Authenticated Key Exchange Protocols with Enhanced Freshness Properties

Hai Huang, Zhenfu Cao

***Abstract*—In this paper, we investigate the security model for authenticated key exchange protocols. We observe that there is further room to extend the latest enhanced Canetti-Krawczyk (eCK) model.**

We further enhance the freshness definition for the three-pass authenticated key exchange protocols such that our new definition gives the adversary more capabilities. We point out that the three-pass authenticated key exchange protocols generically transformed from the two-pass authenticated key exchange protocols secure in the eCK model can not be secure in our new security definition. We then introduce a new authenticated key exchange protocol SIG-DH$^+$ and prove that it satisfies our new definition.

***Index Terms*—Authenticated key exchange, Random oracle, Provably secure, eCK model**

I. Introduction

Key exchange (KE) is a traditional primitive of cryptography. It enables two parties, Alice (A) and Bob (B), to establish a shared session key over an insecure channel. Later, the shared session key can be used to efficiently ensure data confidentiality and integrity between A and B using efficient symmetric encryptions and message authentication codes.

The seminal paper of Diffie and Hellman [9] provides the first solution called Diffie-Hellman (DH) key exchange protocol to this issue. However, the original DH key exchange protocol is only secure against a passive adversary. To resist an active adversary (an active person in-the-middle), we need to provide the basic DH protocol with authentication. This is authenticated key exchange (AKE) in which both parties are assured that no other parties aside from their intended peers may learn the established session key.

A key exchange protocol is said to provide key confirmation, if both parties are sure that the intended peers really hold the session key. A protocol which is an authenticated key exchange with key confirmation protocol is called AKC protocol [4].

A lot of desirable properties for AKE protocols have been identified:

- *Known-key security*: It is reasonable to assume the adversary has the ability to learn the session keys except for the one under attack. A protocol is said to be known-key secure if the compromise of one session key should not compromise other session keys.
- *Forward security*: If the static keys of one party or two parties are compromised, the adversary can not obtain the previously established session keys.
- *Key compromise impersonation resistance*: Suppose A's static key is compromised. Clearly, the adversary can arbitrarily masquerade as A in future. However, we want to guarantee that the adversary cannot masquerade as another party B to communicate with party A.

Hai Huang and Zhenfu Cao (email: {chinesechess@sjtu.edu.cn, zfcao@cs.sjtu.edu.cn}).

- *Ephemeral key reveal resistance*: If the adversary obtains the ephemeral keys of the related sessions, the session key under attack still remains secure.

The authenticated key exchange protocols have been established to be surprisingly difficult to design. The traditional trial-and-error design method has led to the situation that the flaws in the protocols have taken many years to discover. This has highlighted the importance of examining these protocols in a formal security model.

Bellare and Rogaway [2] first propose a formal security model for authentication and key distribution. They model the adversary's capability by providing it with oracle queries, e.g. Send, Reveal and Test queries. Since then, there have been several extensions to the model [3], [1], [6]. Choo, Boyd and Hitchcock [8] compare the most commonly used security models for key exchange protocols. All these models attempt to cover as many of these properties as possible.

A. Related Work

The CK model. Canetti and Krawczyk [6] extend the security definition for the AKE protocols by adding the SessionStateReveal queries, which allows the adversary to reveal the internal state information of the parties. However, the CK model prohibits the adversary from making SessionStateReveal queries against the Test session and its matching session.

Another potential weakness of the CK model is the Corrupt query. Once corrupted, the party will be under the control of the adversary. From the moment on, the adversary is no longer allowed to attack these sessions owned by the corrupted party (no longer fresh). This actually excludes the key compromise impersonation (KCI) attack.

Canetti and Krawczyk [6] provide a generic construction called authenticator which transforms a two-pass DH protocol secure in the authenticated model (AM) to a three-pass AKE protocol named SIG-DH secure in the unauthenticated model (UM).

The eCK model. Recently, LaMacchia, Lauter and Mityagin [14], [15] present a new security model for AKE protocols, the enhanced Canetti-Krawczyk (eCK). The eCK model removes the SessionStateReveal query and instead introduces a new kind of query called EphemeralKeyReveal, which models the adversary's capability to learn the randomness of the target session instead of its internal states.

Another change is that the eCK model replaces the Corrupt query with the StaticKeyReveal query by which the adversary learns the static key of the target party without fully controlling the party. By this, the eCK model covers the KCI attack.

In their paper, the authors give the enhanced freshness definitions for both two-pass and three-pass AKE protocols.

More specifically, the adversary's capability is enhanced to the extent that the adversary is allowed to make arbitrary queries against the Test session and its matching session except for both EphemeralKeyReveal and StaticKeyReveal queries against one of the parties.

The authors also introduce a two-pass AKE protocol named NAXOS [14], [15] and show that it is secure under the gap assumption [17] in the eCK model. For other two-pass AKE protocols in the eCK model, see [16], [20].

Two-pass vs. three-pass. As shown by Krawczyk [13], if the adversary is actively involved with the choice of the DH values X, Y at a session, no two-pass AKE protocols can achieve forward security. So the best the two-pass AKE protocols can achieve is the weak form of forward security (wFS). In addition, the three-pass AKE protocols can provide key confirmation property while no two-pass AKE protocols can.

By adding two message authentication codes (MAC) keyed with the session keys generated by two parties into the two-pass AKE protocols, we can get the corresponding three-pass AKE protocols. This is an established method to transform the two-pass AKE protocols to the three-pass AKE protocols [1], [13].

While the freshness definition for the two-pass AKE protocols in the eCK model above is very strong one, there seems to be room to improve the freshness definition for the three-pass AKE protocols due to the reason which will be further explained in section III-B.

B. Our Contributions

We further enhance the freshness definition for the three-pass AKE protocols in the eCK model. More specifically, in the case that no sessions matching to the Test session exist, we allow that the adversary makes both StaticKeyReveal and EphemeralKeyReveal queries against the Test session (Note that in the eCK model, the adversary is just allowed to make one of these two queries). By this, our new definition gives the adversary more capabilities. We point out that those three-pass AKE protocols transformed from two-pass AKE protocols secure in the eCK model by the method above (adding two MACs) can not be secure in our new security definition for the three-pass AKE protocols.

We then introduce a new authenticated key exchange protocol SIG-DH$^+$ using a generic deterministic signature scheme and prove that it satisfies our new definition.

C. Organization

The paper is organized as follows. In section II, we review the related building techniques. In section III we review the eCK security model and propose our improved one. Then we propose our generic three-pass AKE protocol and give the security proof in our new model in section IV. Finally, concluding remarks are made in section V.

II. Preliminaries

In this section, we present several established results and tools needed in this paper.

A. CDH Assumption

Let the value κ be the security parameter. Let $\mathbb{G} = \langle g \rangle$ be a cyclic group of prime order q and $g \in \mathbb{G}$ be the generator. Define $\text{CDH}(X, Y) := Y^x$ where $X = g^x, Y = g^y$.

For any probabilistic polynomial time (PPT) algorithm A,

$$Pr[A(\mathbb{G}, g, X = g^x, Y = g^y) = \text{CDH}(X, Y)] \leq \epsilon(\kappa).$$

where $x, y \in \mathbb{Z}_q$ and $\epsilon(\kappa)$ is negligible. The probability is taken over the coin tosses of A, the choice of g and the random choices of x, y in $\mathbb{Z}_q$.

We denote by $Succ_{\mathbb{G}}^{cdh}(\kappa)$ the adversary's success probability.

B. Digital Signature Scheme [11]

A signature scheme Σ:=(Gen,Sign,Verify) consists of the following algorithms:

- Gen: A probabilistic algorithm that on input a security parameter 1^κ outputs a private key sk and public key pk.
- Sig: On input a private key sk and message $m \in \{0,1\}^\kappa$, the algorithm (possibly probabilistic) outputs a signature δ.
- Verify: On input a public key pk, message m and its signature δ, the algorithm (deterministic) outputs 1 if the signature is valid. Otherwise outputs 0.

A signature is called to be existentially unforgeable under chosen message attacks (EU-CMA), if for any polynomial time adversary with access to the oracle $\text{Sig}(sk, \cdot)$, the probability $\text{Succ}_{\Sigma}^{eu-cma}(\kappa)$ that the adversary outputs a pair (m, δ) such that $\text{Verify}(pk, m, \delta)$=1 but the adversary never makes oracle queries on message m is negligible.

III. Security Model

A. Review of the eCK model

We first review the original eCK model, in which authors give the freshness definitions for both the two-pass and three-pass AKE protocols[1]. However, in this section we just consider three-pass AKE protocols. For the details of the original eCK model, see [14], [15].

Participants. We model the protocol participants as a finite set U of fixed size with each ID_i being a probabilistic polynomial time (PPT) Turing machine. Each protocol participant $ID_i \in U$ may execute a polynomial number of protocol instances in parallel. We will refer to s-th instance of participant ID_i communicating with peer ID_j as $\Pi^s_{ID_i,ID_j}(i, j \in N)$ (*a session* or *an instance*).

Adversary Model. The adversary M is modeled as a PPT Turing machine and has full control of the communication network and may eavesdrop, delay, replay, alter and insert messages at will. We model the adversary's capability by providing it with oracle queries.

- **EphemeralKeyReveal**($\Pi^s_{ID_i,ID_j}$) The adversary obtains the ephemeral private key of $\Pi^s_{ID_i,ID_j}$. These queries

[1]Actually, the paper in [14] considers the case for both the three-pass and two-pass AKE protocols and the proceedings paper [15] just deals with the case for the two-pass AKE protocols.

are motivated by practical scenarios, such as if session-specific secret information is stored in insecure memory on device or if the random number generator of the party is corrupted.

- **SessionKeyReveal**($\Pi^s_{ID_i,ID_j}$) The adversary obtains the session key for a session s of ID_i, provided that the session holds a session key.
- **StaticKeyReveal**(ID_i) The adversary obtains the static private key of ID_i.
- **EstablishParty**(ID_i) The query models that the adversary can arbitrarily register a legal user on behalf of the party ID_i. In this way the adversary gets the party ID_i's static private key and totally controls the party ID_i. Parties against whom the adversary does not issue this query are called *honest*.
- **Send**($\Pi^s_{ID_i,ID_j}, m$) The adversary sends the message m to the session s executed by ID_i communicating with ID_j and gets a response according to the protocol specification.
- **Test**($\Pi^s_{ID_i,ID_j}$) Only one query of this form is allowed for the adversary. Provided that the session key is defined, the adversary M can execute this query at any time. Then depending on a randomly chosen bit b, with probability 1/2 the session key and with probability 1/2 a uniformly chosen random value $\zeta \in \{0,1\}^\kappa$ is returned.

Definition 1 (Matching Session): Let $\Pi^s_{ID_i,ID_j}$ be a completed session with public output (ID_i, X, Y, ID_j), where ID_i is the owner of the session, ID_j is the peer, and X is ID_i's outgoing message, Y is ID_j's outgoing message. The session $\Pi^t_{ID_j,ID_i}$ is called the *matching session* of $\Pi^s_{ID_i,ID_j}$, if the output of $\Pi^t_{ID_j,ID_i}$ is (ID_j, Y, X, ID_i) (note that $\Pi^t_{ID_j,ID_i}$ may still be incomplete).

Definition 2 (Freshness for Three-Pass AKE Protocols): Let instance $\Pi^s_{ID_i,ID_j}$ be a completed session, which was executed by an honest party ID_i with another honest party ID_j. We define $\Pi^s_{ID_i,ID_j}$ to be *fresh* if none of the following three conditions hold:

- The adversary M reveals the session key of $\Pi^s_{ID_i,ID_j}$ or of its matching session (if the latter exists).
- ID_j is engaged in session $\Pi^t_{ID_j,ID_i}$ matching to $\Pi^s_{ID_i,ID_j}$ and M either reveals:
  - both **StaticKey** of ID_i and **EphemeralKey** of $\Pi^s_{ID_i,ID_j}$; or
  - both **StaticKey** of ID_j and **EphemeralKey** of $\Pi^t_{ID_j,ID_i}$.
- No sessions matching to $\Pi^s_{ID_i,ID_j}$ exist and M either reveals:
  - both **StaticKey** of ID_i and **EphemeralKey** of $\Pi^s_{ID_i,ID_j}$; or
  - **StaticKey** of ID_j before the completion of $\Pi^s_{ID_i,ID_j}$.

Definition 3 (AKE Security): As a function of the security parameter κ, we define the advantage $Adv^{ake}_{M,\Sigma}(\kappa)$ of the PPT adversary M in attacking protocol Σ as

$$Adv^{ake}_{M,\Sigma}(\kappa) \stackrel{def}{=} |2 \cdot Succ^{ake}_{M,\Sigma}(\kappa) - 1|$$

Here $Succ^{ake}_{M,\Sigma}$ is the probability that the adversary queries the **Test** oracle to a *fresh* instance $\Pi^s_{ID_i,ID_j}$, outputs a bit $\hat{b}$ such that $\hat{b} = b$, where the bit b is used by the **Test** oracle.

We call the authenticated key exchange protocol Σ to be AKE secure if for any PPT adversary M the function $Adv^{ake}_{M,\Sigma}(\kappa)$ is negligible.

B. Enhancing the definition of AKE security

While the freshness Definition 2 gives the adversary very strong capabilities, we find that there still is room for improvement. Note that in the case that there are no sessions matching to the Test session $\Pi^s_{ID_i,ID_j}$, the Test session $\Pi^s_{ID_i,ID_j}$ will reject this session except for negligible probability, i.e. the Test session does not generate a session key at all in this case. So it is not necessary to prohibit both the StaticKeyReveal(ID_i) and the EphemeralKeyReveal($\Pi^s_{ID_i,ID_j}$) queries as the freshness Definition 2 does. In view of this, we give our new freshness definition.

Definition 4 (Enhanced Freshness for Three-Pass AKE Protocols): Let instance $\Pi^s_{ID_i,ID_j}$ be a completed session, which was executed by an honest party ID_i with another honest party ID_j. We define $\Pi^s_{ID_i,ID_j}$ to be *fresh* if none of the following three conditions hold:

- The adversary M reveals the session key of $\Pi^s_{ID_i,ID_j}$ or of its matching session (if the latter exists).
- ID_j is engaged in session $\Pi^t_{ID_j,ID_i}$ matching to $\Pi^s_{ID_i,ID_j}$ and M either reveals:
  - both **StaticKey** of ID_i and **EphemeralKey** of $\Pi^s_{ID_i,ID_j}$; or
  - both **StaticKey** of ID_j and **EphemeralKey** of $\Pi^t_{ID_j,ID_i}$.
- No sessions matching to $\Pi^s_{ID_i,ID_j}$ exist and M reveals:
  - **StaticKey** of ID_j before the completion of $\Pi^s_{ID_i,ID_j}$.

Definition 5 (Enhanced AKE Security): We say that an AKE protocol satisfies the enhanced AKE security if it satisfies the Definition 3 where the freshness from Definition 4.

In contrast to the freshness Definition 2, our new freshness Definition 4 relaxes the third condition such that the adversary is allowed to reveal *both* ID_i's static key *and* the ephemeral key of $\Pi^s_{ID_i,ID_j}$. In other words, our enhanced AKE definition gives the adversary stronger capabilities.

Remark 1: We claim that the proven three-pass AKE protocols such as SIGMA [12], [7] and SIG-DH [6] are clearly not secure in our new model (even in the eCK model) since the final session key is only determined by g^{xy} where $X = g^x, Y = g^y$ are two ephemeral DH values which can be directly obtained via EphemeralKeyReveal queries against Test session and its matching session by the adversary.

What we want to stress here is that even the three-pass AKE protocols transformed from the two-pass AKE protocols [14], [15], [16], [20] secure in the eCK model are *no longer* secure in our new model either. For clarity of exposition, Fig. 1 illustrates the generic construction in which we can obtain a three-pass AKE protocol by merging messages.

According to the freshness Definition 4, in our new model the adversary can obtain *both* ID_i's static key *and* the

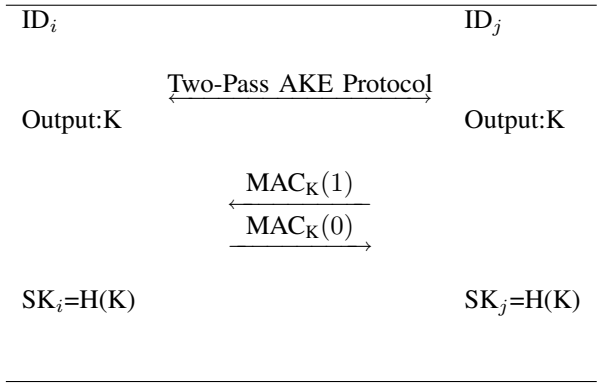

Fig. 1. The Generic Three-Pass Authenticated Key Exchange Protocol

ephemeral key of $\Pi^s_{ID_i,ID_j}$ provided that no sessions matching to $\Pi^s_{ID_i,ID_j}$ exist. We give the attack against this protocol as follows:

First the adversary masquerades as some party, say ID$_j$, to take part in the two-pass AKE protocol with ID$_i$. Then the adversary reveals *both* ID_i's static key *and* the ephemeral key of $\Pi^s_{ID_i,ID_j}$. Next with these values it can compute the value K and produce the message authentication code MAC$_K(1)$ itself. Finally, ID$_i$ will accept this session and believe its peer is ID$_j$.

IV. The SIG-DH$^+$ protocol

In this section, we propose a new three-pass AKE protocol called SIG-DH$^+$ which is secure in our enhanced AKE security definition.

A. Setup

Let the value κ be the security parameter. Let $\mathbb{G} = \langle g \rangle$ be a cyclic group of order q with a generator $g \in \mathbb{G}$. Let $\mathbb{G}^*$ be the non-identity elements set of $\mathbb{G}$. Let $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q$ and $H : \{0,1\}^* \rightarrow \{0,1\}^\kappa$ be two hash functions modeled as random oracles. We denote by pk_U, sk_U the public key and private key of party U respectively. Let $SIG_U(m)$ be a deterministic signature on message m generated by party U using its private key sk_U.

B. Protocol description

Assume Alice(A) and Bob (B) are two parties.

1. A (initiator) chooses an ephemeral private key $\tilde{x} \in \mathbb{Z}_q$ at random, computes and sends $X = g^{H_1(\tilde{x}, sk_A)}$ to B.
2. Upon receiving X, party B (responder) verifies that $X \in \mathbb{G}^*$. If so, B randomly chooses $\tilde{y} \in \mathbb{Z}_q$, computes $Y = g^{H_1(\tilde{y}, sk_B)}$ and sends $B, Y, SIG_B(B, Y, X, A)$ to party A.
3. Upon receiving message from B, party A checks if $Y \in \mathbb{G}^*$ and verifies the signature. If so, A returns $SIG_A(A, X, Y, B)$ to party B, and keeps $K_A = H(Y^{H_1(\tilde{x}, sk_A)}, sid_A)$ as the session key where $sid_A = (X, Y, A, B)$.
4. Party B verifies the received signature. If so, B keeps $K_B = H(X^{H_1(\tilde{y}, sk_B)}, sid_B)$ as the session key where $sid_B = (X, Y, A, B)$.

C. Rationale

In contrast to the SIG-DH protocol [6] where each party, say A, chooses the ephemeral key x and sends $X = g^x$, the main difference in the SIG-DH$^+$ protocol is that party A sends $X = g^x$, where x is computed by combining the ephemeral key $\tilde{x}$ and the static key sk_A.

Intuitively, by combining the ephemeral key and static key, the SIG-DH$^+$ protocol can resist those attacks that reveal one of the ephemeral key and static key, while the SIG-DH protocol is vulnerable to the attacks which only reveal the ephemeral key. The technique has been used to construct the *two-pass* AKE protocol in the eCK model [14], [15], [20].

On the other hand, as shown in [14], [15], if the signatures in the SIG-DH protocol are instantiated by a randomized signature such as ElGamal [10], Schnorr [18], the adversary can obtain the long-term secret key of the party by revealing the random coins used in signature generation. This is why we use the deterministic signature[2] in the SIG-DH$^+$ protocol.

Moreover, in the SIG-DH$^+$ protocol each party, say B, only uses its static key sk_B to generate the signature. Without the static key sk_B, it is not possible for the adversary to masquerade as party B against its peer party A even if the adversary obtains the static key and ephemeral key of party A.

D. Security Proof

Theorem 1: Suppose that the CDH assumption for group $\mathbb{G}$ holds, the signature scheme is deterministic and EU-CMA secure, H_1, H are hash functions modeled as random oracles, then the proposed scheme in Fig. 2 is a secure authenticated key exchange protocol in the sense of the Definition 5. The adversary M's advantage $Adv^{ake}_M(\kappa)$ is bounded by

$$2\left(\frac{q_s^2}{2^\kappa} + \frac{(2q_s + q_r)^2}{2^\kappa} + n \cdot Succ^{eu-cma}_\Sigma(\kappa) + q_s \cdot Succ^{cdh}_{\mathbb{G}}(\kappa)\right)$$

where n is the number of honest parties activated by the adversary, q_s and q_r are the upper bound on the maximum number of protocol sessions and random oracle queries by the adversary, $Succ^{eu-cma}_\Sigma(\kappa)$ is the adversary's probability in breaking the deterministic signature scheme.
The security proof of this theorem appears in Appendix A.

V. Conclusions

In this paper, we have investigated the security model for authenticated key exchange protocols.

We further enhance the freshness definition for the three-pass authenticated key exchange protocols. We point out that those three-pass AKE protocols generically transformed from two-pass AKE protocols secure in the eCK model can not be secure in our new security model for the three-pass AKE protocols.

We then introduce a new authenticated key exchange protocol SIG-DH$^+$ using a generic deterministic signature scheme and prove that it satisfies our new definition.

[2]The short signature proposed by Boneh,Lynn and Shacham (BLS) [5] is an example of the deterministic signature.

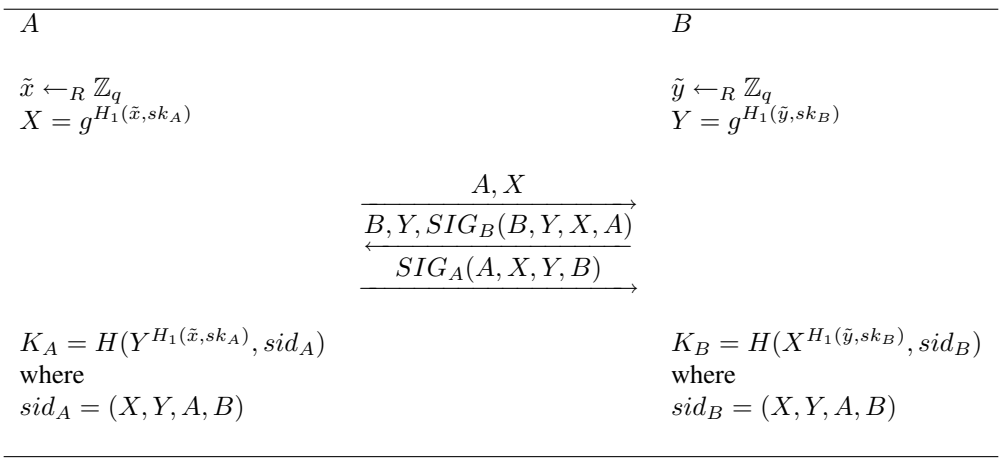

Fig. 2. The proposed SIG-DH$^+$ protocol

REFERENCES

[1] M. Bellare, D. Pointcheval, and P. Rogaway. Authenticated key exchange secure against dictionary attacks. In *EUROCRYPT*, pages 139–155, 2000.

[2] M. Bellare and P. Rogaway. Entity authentication and key distribution. In D. R. Stinson, editor, *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 232–249. Springer, 1993.

[3] M. Bellare and P. Rogaway. Provably secure session key distribution: the three party case. In *STOC*, pages 57–66. ACM, 1995.

[4] S. Blake-Wilson, D. Johnson, and A. Menezes. Key agreement protocols and their security analysis. In M. Darnell, editor, *IMA Int. Conf.*, volume 1355 of *Lecture Notes in Computer Science*, pages 30–45. Springer, 1997.

[5] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. *In Proceedings of Asiacrypt 2001, volume 2248 of LNCS*, 2001.

[6] R. Canetti and H. Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In B. Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 453–474. Springer, 2001.

[7] R. Canetti and H. Krawczyk. Security analysis of IKE's signature-based key-exchange protocol. In M. Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 143–161. Springer, 2002.

[8] K.-K. R. Choo, C. Boyd, and Y. Hitchcock. Examining indistinguishability-based proof models for key establishment protocols. In B. K. Roy, editor, *ASIACRYPT*, volume 3788 of *Lecture Notes in Computer Science*, pages 585–604. Springer, 2005.

[9] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[10] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[11] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[12] H. Krawczyk. Sigma: The 'SIGn-and-MAc' approach to authenticated diffie-hellman and its use in the ike-protocols. In D. Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 400–425. Springer, 2003.

[13] H. Krawczyk. HMQV: A high-performance secure Diffie-Hellman protocol. In V. Shoup, editor, *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 546–566. Springer, 2005.

[14] B. LaMacchia, K. Lauter, and A. Mityagin. Stronger security of authenticated key exchange. Cryptology ePrint Archive, Report 2006/073, 2006. http://eprint.iacr.org.

[15] B. A. LaMacchia, K. Lauter, and A. Mityagin. Stronger security of authenticated key exchange. In W. Susilo, J. K. Liu, and Y. Mu, editors, *ProvSec*, volume 4784 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2007.

[16] T. Okamoto. Authenticated key exchange and key encapsulation in the standard model. In *ASIACRYPT*, pages 474–484, 2007.

[17] T. Okamoto and D. Pointcheval. The gap-problems: A new class of problems for the security of cryptographic schemes. In K. Kim, editor, *Public Key Cryptography*, volume 1992 of *Lecture Notes in Computer Science*, pages 104–118. Springer, 2001.

[18] C. Schnorr. Efficient signature generation by smart cards. *Journal of cryptology*, 4(3):161–174, 1991.

[19] V. Shoup. Sequences of games: a tool for taming complexity in security proofs. *IACR eprint report*, 332, 2004.

[20] B. Ustaoglu. Obtaining a secure and efficient key agreement protocol from (H)MQV and NAXOS. *Des. Codes Cryptography*, 46(3):329–342, 2008.

APPENDIX A

Theorem 1. *Suppose that the CDH assumption for group $\mathbb{G}$ holds, the signature scheme is deterministic and EU-CMA secure, H_1, H are hash functions modeled as random oracles, then the proposed scheme in Fig. 2 is a secure authenticated key exchange protocol in the sense of the Definition 5. The adversary M's advantage $Adv_M^{ake}(\kappa)$ is bounded by*

$$2\left(\frac{q_s^2}{2^\kappa} + \frac{(2q_s + q_r)^2}{2^\kappa} + n \cdot Succ_\Sigma^{eu-cma}(\kappa) + q_s \cdot Succ_{\mathbb{G}}^{cdh}(\kappa)\right)$$

Proof. We give the proof using a sequence of games [19]. We let the adversary M interact with simulator S, who offers the real protocol environment to M in the first game, and subsequently change the simulator's behavior without affecting the adversary M's success probability significantly. In the last game, the adversary M's success probability is equal to (or negligibly close to) the target probability.

Assume that n is the number of the honest parties activated by the adversary, q_s and q_r are the maximum number of the sessions and random oracle queries by the adversary. Let S_i be the event that the adversary M correctly guesses the bit b used in the Test session in Game i and ϵ_i is the adversary M's advantage in Game i. i.e. $\epsilon_i = |2 \cdot Pr[S_i] - 1|$. We have

$$\begin{aligned} \epsilon_{i-1} &= |2 \cdot Pr[S_{i-1}] - 1| \\ &\leq |2 \cdot Pr[S_{i-1}] - 2 \cdot Pr[S_i]| + |2 \cdot Pr[S_i] - 1| \\ &\leq 2 \cdot |Pr[S_{i-1}] - Pr[S_i]| + \epsilon_i \ (i = 1, 2, ..., 6) \end{aligned}$$

Game 0: This game corresponds to the real attack. By definition, we have

$$Adv_{M,\Sigma}^{ake}(\kappa) \stackrel{def}{=} |2 \cdot Pr[S_0] - 1| = \epsilon_0$$

Game 1: The game is the same as Game 0 except that S fails if an event **Repeat** occurs. Hence

$$|Pr[S_1] - Pr[S_0]| \leq Pr[\textbf{Repeat}]; \qquad \epsilon_0 \leq 2 \cdot Pr[\textbf{Repeat}] + \epsilon_1$$

The event **Repeat** happens when a party chooses two identical random ephemeral keys in two different sessions. As there are at most q_s sessions, we have

$$Pr[Repeat] \leq \frac{q_s^2}{2^\kappa}.$$

Game 2: The game is the same as Game 1 except that S fails if an event **Collusion** occurs. Hence

$$|Pr[S_2] - Pr[S_1]| \leq Pr[\textbf{Collision}]; \qquad \epsilon_0 \leq 2 \cdot Pr[\textbf{Collision}] + \epsilon_1$$

The event **Collision** happens when the random oracles H_1, H produces a collusion for any of inputs. Each session requires two random oracles queries. Since there are at most q_s sessions and q_r oracle queries by the adversary, the probability of the event **Collision** is

$$Pr[\textbf{Collision}] \leq \frac{(2q_s + q_r)^2}{2^\kappa}.$$

Game 3: This game is identical to Game 2 except that the simulator aborts if event **No-Matching** occurs. We have

$$|Pr[S_3] - Pr[S_2]| \leq Pr[\textbf{No-Matching}]; \qquad \epsilon_2 \leq 2 \cdot Pr[\textbf{No-Matching}] + \epsilon_3.$$

The event **No-Matching** happens if some session $\Pi^s_{ID_i,ID_j}$ accepts and there are no sessions $\Pi^t_{ID_j,ID_i}$ matching to it. Here we require that the adversary has not asked StaticKeyReveal(ID_j) query before the completion of $\Pi^s_{ID_i,ID_j}$. In order to estimate the probability of the event **No-Matching**, we show how to use the adversary M to construct an EU-CMA forger $\bar{F}$ against the signature scheme as follows:

$\bar{F}$'s operation: $\bar{F}$ is given a public key pk and has access to the corresponding signature oracle. $\bar{F}$'s goal is to forge a signature corresponding to public key pk.

$\bar{F}$ randomly chooses one of n honest parties, say ID_j, and sets party ID_j's public key to be pk. For all other honest parties, $\bar{F}$ assigns the public/private key pairs itself. All the adversary M's Send queries to parties can be answered by the simulator, since the simulator has their private keys, except for the party ID_j whose public key is pk.

To answer Send queries to party ID_j (acting as responder) , the simulator $\bar{F}$ chooses $\tilde{y}, y \in \mathbb{Z}_q$, computes $Y = g^y$, and calls its signature oracle to get its response $(ID_j, Y, SIG_{ID_j}(ID_j, Y, X, ID_i))$, where ID_i is party ID_j's peer and X is party ID_i's outgoing message. Later on, the value $\tilde{y}$ is kept as the ephemeral key and used to answer the EphemeralKeyReveal query. Now the simulation provided by $\bar{F}$ is accurate.

On the other hand, since the session $\Pi^s_{ID_i,ID_j}$ may be an initiator session or a responder session, we consider two subcases below.

CASE 1: Assume that $\Pi^s_{ID_i,ID_j}$ is an initiator session.

If $\Pi^s_{ID_i,ID_j}$ is invoked as an initiator oracle, then at some time τ_0, $\Pi^s_{ID_i,ID_j}$ receives the activation flag and returns $X = g^{H_1(\tilde{x}, sk_{ID_i})}$ to the adversary. If $\Pi^s_{ID_i,ID_j}$ is to accept, it must receive a flow of the form $(ID_j, Y = g^y, SIG_{ID_j}(ID_j, Y, X, ID_i))$, where X is produced by the simulator $\bar{F}$. If the adversary M succeeds against the session $\Pi^s_{ID_i,ID_j}$, i.e. $\Pi^s_{ID_i,ID_j}$ accepts and there is no oracle $\Pi^t_{ID_j,ID_i}$ matching to $\Pi^s_{ID_i,ID_j}$, the simulator halts and outputs the valid signature $SIG_{ID_j}(ID_j, Y, X, ID_i))$.

CASE 2: Assume that $\Pi^s_{ID_i,ID_j}$ is a responder session.

If $\Pi^s_{ID_i,ID_j}$ is invoked as a responder oracle, then at some time τ_1, $\Pi^s_{ID_i,ID_j}$ receives the first message, say X, from the adversary and returns $(ID_i, Y = g^{H_1(\tilde{y}, sk_{ID_i})}, SIG_{ID_i}(U_i, Y, X, U_j))$ to the adversary. If $\Pi^s_{ID_i,ID_j}$ is to accept, it must later receive a flow of the form $SIG_{ID_j}(ID_j, X, Y, ID_i)$, where Y is produced by the simulator $\bar{F}$. If the adversary M succeeds against the session $\Pi^s_{ID_i,ID_j}$, i.e. $\Pi^s_{ID_i,ID_j}$ accepts and there is no oracle $\Pi^t_{ID_j,ID_i}$ matching to $\Pi^s_{ID_i,ID_j}$, the simulator halts and outputs the valid signature $SIG_{ID_j}(ID_j, X, Y, ID_i)$.

So we have

$$Pr[\textbf{No-Matching}] \leq n \cdot Succ^{eu-cma}_{\Sigma}(\kappa)$$

Game 4: In this game, we add following rule: S randomly chooses a value $s^* \in \{1, ...q_s\}$ as the Test session. Assume that two communicating parties are A and B. We denote by $\Pi^s_{A,B}$ the Test session. S aborts if the Test session does not occur in the session $\Pi^s_{A,B}$. We have

$$\epsilon_3 = q_s \cdot \epsilon_4.$$

Game 5: In the game we replace $H_1(\tilde{x}, sk_A)$ with x whenever it is computed in the Test session $\Pi^s_{A,B}$, where $x \leftarrow_R \mathbb{Z}_q$. Similarly, $H_1(\tilde{y}, sk_B)$ is replaced with y whenever it is computed in the matching session, where $y \leftarrow_R \mathbb{Z}_q$.

Since in this game the adversary is passive, i.e. it can not actively choose the DH values X, Y at a session, and there must be a session matching to the Test session. An active adversary will have caused Game 3 to abort. According to freshness definition 4, in this case the adversary can only reveal one of ephemeral key and static key of both the Test session and its matching session. So the probability of the adversary in Game 4 and Game 5 is identical. Hence

$$Pr[S_6] = Pr[S_5]$$

Game 6: In this game we replace the session key K_A of the Test session with a random string $R \in \{0,1\}^\kappa$. Similarly, the session key K_B of the matching session is replaced with the same random string R.

The adversary can distinguish Game 6 and Game 5 only if it asks random oracle H with input $(\text{CDH}(X, Y), sid_A)$ or $(\text{CDH}(X, Y), sid_B)$. We get

$$|Pr[S_6] - Pr[S_5]| \leq Succ^{cdh}_{\mathbb{G}}(\kappa); \qquad \epsilon_5 \leq 2 \cdot Succ^{cdh}_{\mathbb{G}}(\kappa) + \epsilon_6.$$

Finally, it is easy to see that in Game 6, we have

$$|Pr[S_6]| = \tfrac{1}{2}; \qquad \epsilon_6 = 0.$$

From Game 0 to Game 6, we get the desired result.