Tightly-Secure Authenticated Key Exchange without NAXOS' approach based on Decision Linear Problem

Mojahed Mohamed[1 2], Xiaofen Wang[1], and Xiaosong Zhang[1]

[1] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
[2] Department of Electronic Engineering, Karary University, Omdurman, Sudan
mmmmoj@hotmail.com, wangxuedou@sina.com, johnsonzxs@uestc.edu.cn

Abstract. Design secure Authenticated Key Exchange (AKE) protocol without NAXOS approach is remaining as an open problem. NAXOS approach [4] is used to hide the secret ephemeral key from an adversary even if the adversary in somehow may obtain the ephemeral secret key. Using NAXOS approach will cause two main drawbacks, (i) leaking of the static secret key which will be used in computing the exponent of the ephemeral public key. (ii) maximize of using random oracle when applying to the exponent of the ephemeral public key and session key derivation. In this paper, we present another AKE-secure without NAXOS approach based on decision linear assumption in the random oracle model. We fasten our security using games sequences tool which gives tight security for our protocol.

Keywords: AKE, eCK model, NAXOS' approach, Decision Linear assumption

1 Introduction

An Authenticated Key Exchange protocol (AKE) allows two parties to end up with a shared secret key in secure and authenticated manner. 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.
In 1993, Bellare and Rogaway [1] provided the first formal treatment of entity authentication and authenticated key distribution appropriate to the distributed environment. In 1998, Bellare,Canetti, Mihir and Krawczyk [2] provided a model for studying session-oriented security protocols. They also introduce the "authenticator" techniques that allow for greatly simplifying the analysis of protocols. In addition, they proposed a definition of security of KE protocols rooted in the simulatability approach used to define the security of multiparty computation. In 2002 Canetti and Krawczyk [3] presented their security model which had extended by LaMacchia, Lauter, and Mityagin [4] model and proposed NAXOS

protocol which is secure under their model. That 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. This model doesn't give an adversary capability to trivially break an AKE protocol.
To acquire eCK security, NAXOS needs that the ephemeral public key X be computed from an exponent result from hashing an ephemeral private key x and the static private key a, more precisely $X = g^{H(x,a)}$ instead of $X = g^x$. In this paper generating ephemeral public key as $X = g^{H(x,a)}$ is called NAXOS's approach. In NAXOS's approach no one is capable of querying the discrete logarithm of an ephemeral public key X without the pair (x, a); thus, the discrete logarithm of X is hidden via an additional random oracle. Using NAXOS' approach many protocols [5–8] were claimed secure in the eCK model under the random oracle assumption. In the standard model, eCK-secure protocols were claimed secure in the eCK model as Okamoto [9] ; they use pseudo-random functions instead of hash functions.

Motivating Problem. (1) Design AKE-secure protocol without NAXOS trick to 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 is in contrast to protocols that use the NAXOS' approach. (ii) Minimize the use of the random oracle, by applying it only to the session key derivation. Kim, Minkyu, Atsushi Fujioka, and Berkant Ustaolu [10] proposed two strongly secure authenticated key exchange protocols without NAXOSapproach, 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. [19] designed a protocol without NAXOS approach but secure in RO model, they rely the security of their protocol upon security reduction and we use in this paper the game sequences tools to fasten the security and give tightly secure security proof. (2) Design AKE-secure protocol secure under Decision Linear Assumption. Boneh, Boyen, and Shacham [11] introduced a decisional assumption, called Linear, intended to take the place of DDH in groups - in particular, bilinear groups [12] - where DDH is easy. For this setting, the Linear problem has desirable properties, as Boneh, Boyen and Shacham show: it is hard if DDH is hard, but, at least in generic groups [13], remains hard even if DDH is easy.

Contributions. We present a concrete and practical AKE protocol that is eCK secure under Decisional linear assumption in the random oracle model. Our protocol does not rely on any NAXOS trick that yields a more efficient solution when it is implemented with secure device. We give tight proofs reducing eCK security of our protocol to break the used cryptographic primitives under random oracle.
In our protocol, the ephemeral public key is containing each peers generator, which results in two different discrete logarithm problem with two different generators, which increase hardness for DL's solver.

In the derivation of the session key, each party will compete shared secret from ephemeral keys and static keys. We fasten the security of this protocol using games sequences tool which gives tight security.

Organization. Section 2 reviews security definitions and state the hard problem. Section 3 gives brief for the eCK model. Section 4 proposes AKE-secure protocol with its security results. Section 5 compares our protocol with other related AKE protocols and shows its efficiency. And finally, we draw the conclusion in section 6.

2 Preliminaries

In this section, we review security definitions we will use to construct our protocol.

2.1 The Decision Linear Diffie-Hellman Assumption

Let G be a cyclic group of prime order p and along with arbitrary generators u, v and h where

$$g, u, v, h \in G :< g >= G; u = g^{\alpha}; v = g^{\beta}; g^{\lambda} = h; \alpha, \beta, \lambda \in \mathbb{Z}_p^* \tag{1}$$

consider the following problem:

Decision Linear Problem in G [11] Given $u, v, h, u^a, v^b, h^c \in G$ as input, output yes if $a + b = c$ and no otherwise.
One can easily show that an algorithm for solving Decision Linear in G gives an algorithm for solving DDH in G. The converse is believed to be false. That is, it is believed that Decision Linear is a hard problem even in bilinear groups where DDH is easy. More precisely, we define the advantage of an algorithm $\mathcal{A}$ in deciding the Decision Linear problem in G as

$$\begin{aligned} AdvLinear_{\mathcal{A}} \stackrel{\text{def}}{=} \big|&\Pr\left[\mathcal{A}(u, v, h, u^a, v^b, h^{a+b}) = yes : u, v, h \leftarrow_{\$} G; a, b \leftarrow_{\$} \mathbb{Z}_p\right] \\ &- \Pr\left[\mathcal{A}(u, v, h, u^a, v^b, \gamma) = yes : u, v, \gamma \leftarrow_{\$} G; a, b \leftarrow_{\$} \mathbb{Z}_p\right]\big| \end{aligned} \tag{2}$$

The probability is over the uniform random choice of the parameters to $\mathcal{A}$, and over the coin tosses of $\mathcal{A}$. We say that an algorithm $\mathcal{A}(t, \epsilon)$-decides Decision Linear in G if $\mathcal{A}$ runs in time at most t, and $AdvLinear_{\mathcal{A}}$ is at least ϵ.

Definition 2.1. *We say that the (t, ϵ)-Decision Linear Assumption (DLIN) holds in G if no t-time algorithm has advantage at least ϵ in solving the Decision Linear problem in G.*

2.2 Linear Diffie-Hellman

Let $dl_u, dl_v : G \to \mathbb{Z}_p$ be the discrete logarithm (DL) functions which takes an input $X, Y \in G$ and returns $x, y \to \mathbb{Z}_p$ such that $X = v^x$ and $Y = u^y$. Define the Linear Diffie-Hellman functions $ldh : G^2 \to G$ as $ldh(A, B) = A^{dl_v(X)}B^{dl_u(Y)}$,$ldh(X, Y) = X^{dl_v(A)}Y^{dl_u(B)}$, and Decisional Linear predicate $DLIN_{u,v,h} : G^3 \to \{0, 1\}$ as a function which takes an input $(A, B, Z) \in G^3$ and returns 1 if

$$Z = A^{dl_v(X)}B^{dl_u(Y)} = h^{dl_v(X)+dl_u(Y)} \tag{3}$$

or in input $(X, Y, Z) \in G^3$ and returns 1 if

$$Z = X^{dl_v(A)}Y^{dl_u(B)} = h^{dl_v(X)+dl_u(Y)} \tag{4}$$

3 Security Model

In this section, eCK model is outlined [18]. An n different parties $P = P_1, \cdots, P_n$ running the KE protocol Π in eCK model. Each party possesses long-term static (private/public) keys including the corresponding certificate issued by the certifying authority. The protocol Π is executed between two parties $\mathcal{A}$ and $\mathcal{B}$ whose static public key are A and B respectively. $\mathcal{A}$ and $\mathcal{B}$ will interchange their ephemeral public keys X and Y to obtain the same 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. In order to capture the possible attacks, $\mathcal{M}$ is allowed to make the following queries as well as H queries of (hash) random oracles.

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 newly 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 certifying authority 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 $\mathcal{A}$.

EphemeralKeyReveal(sid): $\mathcal{M}$ obtains the ephemeral private key stored in the session state of 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 session has 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 3.1 (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* ***EphemeralKeyReveal***(sid)*, or*
   (b) Both ***StaticKeyReveal***$(\mathcal{B})$ *and* ***EphemeralKeyReveal***$(\overline{sid})$*.*
3. *If $\overline{sid}$ does not exist, $\mathcal{M}$ makes either of the following queries:*
   (a) Both ***StaticKeyReveal***$(\mathcal{A})$ *and* ***EphemeralKeyReveal***(sid)*, or*
   (b) ***StaticKeyReveal***$(\mathcal{B})$*.*

The eCK security notion can be described now.

Definition 3.2 (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 $\mathcal{M}$):*

$$Adv_{\Pi}^{AKE}(\mathcal{M}) = |2\Pr[b' = b] - 1| \tag{5}$$

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 Protocol

Parameters. Let k be the security parameter and G be a cyclic group with generator g and order a k-bit prime p. Let users public key is a triple of generators $u, v, h \in G$. Parties $\mathcal{A}'s$,$\mathcal{B}'s$ static private key is $a_1, a_2, b_1, b_2 \in \mathbb{Z}_p^*$,respectvly. Where $\mathcal{A}'s$ public key is $A_1 = u^{a_1}, A_2 = u^{a_2}$,$\mathcal{B}'s$ public key is $B_1 = v^{b_1}, B_2 = v^{b_2}$. Let $H : \{0,1\}^* \to \{0,1\}^k$ to be a cryptographic hash function modeled as a random oracle.

4.1 Protocol description

$\mathcal{A}$		$\mathcal{B}$
$a_1, a_2 \leftarrow_\$ \mathbb{Z}_p^* \in G$		$b_1, b_2 \leftarrow_\$ \mathbb{Z}_p^*$
$A_1 = u^{a_1}, A_2 = v^{a_2} \in G$		$B_1 = v^{b_1}, B_2 = u^{b_2} \in G$
$x_1, x_2 \leftarrow_\$ \mathbb{Z}_p^*; X_1 = v^{x_1}; X_2 = u^{x_2}$		$y_1, y_2 \leftarrow_\$ \mathbb{Z}_p^*; Y_1 = u^{y_1}, Y_2 = v^{y_2}$
$sid = (\mathcal{A}, \mathcal{B}, X_1, X_2, -, -, \mathcal{I})$		$sid = (\mathcal{B}, \mathcal{A}, Y_1, Y_2, X_1, X_2, \mathcal{R})$
	$\xrightarrow{(\mathcal{B}, \mathcal{A}, X_1, X_2)}$	
	$\xleftarrow{(\mathcal{A}, \mathcal{B}, Y_1, Y_2)}$	
$Y_1, Y_2 \in^? G$		$X_1, X_2 \in^? G$
$\wedge\, (\mathcal{A}, \mathcal{B}, X_1, X_2, *, *)$		$\wedge\, (\mathcal{B}, \mathcal{A}, Y_1, Y_2, *, *)$
$Z = (B_1 Y_2)^{a_2 + x_1} (B_2 Y_1)^{a_1 + x_2}$		$Z = (A_1 X_2)^{b_2 + y_1} (A_2 X_1)^{b_1 + y_2}$
$sk = H(Z, X_1, X_2, Y_1, Y_2, \mathcal{A}, \mathcal{B})$		$sk = H(Z, X_1, X_2, Y_1, Y_2, \mathcal{A}, \mathcal{B})$

Fig. 1. Our Protocol

As follow description, $\mathcal{A}$ will be the session initiator and $\mathcal{B}$ the session responder.

1. $\mathcal{A}$ chooses randomly an ephemeral private key $x_1, x_2 \in_R \mathbb{Z}_p^*$, computing the ephemeral public key $X_1 = v^{x_1}; X_2 = u^{x_2}$ and sends ($\mathcal{B}$,$\mathcal{A}$,X_1,X_2) to $\mathcal{B}$.
2. Upon receiving ($\mathcal{B}$,$\mathcal{A}$,X_1,X_2), $\mathcal{B}$ verifies that $X_1, X_2 \in G$. if so, $\mathcal{B}$ chooses randomly an ephemeral private key $y_1, y_2 \in_R \mathbb{Z}_p^*$, computing the ephemeral public key $Y_1 = u^{y_1}, Y_2 = v^{y_2}$ and sends ($\mathcal{A}$,$\mathcal{B}$,Y_1,Y_2) to $\mathcal{A}$. Then $\mathcal{B}$ computing the shared secret $Z = (A_1X_2)^{b_2+y_1}(A_2X_1)^{b_1+y_2}$, the session $SK = H(Z, X_1, X_2, Y_1, Y_2, \mathcal{A}, \mathcal{B})$ and competes the session.
3. Upon receiving ($\mathcal{A}$,$\mathcal{B}$,Y_1,Y_2), $\mathcal{A}$ checks if he owns a session with sid ($\mathcal{A}$,$\mathcal{B}$,X_1,X_2,$\times$). if so, $\mathcal{A}$ verifies that $Y_1, Y_2 \in G$. if so, $\mathcal{A}$ computing the shared secret $Z = (B_1Y_2)^{a_2+x_1}(B_2Y_1)^{a_1+x_2}$, the session $SK = H(Z, X_1, X_2, Y_1, Y_2, \mathcal{A}, \mathcal{B})$ and competes the session.

Both parties compute the shared secret

$$\mathcal{B} : Z = (A_1X_2)^{b_2+y_1}(A_2X_1)^{b_1+y_2} = u^{(a_1+x_2)(b_2+y_1)}v^{(a_2+x_1)(b_1+y_2)} \tag{6}$$

$$\mathcal{A} : Z = (B_1Y_2)^{a_2+x_1}(B_2Y_1)^{a_1+x_2} = u^{(b_2+y_1)(a_1+x_2)}v^{(b_1+y_2)(a_2+x_1)} \tag{7}$$

4.2 Protocol Security

Theorem 4.1. *If the DLIN assumption holds in G and H is a random oracle, then the Protocol Π is eCK-secure.*

Let $\mathcal{M}$ be a polynomial bounded adversary against protocol Π, sid^* is the target session chosen by adversary $\mathcal{M}$, $\mathcal{A}$ is the owner of the session sid^* and $\mathcal{B}$ is the peer. Let sid^* be $(\mathcal{A}, \mathcal{B}, X_1^*, X_2^*, Y_1^*, Y_2^*, \mathcal{I})$ where $(A_1, A_2), (B_1, B_2)$ is public keys for $(\mathcal{A}, \mathcal{B})$ respectively,$(a_1^*, a_2^*, b_1^*, b_2^* \leftarrow \mathbb{Z}^*, A_1 \leftarrow u^{a_1^*}, A2 \leftarrow u^{a_2^*}, B_1 \leftarrow v^{b_1^*}, B_2 \leftarrow v^{b_2^*})$. Assume also that $\mathsf{Adv}^{\mathrm{ake}}_{\mathcal{M},\Pi}(k)$ is adversary advantage which we want to evaluate in this proof. We will have this two events:

- case1: Existence of a matching session $\overline{sid^*}$ for the target session sid^*.
- case2: No existence of a matching session for the target session sid^*.

case1. To analyze this event, Adversary $\mathcal{M}$will play next games, Game_{1-0}, Game_{1-1}, Game_{1-2} and Game_{1-3} as follows:

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

  Claim. let G_0 be the event that $b = b'$ in Game_{1-0}. we claim that

  $$\Pr[\mathsf{G}_0] = \frac{\mathsf{Adv}^{\mathrm{ake}}_{\mathcal{M},\Pi}(k) + 1}{2} \tag{8}$$

  Proof. it's easy to derive the proof from definition 3.2

- Game_{1-1}: This is reduced game from Game_{1-0}, In this game the adversary will choose only two parties $\mathcal{A}$, $\mathcal{B}$ and only two sessions, the target session and its matching session$(sid^*, \overline{sid^*})$ with identifiers $(\mathcal{A}, \mathcal{B}, X_1^*, X_2^*, Y_1^*, Y_2^*, \mathcal{I})$ and $(\mathcal{B}, \mathcal{A}, Y_1^*, Y_2^*, X_1^*, X_2^*, \mathcal{R})$ respectively. For game state, see Appendix A.2.

Claim. let $\mathsf{G_1}$ be the event that $\mathcal{A}$success in guessing $sid^*, \overline{sid^*}$ in Game_{1-1}. we claim that

$$\Pr[\mathsf{G0}] - \Pr[\mathsf{G1}] \leq \frac{2}{n(k)^2 s(k)} \tag{9}$$

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

$$\Pr[\mathcal{A} \cap \mathcal{B}] = \frac{1}{C_2^{n(k)}} = \frac{1}{\frac{n(k)!}{(n(k)-2)!2!}} = \frac{2}{n(k)(n(k)-1)} \leq \frac{2}{n^2(k)}$$

In another hand, the adversary should success in guessing target session and its matching session. Let $\Pr[sid_{\mathcal{A},\mathcal{B}} \cup sid_{\mathcal{B},\mathcal{A}}]$ denote the probability that adversary successfully guess the target session and its matching session thus:

$$\Pr[sid_{\mathcal{A},\mathcal{B}} \cup sid_{\mathcal{B},\mathcal{A}}] = \Pr[sid_{\mathcal{A},\mathcal{B}}] + \Pr[sid_{\mathcal{B},\mathcal{A}}] - \Pr[sid_{\mathcal{A},\mathcal{B}} \cap sid_{\mathcal{B},\mathcal{A}}]$$

$$\Pr[sid_{\mathcal{A},\mathcal{B}} \cap sid_{\mathcal{B},\mathcal{A}}] = \frac{1}{P_2^{s(k)}} = \frac{1}{\frac{s(k)!}{(s(k)-2)!}} = \frac{1}{s(k)(s(k)-1)}$$

thus

$$\Pr[sid_{\mathcal{A},\mathcal{B}} \cup sid_{\mathcal{B},\mathcal{A}}] = \frac{1}{s(k)} + \frac{1}{s(k)} - \frac{1}{s(k)(s(k)-1)} = \frac{s(k)-2}{s(s(k)-1)} \leq \frac{1}{s(k)}$$

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

$$\begin{aligned}\Pr[\mathsf{G_0}] - \Pr[\mathsf{G_1}] &\leq \Pr[\mathcal{A} \cap \mathcal{B}] \Pr[sid_{\mathcal{A},\mathcal{B}} \cup sid_{\mathcal{B},\mathcal{A}}] \\ &= \frac{2}{n(k)^2 s(k)}\end{aligned}$$

– Game_{1-2}: We transform Game_{1-1} into Game_{1-2}, computing values $Z^* = (B_1^* Y_2^*)^{a_2^*+x_1^*}(B_2^* Y_1^*)^{a_1^*+x_2^*} = u^{(a_1^*+x_2^*)(b_2+y_1)} v^{(a_2^*+x_1^*)(b_1^*+y_2^*)}$ to random value $Z^* \leftarrow_{\$} G$ where DLIN($B_1^* Y_1^*$,$B_2^* Y_1^*$) = 1. For game state, see Appendix A.3.

Claim. let $\mathsf{G_2}$ be the event that $\mathcal{D}$success in solving DLIN problem in Game_{1-2}. we claim that

$$\Pr[\mathsf{G_1}] - \Pr[\mathsf{G_2}] \leq \mathsf{Adv}_{\mathcal{D}}^{\mathrm{dlin}}(k) \tag{10}$$

Proof. We transform game $\mathsf{Game}_{(1-1)}$ into $\mathsf{Game}_{(1-2)}$ computing values $Z^* = (B_1^* Y_2^*)^{a_2^*+x_1^*}(B_2^* Y_1^*)^{a_1^*+x_2^*} = u^{(a_1^*+x_2^*)(b_2+y_1)} v^{(a_2^*+x_1^*)(b_1^*+y_2^*)}$ to random value $Z^* \leftarrow_{\$} G$ where DLIN($B_1^* Y_1^*$,$B_2^* Y_1^*$) = 1. If adversary success in distinguishing between $\mathsf{Game}_{(1-1)}$ and $\mathsf{Game}_{(1-2)}$ with non-negligible probability, then he can solve the DLIN problem, thus we construct adversary $\mathcal{D}$that solves DLIN problem. In this game, $\mathcal{D}$will choose same parameters in

$\mathsf{Game}_{(1-1)}$ except values $(Z*)$ which will be chosen randomly. There for we obtain:

$$\Pr[\mathsf{G}_1] - \Pr[\mathsf{G}_2] \leq \mathsf{Adv}^{\mathrm{dlin}}_{\mathcal{D}}(k)$$

– Game_{1-3}: We transform Game_{1-2} into Game_{1-3}, computing h by choosing it at random, rather than as a hash function. For game state, see Appendix A.4.

Claim. let G_3 be the event that $\mathcal{H}$ success in distinguishing value H from random string in Game_{1-2}. we claim that

$$\Pr[\mathsf{G}_2] - \Pr[\mathsf{G}_3] \leq \epsilon_{es}(k) \tag{11}$$

which ϵ_{es} is ES-advantage of some efficient algorithm(which is negligible assuming $\mathcal{H}$ is entropy smoothing).

Proof. We will prove here using the same idea in the previous game. In this game we transformed from Game_{1-2} by changing the hash value with a random value. The difference between $\Pr[\mathsf{G}_2]$ and $\Pr[\mathsf{G}_3]$ can be parlayed into a corresponding ES-advantage.
Moreover, as h act as a one-time pad in game Game_{1-3}, it's evident that

$$\Pr[\mathsf{G}_3] = \frac{1}{2} \tag{12}$$

Combining (8),(9),(10),(11) and (12), we obtain

$$\mathsf{Adv}^{\mathrm{dlin}}_{\mathcal{D}}(k) \geq \frac{1}{2}\left[\ \mathsf{Adv}^{\mathrm{ake}}_{\mathcal{M},\Pi}(k) - \frac{4}{n(k)^2 s(k)} - 2\varepsilon_{es}(k)\right] \tag{13}$$

case2. To analyze this event, Adversary $\mathcal{M}$will play next games, Game_{2-0}, Game_{2-1}, Game_{2-2} and Game_{2-3} as follows:

– Game_{2-0}: This is an eCK original game where adversary $\mathcal{M}$ try to distinguish the real session key from a random string. For the game state, see Appendix A.5.

Claim. let G_0 be the event that $b = b'$ in Game_{1-0}. we claim that

$$\Pr[\mathsf{G}_0] = \frac{\mathsf{Adv}^{\mathrm{ake}}_{\mathcal{M},\Pi}(k) + 1}{2} \tag{14}$$

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 $\mathcal{A}$, $\mathcal{B}$ and only target session $(sid^*, \overline{sid^*})$ with identifier $(\mathcal{A}, \mathcal{B}, X_1^*, X_2^*, Y_1^*, Y_2^*, \mathcal{I})$. For game state, see Appendix A.6.

Claim. let G_1 be the event that $\mathcal{A}$success in guessing sid^* in Game_{2-1}. we claim that

$$\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_1] \leq \frac{2}{n(k)^2 s(k)} \tag{15}$$

Proof. In this game, it is obvious that this game is similar to game Game_{2-1} except it's required the 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($\mathcal{A}$, $\mathcal{B}$), Let $\Pr[\mathcal{A} \cap \mathcal{B})]$ denotes that event, thus:

$$\Pr[\mathcal{A} \cap \mathcal{B}] = \frac{1}{C_2^{n(k)}} = \frac{1}{\frac{n(k)!}{(n(k)-2)!2!}} = \frac{2}{n(k)(n(k)-1)} \leq \frac{2}{n^2(k)}$$

In another hand, the adversary should success in guessing target session and its matching session. Let $\Pr[sid_{\mathcal{A},\mathcal{B}}]$ denote the probability that adversary successfully guess the target session from $s(k)$ sessions, thus:

$$\Pr[sid_{\mathcal{A},\mathcal{B}}] = \frac{1}{s(k)}$$

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

$$\begin{aligned}\Pr[\mathsf{G0}] - \Pr[\mathsf{G1}] &\leq \Pr[\mathcal{A} \cap \mathcal{B}] \Pr[sid_{\mathcal{A},\mathcal{B}} \cup sid_{\mathcal{B},\mathcal{A}}] \\ &= \frac{2}{n(k)^2 s(k)}\end{aligned}$$

– Game_{2-2}: We transform Game_{2-1} into Game_{2-2}, computing values $X_1^*, X_2^*, Y_1^*, Y_2^*$ randomly as $X_1^*, X_2^*, Y_1^*, Y_2^* \leftarrow_{\$} G$ which lead to computing value Z^* from random values which make it random value. For the game state, see Appendix A.7.

Claim. let G_2 be the event that $\mathcal{D}$success in solving DLIN problem in $\mathsf{Game}_2 - 2$. we claim that

$$\Pr[\mathsf{G}_1] - \Pr[\mathsf{G2}] \leq \frac{q_{DLIN}^2 . \mathsf{Adv}_{\mathcal{D}}^{\mathrm{dlin}}(k)}{2} \tag{16}$$

Proof. We transform game $\mathsf{Game}_{(2-1)}$ into $\mathsf{Game}_{(2-2)}$ computing values $X_1^*, X_2^*, Y_1^*, Y_2^*$ randomly as $X_1^*, X_2^*, Y_1^*, Y_2^* \leftarrow_{\$} G^4$ which lead to compute value Z^* from random values which make it random value. If adversary success in distinguishing between $\mathsf{Game}_{(2-1)}$ and $\mathsf{Game}_{(2-2)}$ with non-negligible probability, then he can solve the DLIN problem, thus we construct adversary $\mathcal{D}$that solve DLIN problem. In this game, $\mathcal{D}$will choose same parameters in $\mathsf{Game}_{(2-1)}$ except values $X_1^*, X_2^*, Y_1^*, Y_2^*$ which will be chosen randomly. Then he will query oracle machine for tuple $(X_{1,i}^*, X_{2,i}^*, Y_{1,i}^*, Y_{2,i}^*, \mathcal{A}, \mathcal{B})$, if a tuple exists oracle will return corresponding Z' to the adversary, else oracle will return random value to an adversary. So we can make q_{DLIN} queries oracle without repeating the same query to oracle. In case repeating the same query we will get halt with probability of:

$$\begin{aligned}\Pr[\perp] = C_2^{q_{DLIN}} &= \frac{q_{DLIN}!}{(q_{DLIN}-2)!2!} \\ &= \frac{q_{DLIN}(q_{DLIN}-1)}{2} \leq \frac{q_{DLIN}^2}{2}\end{aligned}$$

There for, we obtain:

$$\frac{q^2.\mathsf{Adv}_{\mathcal{D}}^{\mathrm{dlin}}(k)}{2}$$

– Game_{2-3}: We transform Game_{2-2} into Game_{2-3}, based on transform hash function $H(.)$ with random oracle function $\mathcal{O}$. For game state, see Appendix A.8.

Claim. let $\mathsf{G_3}$ be the event that $\mathcal{H}$ success in distinguishing value $H(.)$ from random oracle $\mathcal{O}(.)$ in Game_{2-3}. we claim that

$$\Pr[\mathsf{G_2}] - \Pr[\mathsf{G_3}] \leq \frac{q_H^2}{2}.\epsilon_{es}(k) \tag{17}$$

which ε_{es} is ES-advantage of some efficient algorithm(which is negligible assuming $\mathcal{H}$ is entropy smoothing).

Proof. We will prove here using the same idea in the previous game. In this game we transformed from Game_{2-2} by changing the hash value with a random value generated by oracle. Without losing of generality, The adversary will make q_H queries to oracle without a repeat of the same query. Same idea in previous game we can get the probability of halt as:

$$\begin{aligned}\Pr[\perp] = C_2^{q_H} &= \frac{q_H!}{(q_H-2)!2!} \\ &= \frac{q_H(q_H-1)}{2} \leq \frac{q_H^2}{2}\end{aligned}$$

The difference between $\Pr[\mathsf{G2}]$ and $\Pr[\mathsf{G3}]$ can be parlayed into a corresponding ES-advantage.
Moreover, as h act as a one-time pad in game Game_{2-3}, it's evident that

$$\Pr[\mathsf{G3}] = \frac{1}{2} \tag{18}$$

Combining (14),(15),(16),(17) and (18), we obtain

$$\mathsf{Adv}_{\mathcal{D}}^{\mathrm{dlin}}(k) \geq \frac{1}{q_{DLIN}^2}\left[\mathsf{Adv}_{\mathcal{M},\Pi}^{\mathrm{ake}}(k) - \frac{4}{n(k)^2 s(k)} - q_H^2.\epsilon_{es}(k)\right] \tag{19}$$

From the sequence of preceding claims, we can conclude that since the $\mathsf{Adv}_{\mathcal{D}}^{\mathrm{dlin}}(k) \geq \mathsf{Adv}_{\mathcal{M},\Pi}^{\mathrm{ake}}(k)$, and since $\mathsf{Adv}_{\mathcal{D}}^{\mathrm{dlin}}(k)$ is negligible in k - from DLIN assumption - thus our protocol is secure based on decision linear assumption in random oracle model.

5 Efficiency

In this section, we compare our protocols with other related AKE protocols in terms of based assumption, computational efficiency and security model. In Table 1 number of exponentiation in G (E), a number of static public keys (SPK)

Table 1. Protocols Comparison

Protocol	Computation	Security Model	Assumption	NAXOS Approch	SPK/EPK
Okamoto [9]	8E	eCK	Standard	Yes	2/3
HMQV [15]	2.5E	CK, wPFS,KCI, LEP	KEA1, GDH, RO	No	1/1
CMQV [16]	3E	eCK	GDH, RO	Yes	1/1
NAXOS [15]	4E	eCK	GDH, RO	Yes	1/1
NETS [8]	3E	eCK	GDH, RO	Yes	1/1
SMEN [17]	6E	eCK	GDH, RO	No	2/2
KFU [10]	3E	eCK	GDH, RO	No	2/1
Our	3E	eCK	DLIN, RO	No	2/2

and the number of ephemeral public key (EPK). Table 5 presents the naive group exponentiations count; Okamoto's protocol is secure in the standard model, but the proof relies on an existence of πPRF family. In the security proof of HMQV and CMQV, the reduction argument is less tight since the Forking Lemma [14] is essential for the arguments. Our protocol in Table 1, have tighter security reductions and do not use the Forking Lemma and just use one static public key in computation.
It clear that our protocol has same security model with NETS, CMQV, and KFU-P1, but it differs from them in base assumption and computation.
We showed that it is possible to construct eCK-secure AKE protocols without using NAXOS' approach, so our protocol is secure even when the discrete logarithm of the ephemeral public key is revealed and decrease the risk of leaking the static private key which makes our protocol more practical.
Moreover, One of the advantages of our protocols is the use of single random oracle as opposed to two for HMQV and CMQV. The random oracle is merely needed for the session key derivation, which is typical way to attain indistinguishability in random oracle model.
In addition, our protocol uses decision linear assumption with a tight security proof.

6 Conclusions

In this paper, we present AKE protocol secure in the eCK model under Decision Linear assumption(DLIN) without using NAXOS trick with a fastened reduction, which reduces the risk of leaking the static private key, that because of the derivation of the ephemeral public key is independent of the static private key. This is in contrast to protocols that use the NAXOS' approach. And minimize the use of the random oracle, by applying it only to the session key derivation. Moreover, each ephemeral and static key has its particular generator which gives tight security for the protocol. We gave tightly security proof for our protocol based on games. In this paper still remaining as open problem how to preserve the security of to this protocol without using random oracle.

References

1. M. Bellare and P. Rogaway.: "Entity authentication and key distribution". Crypto 1993, LNCS 773, pp. 110-125 (1993)
2. Bellare, Mihir, Ran Canetti, and Hugo Krawczyk.: "A modular approach to the design and analysis of authentication and key exchange protocols". Pro-ceedings of the thirtieth annual ACM symposium on Theory of computing. ACM (1998)
3. R. Canetti, H. Krawczyk.: "Analysis of key-exchange protocols and their use for building secure channels". Eurocrypt 2001, LNCS 2045, pp. 453-474 (2001)
4. B. LaMacchia, K. Lauter, and A. Mityagin.: "Stronger security of authenticated key exchange". ProvSec 2007, LNCS 4784, pp. 1-16 (2007)
5. B. Ustaoglu.: "Obtaining a secure and efficient key agreement protocol for (H)MQV and NAXOS". Designs, Codes and Cryptography, Vol. 46(3), pp. 329-342, 2008. Extended version available at http://eprint.iacr.org/2007/123.
6. H. Huang and Z. Cao.: "Strongly secure authenticated key exchange protocol based on computational Diffie-Hellman problem". Inscrypt 2008.
7. J. Lee and J. Park.: "Authenticated key exchange secure under the computational Diffie-Hellman assumption". http://eprint.iacr.org/2008/344.
8. J. Lee and C. Park.: "An efficient key exchange protocol with a tight security reduction". http://eprint.iacr.org/2008/345.
9. T. Okamoto.: "Authenticated key exchange and key encapsulation in the standard model". Asiacrypt 2007, LNCS 4833, pp.474-484, 2007.
10. Kim, Minkyu, Atsushi Fujioka, and Berkant Ustaolu.: "Strongly secure authenticated key exchange without NAXOSapproach". Advances in Infor-mation and Computer Security. Springer Berlin Heidelberg, 2009. 174-191.
11. D. Boneh, X. Boyen, and H. Shacham.: Short group signatures. In M. Franklin, editor, Proceedings of Crypto 2004, volume 3152 of LNCS, pages 41-55. Springer-Verlag, Aug. 2004.
12. A. Joux and K. Nguyen.: "Separating decision Diffie-Hellman from computa-tional Diffie-Hellman in cryptographic groups". J. Cryptology, 16(4):239-47, Sept. 2003.
13. V. Shoup.: "Lower bounds for discrete logarithms and related problems". In W. Fumy, editor,Proceedings of Eurocrypt 1997, volume 1233 of LNCS, pages 256-66. Springer-Verlag, May1997.
14. D. Pointcheval and J. Stern.: "Security Arguments for Digital Signatures and Blind Signatures". J. of Cryptology, Vol 13(3), pp. 361-396, 2000.
15. H. Krawczyk.: "HMQV: A high-performance secure Diffie-Hellman protocol". Crypto 2005, LNCS 3621, pp. 546-566, 2005.
16. B. Ustaoglu.: "Obtaining a secure and efficient key agreement protocol for (H)MQV and NAXOS". Designs, Codes and Cryptography, Vol. 46(3), pp. 329-342, 2008.
17. J. Wu and B. Ustaoglu.: "Efficient Key Exchange with Tight Security Reduction". Technical Report CACR 2009-23, University of Waterloo, 2009. Available at http://www.cacr.math.uwaterloo.ca/techreports/2009/cacr200923.pdf.
18. Li, Hui, and ChuanKun Wu.: "CMQV+: An authenticated key exchange protocol from CMQV". Science China Information Sciences 55.7 (2012): 1666-1674.
19. Mohamed, Mojahed, Xiaofen Wang, and Xiaosong Zhang. "Efficient Secure Authenticated Key Exchange Without NAXOSApproach Based on Decision Linear Problem." Collaborative Computing: Networking, Applications, and Worksharing. Springer International Publishing, 2015. 243-256.

A Adversary Games

A.1 Game_{1-0}

Game_{1-0}

$P = \bigcup_{i=1}^{n} P_i, SID = \bigcup_{i=1,j=i+1}^{n} sid_{i,j}$

$\overline{K}_0 \leftarrow_\$ \{0,1\}^k$

$\overline{K}_1 = H(Z^*, X_1^*, X_2^*, Y_1^*, Y_2^*, \mathcal{A}, \mathcal{B})$

$b \leftarrow_\$ \{0,1\}$

$sk^* = \overline{K}_b$

$b' \leftarrow_\$ \mathcal{A}(P, SID, sk^*) \in \{0,1\}$

return $b = b'$

A.2 Game_{1-1}

Game_{1-1}

$P = \bigcup_{i=1}^{n} P_i, SID = \bigcup_{i=1,j=i+1}^{n} sid_{i,j}, r \leftarrow_\$ R$

$sid_{\mathcal{A},\mathcal{B}}, sid_{\mathcal{B},\mathcal{A}} \leftarrow_\$ \mathcal{A}(P, SID, r)$

if $sid_{\mathcal{A},\mathcal{B}} \wedge sid_{\mathcal{B},\mathcal{A}} \in SID$

  $\perp$

else

  $\widehat{K}_0 \leftarrow_\$ \{0,1\}^k$

  $\widehat{K}_1 = H(Z^*, X_1^*, X_2^*, Y_1^*, Y_2^*, \mathcal{A}, \mathcal{B})$

  $b \leftarrow_\$ \{0,1\}$

  $sk^* = \widehat{K}_b$

fi

$b' \leftarrow_\$ \mathcal{A}(P, SID, sk^*) \in \{0,1\}$

return $b = b'$

A.3 Game_{1-2}

Game_{1-2}

$P = \bigcup_{i=1}^{n} P_i, SID = \bigcup_{i=1,j=i+1}^{n} sid_{i,j}, r \leftarrow_{\$} R$
$sid_{\mathcal{A},\mathcal{B}}, sid_{\mathcal{B},\mathcal{A}} \leftarrow_{\$} \mathcal{A}(P, SID, r)$
if $sid_{\mathcal{A},\mathcal{B}} \wedge sid_{\mathcal{B},\mathcal{A}} \in SID$
  $\perp$
else
  $Z^* \leftarrow_{\$} G^2$
  $\widehat{K}_0 \leftarrow_{\$} \{0,1\}^k$
  $\widehat{K}_1 = H(Z^*, X_1^*, X_2^*, Y_1^*, Y_2^*, \mathcal{A}, \mathcal{B})$
  $b \leftarrow_{\$} \{0,1\}$
  $sk^* = \widehat{K}_b$
fi
$b' \leftarrow_{\$} \mathcal{A}(P, SID, \widehat{Z}, sk^*) \in \{0,1\}$
return $b = b'$

A.4 Game_{1-3}

Game_{1-3}

$P = \bigcup_{i=1}^{n} P_i, SID = \bigcup_{i=1,j=i+1}^{n} sid_{i,j}, r \leftarrow_{\$} R$
$sid_{\mathcal{A},\mathcal{B}}, sid_{\mathcal{B},\mathcal{A}} \leftarrow_{\$} \mathcal{A}(P, SID, r)$
if $sid_{\mathcal{A},\mathcal{B}} \wedge sid_{\mathcal{B},\mathcal{A}} \in SID$
  $\perp$
else
  $Z^* \leftarrow_{\$} G^2$
  $\widehat{K}_0 \leftarrow_{\$} \{0,1\}^k$
  $\widehat{K}_1 = H(Z^*, X_1^*, X_2^*, Y_1^*, Y_2^*, \mathcal{A}, \mathcal{B})$
  $b \leftarrow_{\$} \{0,1\}$
  $sk^* = \leftarrow_{\$} \{0,1\}^k$
fi
$b' \leftarrow_{\$} \mathcal{A}(P, SID, \widehat{Z}, \delta, sk^*) \in \{0,1\}$
return $b = b'$

A.5 Game_{2-0}

Game_{2-0}

$P = \bigcup_{i=1}^{n} P_i, SID = \bigcup_{i=1,j=i+1}^{n} sid_{i,j}$

$\widehat{K}_0 \leftarrow_{\$} \{0,1\}^k$

$\widehat{K}_1 = H(Z^*, X_1^*, X_2^*, Y_1^*, Y_2^*, \mathcal{A}, \mathcal{B})$

$b \leftarrow_{\$} \{0,1\}$

$sk^* = \overline{K}_b$

$b' \leftarrow_{\$} \mathcal{A}(P, SID, sk^*) \in \{0,1\}$

return $b = b'$

A.6 Game_{2-1}

Game_{2-1}

$P = \bigcup_{i=1}^{n} P_i, SID = \bigcup_{i=1,j=i+1}^{n} sid_{i,j}, r \leftarrow_{\$} R$

$sid_{\mathcal{A},\mathcal{B}}, sid_{\mathcal{B},\mathcal{A}} \leftarrow_{\$} \mathcal{A}(P, SID, r)$

if $sid_{\mathcal{A},\mathcal{B}} \wedge sid_{\mathcal{B},\mathcal{A}} \in SID$

  $\perp$

else

  $\widehat{K}_0 \leftarrow_{\$} \{0,1\}^k$

  $\widehat{K}_1 = H(Z^*, X_1^*, X_2^*, Y_1^*, Y_2^*, \mathcal{A}, \mathcal{B})$

  $b \leftarrow_{\$} \{0,1\}$

  $sk^* = \widehat{K}_b$

fi

$b' \leftarrow_{\$} \mathcal{A}(P, SID, sk^*) \in \{0,1\}$

return $b = b'$

A.7 Game_{2-2}

Game_{2-2}

$P = \bigcup_{i=1}^{n} P_i, SID = \bigcup_{i=1,j=i+1}^{n} sid_{i,j}, r \leftarrow_{\$} R$

$sid_{\mathcal{A},\mathcal{B}}, sid_{\mathcal{B},\mathcal{A}} \leftarrow_{\$} \mathcal{A}(P, SID, r)$

if $sid_{\mathcal{A},\mathcal{B}} \wedge sid_{\mathcal{B},\mathcal{A}} \in SID$

$\perp$

else

// $\mathrm{z}^{list} \equiv (W_1, W_2, W_1{}', W_2{}', P_i, P_j, Z') \in (G^4, \{0,1\}^*, \{0,1\}^*, G^2)$

for $i ... q_{DLIN}$ **do** $X^*_{1,i}, X^*_{2,i}, Y^*_{1,i}, Y^*_{2,i} \leftarrow_{\$} G$

$\delta_i = (X^*_{1,i}, X^*_{2,i}, Y^*_{1,i}, Y^*_{2,i}, \mathcal{A}, \mathcal{B})$

$\delta_i \leftarrow \mathcal{A}(P, SID, \widehat{Z}_1, .., \widehat{Z}_{i-1})$

if $(\delta \in Z^{list})$

$\widehat{Z}_i = SK$

else

$\widehat{Z}_i \leftarrow_{\$} \{0,1\}G$

fi

endfor

$\rho \leftarrow \mathcal{A}(P, SID, \widehat{Z}_1, .., \widehat{Z}_{q_{DLIN}}) \in G^2$

$\widehat{K}_0 \leftarrow_{\$} \{0,1\}^k$

$\widehat{K}_1 = H(\rho, X^*_1, X^*_2, Y^*_1, Y^*_2, \mathcal{A}, \mathcal{B})$

$b \leftarrow_{\$} \{0,1\}$

$sk^* = \widehat{K}_b$

fi

$b' \leftarrow_{\$} \mathcal{A}(P, SID, \widehat{Z}, sk^*) \in 0, 1$

return $b = b'$

A.8 $\mathbf{Game_{2-3}}$

Game_{2-3}

$P = \bigcup_{i=1}^{n} P_i, SID = \bigcup_{i=1,j=i+1}^{n} sid_{i,j}, r \leftarrow_{\$} R$

$sid_{\mathcal{A},\mathcal{B}}, sid_{\mathcal{B},\mathcal{A}} \leftarrow_{\$} \mathcal{A}(P, SID, r)$

if $sid_{\mathcal{A},\mathcal{B}} \wedge sid_{\mathcal{B},\mathcal{A}} \in SID$

$\perp$

else

// $\mathrm{Z}^{list} \equiv (W_1, W_2, {W_1}', {W_2}', P_i, P_j, Z') \in (G^4, \{0,1\}^*, \{0,1\}^*, G^2)$

for $i ... q_{DLIN}$ **do** $X^*_{1,i}, X^*_{2,i}, Y^*_{1,i}, Y^*_{2,i} \leftarrow_{\$} G$

$\delta_i = (X^*_{1,i}, X^*_{2,i}, Y^*_{1,i}, Y^*_{2,i}, \mathcal{A}, \mathcal{B})$

$\delta_i \leftarrow \mathcal{A}(P, SID, \widehat{Z}_1, .., \widehat{Z}_{i-1})$

if $(\delta \in Z^{list})$

$\widehat{Z}_i = SK$

else

$\widehat{Z}_i \leftarrow_{\$} \{0,1\}G$

fi

endfor // $\mathrm{H}^{list} \equiv (Z', W_1, W_2, {W_1}', {W_2}', P_i, P_j, SK) \in (G^6, \{0,1\}^*, \{0,1\}^*, \{0,1\}^k)$

for $i ... q_H$ **do** $X^*_{1,i}, X^*_{2,i}, Y^*_{1,i}, Y^*_{2,i} \leftarrow_{\$} G$

$\delta_i = (Z'_i, X^*_{1,i}, X^*_{2,i}, Y^*_{1,i}, Y^*_{2,i}, \mathcal{A}, \mathcal{B})$

$\delta_i \leftarrow \mathcal{A}(P, SID, \widehat{Z}, SK'_1, .., SK'_{i-1})$

if $(\delta \in H^{list})$

$SK'_i = SK$

else

$SK'_i \leftarrow_{\$} \{0,1\}\{0,1\}^k$

fi

endfor

$\rho \leftarrow \mathcal{A}(P, SID, \widehat{Z}, SK'_1, .., SK'_{i-1}) \in G^2$

$\widehat{K}_0 \leftarrow_{\$} \{0,1\}^k$

$\widehat{K}_1 = H(\rho, X^*_1, X^*_2, Y^*_1, Y^*_2, \mathcal{A}, \mathcal{B})$

$b \leftarrow_{\$} \{0,1\}$

$sk^* = \widehat{K}_b$

fi

$b' \leftarrow_{\$} \mathcal{A}(P, SID, \widehat{Z}, sk^*) \in 0, 1$

return $b = b'$