac{4}{n^2(\lambda)s(\lambda)}+2\mathsf{Adv}^{\mathrm{smp}}_{\mathcal{A},HPS}(\lambda)+4\epsilon_{\mathsf{Ext}}+2\epsilon_{\mathsf{PRF}}+1 \tag{9}$$

From the sequence of preceding claims, and since those following probabilities $\mathsf{Adv}^{\mathrm{smp}}_{\mathcal{A},HPS}(\lambda), \epsilon_{\mathsf{Ext}}$ and ϵ_{PRF} are negligible in λ, then we conclude that $\mathsf{Adv}^{\mathrm{ake}}_{\mathcal{M},\Pi}(\lambda)$ is negligible in λ. Thus our protocol is secure.

Case 2

To analyze this event, Adversary $\mathcal{M}$will play next games:

- Game_{2-0}: This is eCK original game where adversary $\mathcal{M}$ try to distinguish the real session key from random string.

Claim 7. *let* G20 *be the event that* $b=b'$ *in* Game_{2-0}. *we claim that*

$$\Pr[\mathsf{G20}]=\frac{\mathsf{Adv}^{\mathrm{ake}}_{\mathcal{M},\Pi}(\lambda)+1}{2}. \tag{10}$$

Proof. That proof can be derived from Game_{1-0}. □

- Game_{2-1}: This is reduced game from Game_{2-0}, In this game the adversary will choose only two parties $\hat{A},\hat{B}$ and only target session (sid^*) with identifier $(\hat{A},\hat{A},\tilde{C}_{\hat{A}},\tilde{C}_{\hat{B}},\mathcal{I})$.

Claim 8. *let* G21 *be the event that* $\hat{A}$ *success in guessing* sid^* *in* Game_{2-1}. *we claim that*

$$\Pr[\mathsf{G20}]-\Pr[\mathsf{G21}]\leq\frac{2}{n^2(\lambda)s(\lambda)} \tag{11}$$

Proof. In this game, it's obvious that this game is similar to game Game_{2-1} except it's required adversary to guess target session correctly to win this game. To select correct parties $\mathcal{A}$nad $\mathcal{B}$, adversary should choose between $n(k)$ parties the couple$(\hat{A},\hat{B})$, Let $\Pr\left[\hat{A}\cap\hat{B})\right]$ denotes that event, thus:

$$\Pr\left[\hat{A}\cap\hat{B}\right]=\frac{1}{C_2^{n(\lambda)}}=\frac{1}{\frac{n(\lambda)!}{(n(\lambda)-2)!2!}}=\frac{2}{n(\lambda)(n(\lambda)-1)}\leq\frac{2}{n^2(\lambda)}$$

where C_b^a is the combination
In other hand, the adversary should success in guessing target session and its matching session. Let $\Pr\left[sid^*_{\hat{A},\hat{B}}\right]$ denote the probability that adversary successfully guess the target session from $s(\lambda)$ sessions, thus:

$$\Pr\left[sid^*_{\hat{A},\hat{B}}\right]=\frac{1}{s(\lambda)}$$

From these two probability we can derive the whole probability that adversary success in guessing parties $\hat{A}$ and $\hat{B}$ with target session and its matching session with the form:

$$\Pr[\mathsf{G20}]-\Pr[\mathsf{G21}]\leq\Pr\left[\hat{A}\cap\hat{B}\right]\Pr\left[sid^*_{\hat{A},\hat{B}}\cup sid^*_{\hat{B},\hat{A}}\right]=\frac{2}{n(\lambda)^2s(\lambda)}$$

□

- Game_{2-2}: This game is behave similiary to game Game_{1-2}. Thus, we concludes

$$\Pr[\mathsf{G1}]=\Pr[\mathsf{G2}] \tag{12}$$

- Game_{2-3}: We transform Game_{2-2} into Game_{2-3}, the way of generation of $\tilde{C}_{(.)}$ will be change. In this game, Sim samples $\tilde{C}_{(.)}$ from $\mathcal{C}\setminus\mathcal{V}$.

Claim 9. *let* G23 *be the event that* $\tilde{C}_{(.)} \leftarrow_{\$} \mathcal{C} \setminus \mathcal{V}$. *we claim that*

$$\Pr[\mathsf{G22}] - \Pr[\mathsf{G23}] \leq \frac{q^2_{HPS.Priv}}{2}.\mathsf{Adv}^{\mathrm{smp}}_{HPS,\mathcal{A}}(\lambda) \quad (13)$$

which $\mathsf{Adv}^{\mathrm{smp}}_{HPS,\mathcal{A}}(\lambda)$ *is advantage of some efficient adversary* $\mathcal{A}$ *to beaks HPS, and* $q_{HPS.Priv}$ *is the number of queried made by* $\mathcal{A}$*on HPS.Priv.*

Proof. In this game we transformed from Game_{2-2} by changing $\tilde{C}_{(.)}$ with $\tilde{C}_{(.)} \leftarrow_{\$} \mathcal{C} \setminus \mathcal{V}$. Without losing of generality, The adversary will make $q_{HPS.Priv}$ queries to oracle without repeat of the same query. We can get the probability of halt as:

$$\Pr[\perp] = C_2^{q_{HPS.Priv}} = \frac{q_{HPS.Priv}!}{(q_{HPS.Priv}-2)!2!} = \frac{q_{HPS.Priv}(q_{HPS.Priv}-1)}{2} \leq \frac{q^2_{HPS.Priv}}{2}$$

The difference between $\Pr[\mathsf{G22}]$ and $\Pr[\mathsf{G23}]$ can be parlayed into a corresponding $\mathsf{Adv}^{\mathrm{smp}}_{HPS,\mathcal{A}}(\lambda)$. And that can be conclude clearly from the indistinguishability of the subset membership problem yields we can conclude. □

- Game_{2-4}: We transform Game_{2-3} into Game_{2-4}. This game is the same as Game_{2-3}, except that we choose $\Phi_{(.)}$ randomly instead of computing it from the Ext function.

Claim 10. *let* G24 *be the event of evaluation of* $\Phi_{(.)} \leftarrow_{\$} \{0,1\}^m$ *randomly,* ϵ_{Ext} *be the advantage that* $\mathcal{A}_{\mathsf{Ext}}$ *can breaks* Ext *security and* q_{Ext} *the number of queries made by* $\mathcal{A}_{\mathsf{Ext}}$. *We claim that*

$$\Pr[\mathsf{G23}] - \Pr[\mathsf{G24}] \leq q^2_{\mathsf{Ext}}.\epsilon_{\mathsf{Ext}} \quad (14)$$

Proof. It shows clearly that if the adversary can distinguish between Game_{2-3} and Game_{2-4} then he can generate the same value of Ext function. In Game_{2-3}, $\Pr[G23]$ represents $\Pr\left[\mathsf{Ext}(k_{(.)}), s_{(.)}\right]$ where $s_{(.)}$ represent the seed of the extraction function and key generated from HPS. In Game_{2-4}, $\Pr[G24]$ represents $r \leftarrow_{\$} \{0,1\}^m$. From (k,$\epsilon$)-Strong Extractor definition we get

$$\frac{1}{2}\sum_{y\in(m,k)} \left|Pr[Ext(s,x) = \Phi_{(.)}] - Pr[r = \Phi_{(.)}]\right| = \epsilon$$

We can write the above equation in form

$$\frac{1}{2}\sum_{y\in(m,k)} |\Pr[G3] - \Pr[G4]| = \epsilon$$

Without losing of generality, We let the adversary to make q_{Ext} queries to oracle without repeat of the same query. We can get the probability of halt as:

$$\Pr[\perp] = C_2^{q_{\mathsf{Ext}}} = \frac{q_{\mathsf{Ext}}!}{(q_{\mathsf{Ext}}-2)!2!} = \frac{q_{\mathsf{Ext}}(q_{\mathsf{Ext}}-1)}{2} \leq \frac{q^2_{\mathsf{Ext}}}{2}$$

combining above equations we conclude our proof. □

- Game_{2-5}: We transform Game_{2-4} into Game_{2-5}. This game is the same as Game_{2-4}, except that we choose k_s randomly instead of computing it from the PRF function.

Claim 11. *let* G25 *be the event of computing* $k_s \leftarrow_{\$} \mathcal{R}_{\mathsf{PRF}}$ *randomly,* ϵ_{PRF} *be the advantage that* $\mathcal{A}_{\mathsf{PRF}}$ *can breaks* PRF *security and* q_{PRF} *the number of queries made by* $\mathcal{A}_{\mathsf{PRF}}$. *We claim that*

$$\Pr[\mathsf{G24}] - \Pr[\mathsf{G25}] \leq \frac{q^2_{\mathsf{PRF}}.\epsilon_{\mathsf{Ext}} + 1}{2} \quad (15)$$

Proof. It shows clearly that if the adversary can distinguish between Game_{2-4} and Game_{2-5} then he can generate the same value of PRF function.
$\Pr\left[EXP^{ind-cma}_{\mathsf{PRF},\mathcal{A}_{\mathsf{PRF}}}(\lambda) = 1\right]$ represents $\Pr[\mathsf{G24}] - \Pr[\mathsf{G25}]$. From Pseudo-Random function definition we get

$$\left|\Pr\left[EXP^{ind-cma}_{\mathsf{PRF},\mathcal{A}_{\mathsf{PRF}}}(\lambda) = 1\right] - \frac{1}{2}\right| \leq \epsilon_{\mathsf{PRF}}$$

We can write above equation in form

$$|\Pr[\mathsf{G4}] - \Pr[\mathsf{G5}]| \leq \epsilon_{\mathsf{PRF}} + \frac{1}{2}$$

Without losing of generality, We let the adversary to make q_{PRF} queries to oracle without repeat of the same query. We can get the probability of halt as:

$$\Pr[\perp] = C_2^{q_{\mathsf{PRF}}} = \frac{q_{\mathsf{PRF}}!}{(q_{\mathsf{PRF}}-2)!2!} = \frac{q_{\mathsf{PRF}}(q_{\mathsf{PRF}}-1)}{2} \leq \frac{q^2_{\mathsf{PRF}}}{2}$$

which complete the proof.
Apparently, Pseudo-Random function behave as one time pad in game Game_{1-5}, which imply

$$\Pr[\mathsf{G5}] = \frac{1}{2} \quad (16)$$

□

Combining 11,12,13,14,15 and 16 we obtain:

$$\mathsf{Adv}^{\mathrm{ake}}_{\mathcal{M},\Pi}(\lambda) \leq \frac{4}{n^2(\lambda)s(\lambda)} + q^2_{HPS}\mathsf{Adv}^{\mathrm{smp}}_{\mathcal{A},HPS}(\lambda) + 2q^2_{\mathsf{Ext}}\epsilon_{\mathsf{Ext}} + q^2_{\mathsf{PRF}}\epsilon_{\mathsf{PRF}} + 1 \quad (17)$$

From the sequence of preceding claims, and since those following probabilities $\mathsf{Adv}^{\mathrm{smp}}_{\mathcal{A},HPS}(\lambda), \epsilon_{\mathsf{Ext}}$ and ϵ_{PRF} are negligible in λ, then we conclude that $\mathsf{Adv}^{\mathrm{ake}}_{\mathcal{M},\Pi}(\lambda)$ is negligible in λ. Thus our protocol is secure.

Appendix: Proof of Theorem 2

To prove this theorem we retrieve the definition of the universal projective hash function defined by Cramer and Shop [7]. To follow the previous definition of HPS, we should define following:

- Existence of a subset membership problem M.
- Existence of a ϵ-universal hash projective function.

Let M be a subset membership problem, we write $\Lambda[\mathcal{C}, \mathcal{V}, \mathcal{W}, \mathcal{R}]$ to indicate the instance Λ where $\mathcal{C}, \mathcal{V} = G^2, \mathcal{V} \subset \mathcal{C}, \mathcal{W} = \mathbb{Z}_q, \mathcal{R} = \mathbb{Z}_q^n$. For $(g_1^r, g_2^r) \in \mathcal{C}$ with witness $r \in \mathcal{W}$. We define two sequences of random variables as follows $C \leftarrow \mathcal{V}, C' \leftarrow \mathcal{C} \setminus \mathcal{V}$ at random.

Claim 12. *We say M is a hard subset membership problem if (Λ, C) and (Λ, C') are computationally indistinguishable.*

Proof. Let $C_0 = (u_1, u_2) = (g_1^r, g_2^r) \in \mathcal{V}$ where $r \in \mathbb{Z}_q$ is a valid witness. Let $C_1 = (u_1^*, u_2^*) \in \mathcal{C} \setminus \mathcal{V}$ where $u_1^*, u_2^* \leftarrow_{\$} G$. Retrieve the advantage of adversary in Section 2.3 we derive:

$$\mathsf{Adv}^{\mathrm{smp}}_{HPS,\mathcal{A}}(k) = |\Pr[\mathcal{A}(\mathcal{C}, \mathcal{V}, C_0) = 1 | C_0 \leftarrow_{\$} \mathcal{V}] - \Pr[\mathcal{A}(\mathcal{C}, \mathcal{V}, C_1) = 1 | C_1 \leftarrow_{\$} \mathcal{C} \setminus \mathcal{V}]|$$

Obviously, to distinguish between C_0 and C_1 is to solve the logarithm problem which is hard to solve by assumption. Thus, we say the subset membership is hard. □

Let $\Lambda_{sk} : \mathcal{C} \to \mathcal{K}$ be a projective hash function indexed with $sk \in \mathcal{SK}$ instantiated with $\Lambda_{sk}(C) = \hat{\Gamma}((u_1^{x_{i,1}} u_2^{x_{i,2}})_{i\in[n]})$.

Claim 13. *We say Λ_{sk} is an ϵ-universal projective hash function.*

Proof. To show the ϵ-universality we show that, for any fixed $C = (u_1, u_2) \in \mathcal{C} \setminus \mathcal{V}$ in the distripution of $(C, \Lambda_{sk}(C))$ is that of two random and independent group elements. Consider the map

$$f((x_{i,1}, x_{i,2})_{i\in[n]}) = (pk, C) = (g_1^{x_{i,1}} g_2^{x_{i,2}}, u_1^{x_{i,1}} u_2^{x_{i,2}})_{i\in[n]}$$

mapping trapdoor $sk = (x_{i,1}, x_{i,2})_{i\in[n]}$ to (pk, C) pairs. If we show that this map is injective then we are done, since we are applying this map to a random input and hence will get a random output. We will show an equivalent statement that the map $f'((x_{i,1}, x_{i,2})_{i\in[n]}) = \log_{g_1}(f((x_{i,1}, x_{i,2})_{i\in[n]})) = (\log g_1(pk), \log_{g_1}(C))$ is injective. Let

$$u_1 = g_1^{r_1}, u_2 = g_2^{r_2} = g_1^{\beta r_2}$$

for some $r_1 \neq r_2$ and $\beta = \log_{g_1}(g_2)$. We write

$$pk = (g_1^{x_{i,1}} g_2^{x_{i,2}})_{i\in[n]} = (g_1^{x_{i,1}+\beta x_{i,2}})_{i\in[n]}$$

$$C = (u_1^{x_{i,1}} u_2^{x_{i,2}})_{i\in[n]} = (g_1^{r_1 x_{i,1} + \beta r_2 x_{i,2}})_{i\in[n]}$$

so $(pk, C) = (g_1^{z_{i,1}}, g_1^{z_{i,2}})$ for

$$\begin{pmatrix} z_{1,1} \\ z_{2,1} \\ \vdots \\ z_{n,1} \\ z_{1,2} \\ \vdots \\ z_{n,2} \end{pmatrix} = \rho \begin{pmatrix} x_{1,1} \\ x_{1,2} \\ x_{2,1} \\ x_{2,2} \\ \vdots \\ x_{n,1} \\ x_{n,2} \end{pmatrix}$$

where

$$\rho = \begin{pmatrix} 1 & \beta & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \beta & \cdots & 0 & 0 \\ \vdots & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & 1 & \beta \\ r_1 & r_2\beta & 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & r_1 & r_2\beta \end{pmatrix}$$

We can write it in the form $f'((x_{i,1}, x_{i,2})_{i\in[n]}) = \rho((x_{i,1}, x_{i,2})_{i\in[n]})^T$. Since $det(\rho) = \beta^n(r_2 - r_1)^n \neq 0$ the ρ is not singular, which shows that f' is an injective map and concluds the proof. □

Biography

Mojahed Ismail Mohamed currently, a PhD student of the School of Computer Science and Engineering at University of Electronic Science and Technology of China (UESTC), and have been with UESTC since Jan. 2011. He received his Master degree (2013) in Information Security from UESTC, Chengdu, China, and Bachelor degree (2005) in computer Engineering from Karary University. He did his researches in key exchange protocols (KE).

Xiaofen Wang currently, an associate professor of the School of Computer Science and Engineering at University of Electronic Science and Technology of China (UESTC), and have been with UESTC since Jan. 2010. She received her Ph.D. degree (2009) and Master degree (2006) in cryptography from Xidian University, Xi'an, China, under the supervision of Prof. Guozhen Xiao, and Bachelor degree (2003) in computer science from University of Electronic Science and Technology of China. She was a visiting research fellow at University of Wollongong, working with Prof. Yi Mu from Aug. 2014 to Aug. 2015. Her major research interests include Cryptography, Information Security, Network Security, and Data Security, etc.

Xiaosong Zhang He is the director of the big data research center and big data research institute of University of Electronic Science and Technology (UESTC). For his Bachelor, he graduated from Shanghai Jiaotong University, Master, Ph.D. from the University of Electronic

Science and Technology of China. He is a long-term commitment to network and information security, computer application technology research, focusing on cyberspace security, IT infrastructure, big data security and applications, embedded platform security, network attack detection and software vulnerability and other key areas and Support technology To carry out innovative research and technological research.