Authenticated Key Exchange and Key Encapsulation in the Standard Model

Tatsuaki Okamoto

NTT, Japan
okamoto.tatsuaki@lab.ntt.co.jp

Abstract. This paper introduces a new paradigm to realize various types of cryptographic primitives such as authenticated key exchange and key encapsulation in the standard model under three standard assumptions: the decisional Diffie-Hellman (DDH) assumption, target collision resistant (TCR) hash functions and pseudo-random functions (PRFs). We propose the first (PKI-based) two-pass authenticated key exchange (AKE) protocol that is comparably as efficient as the existing most efficient protocols like MQV and that is secure in the standard model (under these standard assumptions), while the existing efficient two-pass AKE protocols such as HMQV, NAXOS and CMQV are secure in the random oracle model. Our protocol is shown to be secure in the (currently) strongest security definition, the extended Canetti-Krawczyk (eCK) security definition introduced by LaMacchia, Lauter and Mityagin. This paper also proposes a CCA-secure key encapsulation mechanism (KEM) under these assumptions, which is almost as efficient as the Kurosawa-Desmedt KEM. This scheme is also secure in a stronger security notion, the chosen public-key and ciphertext attack (CPCA) security. The proposed schemes in this paper are redundancy-free (or validity-check-free) and the implication is that combining them with redundancy-free symmetric encryption (DEM) will yield redundancy-free (e.g., MAC-free) CCA-secure hybrid encryption.

1 Introduction

The most common paradigm to design practical public-key cryptosystems secure in the standard model is to combine a trapdoor function (e.g., Diffie-Hellman or RSA function) and target collision resistance (TCR) hash functions, where the security is proven under a trapdoor function assumption (e.g., DDH or SRSA assumption) and the TCR hash function assumption.

This paper introduces a new paradigm to design practical public-key cryptosystems, where a *pseudo-random function* (PRF) is employed in addition to a trapdoor function (DH) and target collision resistant (TCR) hash function.

The concept of a PRF was introduced by Goldreich, Goldwasser and Micali [4], and has been shown to exist if and only if a one-way function exists [4, 5]. Therefore, the existence of a pseudo-random function is one of the weakest assumptions, and it is one of the most fundamental primitives in cryptography.

Since a target collision resistant (TCR) hash function (and the slightly bit more general concept, the universal one-way hash function) have also been shown to exist if and only if a one-way function exists [12, 13], TCR hash function and PRF are the same level of (the most) fundamental primitives in cryptography. In practice, a well-designed efficient hash function can be assumed to be a TCR hash function, and such a hash function with a random seed as a part of the input (or a keyed hash function) can be assumed to be a PRF.

First, this paper presents a two-pass AKE protocol that offers the following properties:

1. its efficiency is comparable to those of MQV [9], HMQV [6] and CMQV [14] (the message size of our scheme is that of MQV plus the size of two group elements, and the computational complexity for a session of our scheme is around 3.3 group exponentiations, while that of MQV is around 2.2 group exponentiations),
2. the assumption and model for its security proof are standard assumptions (DDH, TCR hash function and PRF) and standard model (not the random oracle model),
3. its underlying security definition is (currently) the strongest one, the extended Canetti-Krawczyk (eCK) security definition introduced by LaMacchia, Lauter and Mityagin [8],
4. its security proof reduction efficiency is better than those of previous protocols in the random oracle model.

This paper also proposes a *CCA-secure* key encapsulation mechanism (KEM) under these assumptions, which is almost as efficient as the Kurosawa-Desmedt KEM [7]. This scheme is also secure in a stronger security notion, the *chosen public-key and ciphertext attack (CPCA)* security, in which an adversary, given a target public key pk^* and ciphertext c^*, is allowed to query a pair of public key pk and ciphertext c to the decryption oracle, which answers the adversary with the decrypted result of c by the secret key of pk.

The proposed schemes in this paper are redundancy-free (or validity-check-free) and implies redundancy-free (e.g., MAC-free) CCA-secure hybrid encryption by combining with redundancy-free CCA-secure symmetric encryption (DEM).

2 Preliminaries

2.1 Notations

$\mathbb{N}$ is the set of natural numbers and $\overline{\mathbb{R}}$ is the set of real numbers. $\perp$ denotes a null string.

A function $f : \mathbb{N} \rightarrow \overline{\mathbb{R}}$ is *negligible* in k, if for every constant $c > 0$, there exists integer n such that $f(k) < k^{-c}$ for all $k > n$. Hereafter, we often use $f(k) < \epsilon(k)$ to mean that f is negligible in k.

When A is a probabilistic machine or algorithm, $A(x)$ denotes the random variable of A's output on input x. Then, $y \overset{\mathsf{R}}{\leftarrow} A(x)$ denotes that y is randomly selected from $A(x)$ according to its distribution. When a is a value, $A(x) \rightarrow a$ denotes the event that A outputs a on input x. When A is a set, $y \overset{\mathsf{U}}{\leftarrow} A$ denotes that y is uniformly selected from A. When A is a value, $y \leftarrow A$ denotes that y is set as A.

In this paper, we consider that the underlying machines are uniform Turing machines. But it is easy to extend our results to non-uniform Turing machines.

2.2 The DDH Assumption

Let k be a security parameter and $\mathbb{G}$ be a group with security parameter k, where the order of $\mathbb{G}$ is prime p and $|p| = k$. Let $\{\mathbb{G}\}_k$ be the set of group $\mathbb{G}$ with security parameter k.

For all $k \in \mathbb{N}$ we define the sets $\mathbb{D}$ and $\mathbb{R}$ as follows:

$$\mathbb{D}(k) \leftarrow \{(\mathbb{G}, g_1, g_2, g_1^x, g_2^x) \mid \mathbb{G} \overset{\mathsf{U}}{\leftarrow} \{\mathbb{G}\}_k, (g_1, g_2) \overset{\mathsf{U}}{\leftarrow} \mathbb{G}^2, x \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p\}$$
$$\mathbb{R}(k) \leftarrow \{(\mathbb{G}, g_1, g_2, y_1, y_2) \mid \mathbb{G} \overset{\mathsf{U}}{\leftarrow} \{\mathbb{G}\}_k, (g_1, g_2, y_1, y_2) \overset{\mathsf{U}}{\leftarrow} \mathbb{G}^4\}.$$

Let $\mathcal{A}$ be a probabilistic polynomial-time machine. For all $k \in \mathbb{N}$, we define the DDH advantage of $\mathcal{A}$ as

$$\mathsf{Adv}\mathrm{DDH}_{\mathcal{A}}(k) \leftarrow |\ \Pr[\mathcal{A}(1^k, \rho) \rightarrow 1 \mid \rho \overset{\mathsf{U}}{\leftarrow} \mathbb{D}(k)] - \Pr[\mathcal{A}(1^k, \rho) \rightarrow 1 \mid \rho \overset{\mathsf{U}}{\leftarrow} \mathbb{R}(k)]\ |.$$

The DDH assumption for $\{\mathbb{G}\}_{k\in\mathbb{N}}$ is: For any probabilistic polynomial-time adversary $\mathcal{A}$, $\mathsf{Adv}\mathrm{DDH}_{\mathcal{A}}(k)$ is negligible in k.

2.3 Pseudo-Random Function (PRF)

Let $k \in \mathbb{N}$ be a security parameter. A pseudo-random function (PRF) family F associated with $\{\mathsf{Seed}_k\}_{k\in\mathbb{N}}$, $\{\mathsf{Dom}_k\}_{k\in\mathbb{N}}$ and $\{\mathsf{Rng}_k\}_{k\in\mathbb{N}}$ specifies two items:

- A family of random seeds $\{\mathsf{Seed}_k\}_{k\in\mathbb{N}}$.
- A family of pseudo-random functions indexed by k, $\Sigma \overset{\mathsf{R}}{\leftarrow} \mathsf{Seed}_k$, $\sigma \overset{\mathsf{U}}{\leftarrow} \Sigma$, $\mathcal{D} \overset{\mathsf{R}}{\leftarrow} \mathsf{Dom}_k$, and $\mathcal{R} \overset{\mathsf{R}}{\leftarrow} \mathsf{Rng}_k$, where each such function $\mathsf{F}_\sigma^{k,\Sigma,\mathcal{D},\mathcal{R}}$ maps an element of $\mathcal{D}$ to an element of $\mathcal{R}$. There must exist a deterministic polynomial-time algorithm that on input 1^k, σ and ρ, outputs $\mathsf{F}_\sigma^{k,\Sigma,\mathcal{D},\mathcal{R}}(\rho)$.

Let $\mathcal{A}^O$ be a probabilistic polynomial-time machine with oracle access to O. For all k, we define

$$\mathsf{Adv}\mathrm{PRF}_{\mathsf{F},\mathcal{A}}(k) \leftarrow |\Pr[\mathcal{A}^F(1^k, \mathcal{D}, \mathcal{R}) \rightarrow 1] - \Pr[\mathcal{A}^{RF}(1^k, \mathcal{D}, \mathcal{R}) \rightarrow 1]|,$$

where $\Sigma \overset{\mathsf{R}}{\leftarrow} \mathsf{Seed}_k$, $\sigma \overset{\mathsf{U}}{\leftarrow} \Sigma$, $\mathcal{D} \overset{\mathsf{R}}{\leftarrow} \mathsf{Dom}_k$, $\mathcal{R} \overset{\mathsf{R}}{\leftarrow} \mathsf{Rng}_k$, $F \leftarrow \mathsf{F}_\sigma^{k,\Sigma,\mathcal{D},\mathcal{R}}$, and $RF : \mathcal{D} \rightarrow \mathcal{R}$ is a truly random function ($\forall \rho \in \mathcal{D} \quad RF(\rho) \overset{\mathsf{U}}{\leftarrow} \mathcal{R}$).

F is a pseudo-random function (PRF) family if for any probabilistic polynomial-time adversary $\mathcal{A}$, $\mathsf{Adv}\mathrm{PRF}_{\mathsf{F},\mathcal{A}}(k)$ is negligible in k.

2.4 Target Collision Resistant (TCR) Hash Function

Let $k \in \mathbb{N}$ be a security parameter. A target collision resistant (TCR) hash function family H associated with $\{\mathsf{Dom}_k\}_{k\in\mathbb{N}}$ and $\{\mathsf{Rng}_k\}_{k\in\mathbb{N}}$ specifies two items:

- A family of key spaces indexed by k. Each such key space is a probability space on bit strings denoted by KH_k. There must exist a probabilistic polynomial-time algorithm whose output distribution on input 1^k is equal to KH_k.

– A family of hash functions indexed by k, $h \xleftarrow{\mathsf{R}} \mathsf{KH}_k$, $\mathcal{D} \xleftarrow{\mathsf{R}} \mathsf{Dom}_k$, and $\mathcal{R} \xleftarrow{\mathsf{R}} \mathsf{Rng}_k$, where each such function $\mathsf{H}_h^{k,\mathcal{D},\mathcal{R}}$ maps an element of $\mathcal{D}$ to an element of $\mathcal{R}$. There must exist a deterministic polynomial-time algorithm that on input 1^k, h and ρ, outputs $\mathsf{H}_h^{k,\mathcal{D},\mathcal{R}}(\rho)$.

Let $\mathcal{A}$ be a probabilistic polynomial-time machine. For all k, we define

$$\mathsf{Adv}\mathrm{TCR}_{\mathsf{H},\mathcal{A}}(k) \leftarrow$$
$$\Pr[\rho \in \mathcal{D} \wedge \rho \neq \rho^* \wedge \mathsf{H}_h^{k,\mathcal{D},\mathcal{R}}(\rho) = \mathsf{H}_h^{k,\mathcal{D},\mathcal{R}}(\rho^*) \mid \rho \xleftarrow{\mathsf{R}} \mathcal{A}(1^k, \rho^*, h, \mathcal{D}, \mathcal{R})],$$

where $\mathcal{D} \xleftarrow{\mathsf{R}} \mathsf{Dom}_k, \mathcal{R} \xleftarrow{\mathsf{R}} \mathsf{Rng}_k, \rho^* \xleftarrow{\mathsf{U}} \mathcal{D}$ and $h \xleftarrow{\mathsf{R}} \mathsf{KH}_k$. H is a target collision resistance (TCR) hash function family if for any probabilistic polynomial-time adversary $\mathcal{A}$, $\mathsf{Adv}\mathrm{TCR}_{\mathsf{H},\mathcal{A}}(k)$ is negligible in k.

2.5 PKI-Based Authenticated Key Exchange (AKE) and the Extended Canetti-Krawczyk (eCK) Security Definition

This section outlines the extended Canetti-Krawczyk (eCK) security definition for two pass PKI-based authenticated key exchange (AKE) protocols that was introduced by LaMacchia, Lauter and Mityagin [8], and follows the description in [14].

In the eCK definition, we suppose there are n parties which are modeled as probabilistic polynomial-time Turing machines. We assume that some agreement on the common parameters in the AKE protocol has been made among the parties before starting the protocol. The mechanism by which these parameters are selected is out of scope of the AKE protocol and the (eCK) security model.

Each party has a static public-private key pair together with a certificate that binds the public key to that party. $\hat{A}$ ($\hat{B}$) denotes the static public key A (B) of party $\mathcal{A}$ ($\mathcal{B}$) together with a certificate. We do not assume that the certifying authority (CA) requires parties to prove possession of their static private keys, but we require that the CA verifies that the static public key of a party belongs to the domain of public keys.

Here, two parties exchange static public keys A, B and ephemeral public keys X, Y; the session key is obtained by combining A, B, X, Y and possibly session identities. A party $\mathcal{A}$ can be activated to execute an instance of the protocol called a *session*. Activation is made via an incoming message that has one of the following forms: $(\hat{A}, \hat{B})$ or $(\hat{B}, \hat{A}, X)$. If $\mathcal{A}$ was activated with $(\hat{A}, \hat{B})$, then $\mathcal{A}$ is called the session initiator, otherwise the session responder. Session initiator $\mathcal{A}$ creates ephemeral public-private key pair, (X, x) and sends $(\hat{B}, \hat{A}, X)$ to session responder $\mathcal{B}$. $\mathcal{B}$ then creates ephemeral public-private key pair, (Y, y) and sends $(\hat{A}, \hat{B}, X, Y)$ to $\mathcal{A}$.

The session of initiator $\mathcal{A}$ with responder $\mathcal{B}$ is identified via session identifier $(\hat{A}, \hat{B}, X, Y)$, where $\mathcal{A}$ is said the owner of the session, and $\mathcal{B}$ the peer of the session. The session of responder $\mathcal{B}$ with initiator $\mathcal{A}$ is identified as $(\hat{B}, \hat{A}, Y, X)$, where $\mathcal{B}$ is the owner, and $\mathcal{A}$ is the peer. Session $(\hat{B}, \hat{A}, Y, X)$ is said a matching session of $(\hat{A}, \hat{B}, X, Y)$. We say that a session is completed if its owner computes a session key.

The adversary $\mathcal{M}$ is modeled as a probabilistic polynomial-time Turing machine and controls all communications. Parties submit outgoing messages to the adversary, who makes decisions about their delivery. The adversary presents parties with incoming messages via $\mathsf{Send}(message)$, thereby controlling the activation of sessions. In order to capture possible leakage of private information, adversary $\mathcal{M}$ is allowed the following queries:

- $\mathsf{EphemeralKeyReveal}(\mathsf{sid})$ The adversary obtains the ephemeral private key associated with session sid.
- $\mathsf{SessionKeyReveal}(\mathsf{sid})$ The adversary obtains the session key for session sid, provided that the session holds a session key.
- $\mathsf{StaticKeyReveal}(\mathsf{pid})$ The adversary learns the static private key of party pid.
- $\mathsf{EstablishParty}(\mathsf{pid})$ This query allows the adversary to register a static public key on behalf of a party. In this way the adversary totally controls that party.

If a party pid is established by $\mathsf{EstablishParty}(\mathsf{pid})$ query issued by adversary $\mathcal{M}$, then we call the party *dishonest*. If a party is not dishonest, we call the party *honest*.

The aim of adversary $\mathcal{M}$ is to distinguish a session key from a random key. Formally, the adversary is allowed to make a special query $\mathsf{Test}(\mathsf{sid}^*)$, where sid^* is called the *target session*. The adversary is then given with equal probability either the session key, K^*, held by sid^* or a random key, $R^* \stackrel{\mathsf{U}}{\leftarrow} \{0,1\}^{|K^*|}$. The adversary wins the game if he guesses correctly whether the key is random or not. To define the game, we need the notion of *fresh session* as follows:

Definition 1. *(fresh session) Let* sid *be the session identifier of a completed session, owned by an honest party* $\mathcal{A}$ *with peer* $\mathcal{B}$*, who is also honest. Let* $\overline{\mathsf{sid}}$ *be the session identifier of the matching session of* sid*, if it exists. Define session* sid *to be "fresh" if none of the following conditions hold:*

- $\mathcal{M}$ *issues a* $\mathsf{SessionKeyReveal}(\mathsf{sid})$ *query or a* $\mathsf{SessionKeyReveal}(\overline{\mathsf{sid}})$ *query (if* $\overline{\mathsf{sid}}$ *exists),*
- $\overline{\mathsf{sid}}$ *exists and* $\mathcal{M}$ *makes either of the following queries:*
  both $\mathsf{StaticKeyReveal}(\mathcal{A})$ *and* $\mathsf{EphemeralKeyReveal}(\mathsf{sid})$*, or*
  both $\mathsf{StaticKeyReveal}(\mathcal{B})$ *and* $\mathsf{EphemeralKeyReveal}(\overline{\mathsf{sid}})$*,*
- $\overline{\mathsf{sid}}$ *does not exist and* $\mathcal{M}$ *makes either of the following queries:*
  both $\mathsf{StaticKeyReveal}(\mathcal{A})$ *and* $\mathsf{EphemeralKeyReveal}(\mathsf{sid})$*, or*
  $\mathsf{StaticKeyReveal}(\mathcal{B})$*.*

We are now ready to present the eCK security notion.

Definition 2. *(eCK security) Let* K^* *be a session key of the target session* sid^* *that should be "fresh",* $R^* \stackrel{\mathsf{U}}{\leftarrow} \{0,1\}^{|K^*|}$*, and* $b^* \stackrel{\mathsf{U}}{\leftarrow} \{0,1\}$*. As a reply to* $\mathsf{Test}(\mathsf{sid}^*)$ *query by* $\mathcal{M}$*,* K^* *is given to* $\mathcal{M}$ *if* $b^* = 0$*;* R^* *is given otherwise. Finally* $\mathcal{M}$ *outputs* $b \in \{0,1\}$*. We define*

$$\mathsf{AdvAKE}_{\mathcal{M}}(k) \leftarrow |\Pr[b = b^*] - 1/2|.$$

A key exchange protocol is secure if the following conditions hold:

- *If two honest parties complete matching sessions, then they both compute the same session key (or both output indication of protocol failure).*

– *For any probabilistic polynomial-time adversary $\mathcal{M}$, $\mathsf{Adv}\mathrm{AKE}_{\mathcal{M}}(k)$ is negligible in k.*

This security definition is stronger than CK-security [2] and it simultaneously captures all the known desirable security properties for authenticated key exchange including resistance to key-compromise impersonation attacks, weak perfect forward secrecy, and resilience to the leakage of ephemeral private keys.

2.6 Key-Encapsulation Mechanism (KEM)

A key encapsulation mechanism (KEM) scheme is the triple of algorithms, $\Sigma = (\mathsf{K}, \mathsf{E}, \mathsf{D})$, where

1. K, the key generation algorithm, is a probabilistic polynomial time (PPT) algorithm that takes a security parameter $k \in \mathbb{N}$ (provided in unary) and returns a pair (pk, sk) of matching public and secret keys.
2. E, the key encryption algorithm, is a PPT algorithm that takes as input public key pk and outputs a key/ciphertext pair (K^*, C^*).
3. D, the decryption algorithm, is a deterministic polynomial time algorithm that takes as input secret key sk and ciphertext C^*, and outputs key K^* or $\perp$ ($\perp$ means that the ciphertext is invalid).

We require that for all (pk, sk) output by key generation algorithm K and for all (K^*, C^*) output by key encryption algorithm $\mathsf{E}(pk)$, $\mathsf{D}(sk, C^*) = K^*$ holds. Here, the length of the key, $|K^*|$, is specified by $l(k)$, where k is the security parameter.

Let $\mathcal{A}$ be an adversary. The attack game is defined in terms of an interactive computation between adversary $\mathcal{A}$ and its challenger, $\mathcal{C}$. The challenger $\mathcal{C}$ responds to the oracle queries made by $\mathcal{A}$. We now describe the attack game (IND-CCA2 game) used to define security against adaptive chosen ciphertext attacks (IND-CCA2).

1. The challenger $\mathcal{C}$ generates a pair of keys, $(pk, sk) \stackrel{\mathsf{R}}{\leftarrow} \mathsf{K}(1^k)$ and gives pk to adversary $\mathcal{A}$.
2. Repeat the following procedure $q_1(k)$ times, for $i = 1, \ldots, q_1(k)$, where $q_1(\cdot)$ is a polynomial. $\mathcal{A}$ submits string C_i to a decryption oracle, DO (in $\mathcal{C}$), and DO returns $\mathsf{D}_{sk}(C_i)$ to $\mathcal{A}$.
3. $\mathcal{A}$ submits the encryption query to $\mathcal{C}$. The encryption oracle, EO, in $\mathcal{C}$ selects $b^* \stackrel{\mathsf{U}}{\leftarrow} \{0, 1\}$ and computes $(C^*, K^*) \leftarrow \mathsf{E}(pk)$ and returns (C^*, K^*) to $\mathcal{A}$ if $b^* = 0$ and (C^*, R^*) if $b^* = 1$, where $R^* \stackrel{\mathsf{U}}{\leftarrow} \{0, 1\}^{|K^*|}$ (C^* is called "target ciphertext").
4. Repeat the following procedure $q_2(k)$ times, for $j = q_1(k) + 1, \ldots, q_1(k) + q_2(k)$, where $q_2(\cdot)$ is a polynomial. $\mathcal{A}$ submits string C_j to a decryption oracle, DO (in $\mathcal{C}$), subject only to the restriction that a submitted text C_j is not identical to C^*. DO returns $\mathsf{D}_{sk}(C_j)$ to $\mathcal{A}$.
5. $\mathcal{A}$ outputs $b \in \{0, 1\}$.

We define the IND-CCA2 advantage of $\mathcal{A}$, $\mathsf{Adv}\mathrm{KEM}_{\mathcal{A}}^{\text{IND-CCA2}}(k) \leftarrow |\Pr[b = b^*] - 1/2|$ in the above attack game.

We say that a KEM scheme is IND-CCA2-secure (secure against adaptive chosen ciphertext attacks) if for any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, $\mathsf{Adv}\mathrm{KEM}_{\mathcal{A}}^{\text{IND-CCA2}}(k)$ is negligible in k.

3 The Proposed AKE Protocol

3.1 Protocol

Let $k \in \mathbb{N}$ be a security parameter, $\mathbb{G} \xleftarrow{\mathsf{U}} \{\mathbb{G}\}_k$ be a group with security parameter k, and $(g_1, g_2) \xleftarrow{\mathsf{U}} \mathbb{G}^2$, where the order of $\mathbb{G}$ is prime p and $|p| = k$. Let H be a TCR hash function family, and F, $\tilde{\mathsf{F}}$ and $\hat{\mathsf{F}}$ be PRF families. $(\mathbb{G}, g_1, g_2)$, H, F, $\tilde{\mathsf{F}}$ and $\hat{\mathsf{F}}$ are the system parameters common among all users of the proposed AKE protocol (although $\tilde{\mathsf{F}}$ and $\hat{\mathsf{F}}$ can be set privately by each party) We assume that the systems parameters are selected by a trusted third party.

Party $\mathcal{A}$'s static private key is $(a_1, a_2, a_3, a_4) \xleftarrow{\mathsf{U}} (\mathbb{Z}_p)^4$ and $\mathcal{A}$'s static public key is $A_1 \leftarrow g_1^{a_1} g_2^{a_2}$, $A_2 \leftarrow g_1^{a_3} g_2^{a_4}$. $h_A \xleftarrow{\mathsf{R}} \mathsf{KH}_k$ indexes a TCR hash function $H_A \leftarrow \mathsf{H}_{h_A}^{k,\mathcal{D}_H,\mathcal{R}_H}$, where $\mathcal{D}_H \leftarrow \Pi_k \times \mathbb{G}^4$, $\mathcal{R}_H \leftarrow \mathbb{Z}_p$ and Π_k denotes the space of possible certificates for static public keys.

Similarly, Party $\mathcal{B}$'s static private key is $(b_1, b_2, b_3, b_4) \xleftarrow{\mathsf{U}} (\mathbb{Z}_p)^4$ and $\mathcal{B}$'s static public key is $B_1 \leftarrow g_1^{b_1} g_2^{b_2}$, $B_2 \leftarrow g_1^{b_3} g_2^{b_4}$. $h_B \xleftarrow{\mathsf{R}} \mathsf{KH}_k$ indexes a TCR hash function $H_B \leftarrow \mathsf{H}_{h_B}^{k,\mathcal{D}_H,\mathcal{R}_H}$.

$\mathcal{A}$ and $\mathcal{B}$ set PRFs $F \leftarrow \mathsf{F}^{k,\Sigma_{\mathsf{F}},\mathcal{D}_{\mathsf{F}},\mathcal{R}_{\mathsf{F}}}$, $\tilde{F} \leftarrow \tilde{\mathsf{F}}^{k,\Sigma_{\tilde{\mathsf{F}}},\mathcal{D}_{\tilde{\mathsf{F}}},\mathcal{R}_{\tilde{\mathsf{F}}}}$ and $\hat{F} \leftarrow \hat{\mathsf{F}}^{k,\Sigma_{\hat{\mathsf{F}}},\mathcal{D}_{\hat{\mathsf{F}}},\mathcal{R}_{\hat{\mathsf{F}}}}$, where $\Sigma_{\mathsf{F}} \leftarrow \mathbb{G}$, $\mathcal{D}_{\mathsf{F}} \leftarrow (\Pi_k)^2 \times \mathbb{G}^8$, $\mathcal{R}_{\mathsf{F}} \leftarrow \{0,1\}^k$, $\Sigma_{\tilde{\mathsf{F}}} \leftarrow (\mathbb{Z}_p)^4$, $\mathcal{D}_{\tilde{\mathsf{F}}} \leftarrow \{0,1\}^k$, $\mathcal{R}_{\tilde{\mathsf{F}}} \leftarrow \mathbb{Z}_p$, $\Sigma_{\hat{\mathsf{F}}} \leftarrow \{0,1\}^k$, $\mathcal{D}_{\hat{\mathsf{F}}} \leftarrow (\mathbb{Z}_p)^4$, and $\mathcal{R}_{\hat{\mathsf{F}}} \leftarrow \mathbb{Z}_p$.

To establish a session key with party $\mathcal{B}$, party $\mathcal{A}$ performs the following procedure.

1. Select an ephemeral private key $\tilde{x} \xleftarrow{\mathsf{U}} \{0,1\}^k$.
2. Compute $x \leftarrow \hat{F}_{\tilde{x}}(a_1, a_2, a_3, a_4) + \tilde{F}_{(a_1,a_2,a_3,a_4)}(\tilde{x}) \bmod p$ and the ephemeral public key $(X_1 \leftarrow g_1^x, X_2 \leftarrow g_2^x)$.
3. Erase x.
4. Send $(\hat{B}, \hat{A}, X_1, X_2)$ to $\mathcal{B}$.

Upon receiving $(\hat{B}, \hat{A}, X_1, X_2)$, party $\mathcal{B}$ verifies that $(X_1, X_2) \in \mathbb{G}^2$. If so, perform the following procedure.

1. Select an ephemeral private key $\tilde{y} \xleftarrow{\mathsf{U}} \{0,1\}^k$.
2. Compute $y \leftarrow \hat{F}_{\tilde{y}}(b_1, b_2, b_3, b_4) + \tilde{F}_{(b_1,b_2,b_3,b_4)}(\tilde{y}) \bmod p$ and the ephemeral public key $(Y_1 \leftarrow g_1^y, Y_2 \leftarrow g_2^y)$.
3. Erase y.
4. Send $(\hat{A}, \hat{B}, X_1, X_2, Y_1, Y_2)$ to $\mathcal{A}$.

Upon receiving $(\hat{A}, \hat{B}, X_1, X_2, Y_1, Y_2)$, party $\mathcal{A}$ checks if he sent $(\hat{B}, \hat{A}, X_1, X_2)$ to $\mathcal{B}$. If so, $\mathcal{A}$ verifies that $(Y_1, Y_2) \in \mathbb{G}^2$.

To compute the session key, $\mathcal{A}$ computes $\sigma_A \leftarrow Y_1^{a_1+ca_3+x} Y_2^{a_2+ca_4+x} B_1^x B_2^{dx}$, and $\mathcal{B}$ computes $\sigma_B \leftarrow X_1^{b_1+db_3+y} X_2^{b_2+db_4+y} A_1^y A_2^{cy}$, where $c \leftarrow H_A(\hat{A}, Y_1, Y_2)$ and $d \leftarrow H_B(\hat{B}, X_1, X_2)$. If they are correctly computed, $\sigma \leftarrow \sigma_A (= \sigma_B)$. The session key is $K \leftarrow F_\sigma(\mathsf{sid})$, where $\mathsf{sid} \leftarrow (\hat{A}, \hat{B}, X_1, X_2, Y_1, Y_2)$.

$\mathcal{A}$		$\mathcal{B}$
$(a_1, a_2, a_3, a_4) \overset{\mathsf{U}}{\leftarrow} (\mathbb{Z}_p)^4$		$(b_1, b_2, b_3, b_4) \overset{\mathsf{U}}{\leftarrow} (\mathbb{Z}_p)^4$
$A_1 \leftarrow g_1^{a_1} g_2^{a_2}, A_2 \leftarrow g_1^{a_3} g_2^{a_4},$		$B_1 \leftarrow g_1^{b_1} g_2^{b_2}, B_2 \leftarrow g_1^{b_3} g_2^{b_4},$
h_A		h_B
$\tilde{x} \overset{\mathsf{U}}{\leftarrow} \{0,1\}^k$		
$x \leftarrow \hat{F}_{\tilde{x}}(a_1, a_2, a_3, a_4)$ $+\tilde{F}_{(a_1,a_2,a_3,a_4)}(\tilde{x}) \bmod p$		
$X_1 \leftarrow g_1^x, X_2 \leftarrow g_2^x$	$\xrightarrow{(\hat{B}, \hat{A}, X_1, X_2)}$	
		$(X_1, X_2) \in \mathbb{G}^2?$
		$\tilde{y} \overset{\mathsf{U}}{\leftarrow} \{0,1\}^k$
		$y \leftarrow \hat{F}_{\tilde{y}}(b_1, b_2, b_3, b_4)$ $+\tilde{F}_{(b_1,b_2,b_3,b_4)}(\tilde{y}) \bmod p$
	$\xleftarrow{(\hat{A}, \hat{B}, X_1, X_2, Y_1, Y_2)}$	$Y_1 \leftarrow g_1^y, Y_2 \leftarrow g_2^y$
$(Y_1, Y_2) \in \mathbb{G}^2?$		
$c \leftarrow H_A(\hat{A}, Y_1, Y_2)$		$c \leftarrow H_A(\hat{A}, Y_1, Y_2)$
$d \leftarrow H_B(\hat{B}, X_1, X_2)$		$d \leftarrow H_B(\hat{B}, X_1, X_2)$
$\sigma \leftarrow Y_1^{a_1+ca_3+x} Y_2^{a_2+ca_4+x} \cdot$ $B_1^x B_2^{dx}$		$\sigma \leftarrow X_1^{b_1+db_3+y} X_2^{b_2+db_4+y} \cdot$ $A_1^y A_2^{cy}$
$K \leftarrow F_\sigma(\mathsf{sid})$		$K \leftarrow F_\sigma(\mathsf{sid})$

Here, $\mathsf{sid} \leftarrow (\hat{A}, \hat{B}, X_1, X_2, Y_1, Y_2)$. Note that $(A_1, A_2, B_1, B_2) \in \mathbb{G}^4$ is confirmed indirectly through the certificates.

Fig. 1. The Proposed AKE

3.2 Security

Theorem 1. *The proposed AKE protocol is secure (in the sense of Definition 2) if the DDH assumption holds for* $\{\mathbb{G}\}_{k\in\mathbb{N}}$, H *is a TCR hash function family, and* F, $\tilde{\mathsf{F}}$ *and* $\hat{\mathsf{F}}$ *are PRF families.*

The proof will be given in the full paper version of this paper.

4 The Proposed KEM Scheme

4.1 Scheme

In this section, we show a CCA secure KEM scheme.

Let $k \in \mathbb{N}$ be a security parameter, and let $\mathbb{G} \overset{\mathsf{U}}{\leftarrow} \{\mathbb{G}\}_k$ be a group with security parameter k, where the order of $\mathbb{G}$ is prime p and $|p| = k$.

Let H be a TCR hash function family, and F be a PRF family.

Secret Key: The secret key is $sk \leftarrow (x_1, x_2, y_1, y_2) \xleftarrow{\mathsf{U}} \mathbb{Z}_p^4$.

Public Key: $g_1 \xleftarrow{\mathsf{U}} \mathbb{G}$, $g_2 \xleftarrow{\mathsf{U}} \mathbb{G}$, $z \leftarrow g_1^{x_1} g_2^{x_2}$, $w \leftarrow g_1^{y_1} g_2^{y_2}$, $H \leftarrow \mathsf{H}_h^{k,\mathcal{D}_H,\mathcal{R}_H}$ and $F \leftarrow \mathsf{F}^{k,\Sigma_\mathsf{F},\mathcal{D}_\mathsf{F},\mathcal{R}_\mathsf{F}}$, where $h \xleftarrow{\mathsf{R}} \mathsf{KH}_k$, $\mathcal{D}_H \leftarrow \{pk\} \times \mathbb{G}^2$ (pk is a possible public-key value), $\mathcal{R}_H \leftarrow \mathbb{Z}_p$, $\Sigma_\mathsf{F} \leftarrow \mathbb{G}$, $\mathcal{D}_\mathsf{F} \leftarrow \{pk\} \times \mathbb{G}^2$ and $\mathcal{R}_\mathsf{F} \leftarrow \{0,1\}^k$.
The public key is $pk \leftarrow (\mathbb{G}, g_1, g_2, z, w, H, F)$.

Encryption: Choose $r \xleftarrow{\mathsf{U}} \mathbb{Z}_p$ and compute

$$\begin{aligned}
&C_1 \leftarrow g_1^r,\\
&C_2 \leftarrow g_2^r,\\
&d \leftarrow H(z, w, C_1, C_2)\\
&\sigma \leftarrow z^r w^{rd}\\
&K \leftarrow F_\sigma(pk, C_1, C_2).
\end{aligned}$$

(C_1, C_2) is a ciphertext, and K is the secret key to be shared.

Decryption: Given (z, w, C_1, C_2), check whether

$$(z, w, C_1, C_2) \in \mathbb{G}^4.$$

If it holds, computes

$$\begin{aligned}
&d \leftarrow H(z, w, C_1, C_2)\\
&\sigma \leftarrow C_1^{x_1+dy_1} C_2^{x_2+dy_2}\\
&K \leftarrow F_\sigma(pk, C_1, C_2).
\end{aligned}$$

4.2 CCA Security

Theorem 2. *The proposed KEM scheme is IND-CCA2 secure if the DDH assumption holds for $\{\mathbb{G}\}_{k\in\mathbb{N}}$, H is a TCR hash function family, and F is a PRF family.*

The proof will be given in the full paper version of this paper.

4.3 CPCA Security

In this paper, we define a stronger security notion than the CCA security on KEM and PKE.

Here, we consider a trapdoor commitment, where committer (sender) $\mathcal{S}$ commits to x by sending $C \leftarrow \mathsf{E}_{pk}(x)$ to receiver $\mathcal{R}$, then $\mathcal{S}$ opens x by sending sk to $\mathcal{R}$, where (pk, sk) is a pair of public key and secret key, and $x = \mathsf{D}_{sk}(C)$. Using a trapdoor commitment, several committers, $\mathcal{S}_1, \ldots, \mathcal{S}_n$, commits to $x_1, \ldots, x_n$ respectively by sending $C_1 \leftarrow \mathsf{E}_{pk}(x_1), \ldots, C_n \leftarrow \mathsf{E}_{pk}(x_n)$ to receiver $\mathcal{R}$. Another party can open them simultaneously by sending sk to receiver $\mathcal{R}$. A possible malleable attack is as follows: after looking at pk and $C \leftarrow \mathsf{E}_{pk}(x)$ sent to receiver $\mathcal{R}$, adversary $\mathcal{A}$ computes

pk', C', algorithm Conv and non-trivial relation Rel. $\mathcal{A}$ registers pk' and sends C' to $\mathcal{R}$ as a commitment to x' such that $\mathsf{Rel}(x, x')$. When sk is opened, $\mathcal{A}$ computes $sk' \leftarrow \mathsf{Conv}(sk)$ and sends sk' to $\mathcal{R}$ such that $x' = \mathsf{D}_{sk'}(C')$.

To capture the security against such malleable attacks, we now define the CPCA (Chosen Public-key and Ciphertext Attacks) security for KEM schemes.

Let $\Sigma = (\mathsf{K}, \mathsf{E}, \mathsf{D})$ be a KEM scheme. Let C^*, pk^* and sk^* be the target ciphertext, public key and secret key of KEM scheme Σ. In the CPCA security, an adversary $\mathcal{A}$, given pk^* and C^*, is allowed to submit a pair of a public key pk and a ciphertext C along with a polynomial-time algorithm Conv to the decryption oracle DO (with sk^*) under the condition that $(pk, C) \neq (pk^*, C^*)$. DO returns $\mathsf{D}_{sk}(C)$ to $\mathcal{A}$, where DO computes and confirms that $sk \leftarrow \mathsf{Conv}(sk^*, pk^*)$, $(c, k) \leftarrow \mathsf{E}_{pk}(1^k)$ and $k \leftarrow \mathsf{D}_{sk}(c)$. (Here, D_{sk} is equivalent to D_{sk^*} except for the difference of sk and sk^*.)

We can define the advantage of $\mathcal{A}$ for the IND-CPCA game, $\mathsf{Adv}\mathrm{KEM}_{\mathcal{A}}^{\text{IND-CPCA}}(k)$. We say that a KEM scheme is IND-CPCA-secure if for any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, $\mathsf{Adv}\mathrm{KEM}_{\mathcal{A}}^{\text{IND-CPCA}}(k)$ is negligible in k.

We now show that the proposed KEM scheme is CPCA secure. To prove the security, we need a new requirement for a hash function family, the generalized TCR (GTCR) hash function family.

Let $k \in \mathbb{N}$ be a security parameter. Let $\mathbb{G}$ be a group with security parameter k, where the order of $\mathbb{G}$ is prime p and $|p| = k$, and $\{\mathbb{G}\}_k$ be the set of group $\mathbb{G}$ with security parameter k.

Let H be a TCR hash function family associated with $\mathsf{Dom}_k \leftarrow \{\mathbb{G}^4\}_k$, $\mathsf{Rng}_k \leftarrow \{\mathbb{G}\}_k$.

For all k, we define

$$\begin{aligned}
&\mathsf{Adv}\mathrm{GTCR}_{\mathsf{H},\mathcal{A}}^{\mathbb{G}}(k) \leftarrow \Pr[\rho_3 \in \mathbb{G}^2 \wedge \rho^* \neq ((\rho_1^*)^u, (\rho_2^*)^v, \rho_3) \ \wedge \\
&\quad \mathsf{H}_h^{k,\mathbb{G}^4,\mathbb{G}}(\rho^*) = (v/u) \cdot \mathsf{H}_h^{k,\mathbb{G}^4,\mathbb{G}}((\rho_1^*)^u, (\rho_2^*)^v, \rho_3) \bmod p \ \mid \\
&\quad (u, v, \rho_3) \stackrel{\mathsf{R}}{\leftarrow} \mathcal{A}(1^k, \rho^*, h, \mathbb{G})],
\end{aligned}$$

where $\mathbb{G} \stackrel{\mathsf{U}}{\leftarrow} \{\mathbb{G}\}_k$, $\rho^* \leftarrow (\rho_1^*, \rho_2^*, \rho_3^*) \stackrel{\mathsf{U}}{\leftarrow} \mathbb{G} \times \mathbb{G} \times \mathbb{G}^2$ and $h \stackrel{\mathsf{R}}{\leftarrow} \mathsf{KH}_k$.

TCR hash function family H is a generalized target collision resistant (GTCR) hash function family associated with $\{\mathbb{G}\}_k$ if for any probabilistic polynomial-time adversary $\mathcal{A}$, $\mathsf{Adv}\mathrm{GTCR}_{\mathsf{H},\mathcal{A}}^{\mathbb{G}}(k)$ is negligible in k.

Theorem 3. *The proposed KEM scheme is IND-CPCA secure, if the DDH assumption holds for $\{\mathbb{G}\}_{k\in\mathbb{N}}$, H is a GTCR hash function family, and F is a PRF family.*

The proof will be given in the full paper version of this paper.

Acknowledgments

The author would like to thank Masayuki Abe for his invaluable discussions and suggestions on the proposed paradigm, KEM and redundancy-free hybrid encryption. I also thank Alfred Menezes for his lectures on the AKE protocols and security models as well as insightful discussions and valuable comments. I am also grateful to David

Pointcheval and Michel Abdalla for their valuable comments on a preliminary version of this manuscript. I would like to express my appreciation of Kaoru Kurosawa, the PC chair of Asiacrypt 2007, for giving me an opportunity to present this paper at the proceedings as an invited talk manuscript.

References

1. Abe, M., Gennaro, R., Kurosawa, K. and Shoup, V., Tag-KEM/DEM: A New Framework for Hybrid Encryption and New Analysis of Kurosawa-Desmedt KEM, Adv. in Cryptology – Eurocrypt 2005, LNCS 3494, pp. 128-146 (2005).
2. Canetti, R. and Krawczyk, H., Analysis of key-exchange protocols and their use for building secure channels, Advances in Cryptology, EUROCRYPT 2001, LNCS 2045 (2001), http://eprint.iacr.org/2001/040.
3. Cramer, R. and Shoup, V., Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing, 33(1), 167-226, (2003).
4. Goldreich, O., Goldwasser, S. and Micali, S., How to Construct Random Functions. In Journal of the ACM, vol.33, no.4, pp.792-807 (1986).
5. Hastad, J., Impagliazzo, R., Levin, L. and Luby, M., A Pseudorandom Generator from any One-way Function. SIAM Journal on Computing, v28 n4, pp.1364-1396 (1999).
6. Krawczyk, H., HMQV: A high-performance secure Diffie-Hellman protocol, Advances in Cryptology, CRYPTO 2005, LNCS 3621 (2005), http://eprint.iacr.org/2005/176.
7. Kurosawa, K. and Desmedt, Y., A New Paradigm of Hybrid Encryption Scheme, Advances in Cryptology- CRYPTO 2004, LNCS 3152, Springer-Verlag, pp. 426-442 (2004).
8. LaMacchia, B., Lauter, K. and Mityagin, A., Stronger security of authenticated key exchange, Cryptology ePrint Archive, Report 2006/073, 2006, http://eprint.iacr.org/2006/073.
9. Law, L., Menezes, A., Qu, M., Solinas, J. and Vanstone, S., An efficient protocol for authenticated key agreement, Designs, Codes and Cryptography 28, pp.119–134 (2003).
10. Menezes, A., Another look at HMQV, Journal of Mathematical Cryptology 1, pp.148–175 (2007).
11. Matsumoto, T., Takashima, Y. and Imai, H., On Seeking Smart Public-key Distribution Systems. Transactions of the IECE of Japan, E69:99-106 (1986).
12. Naor, M. and Yung, M., Universal one-way hash functions and their cryptographic applications. In Proceedings of the 21st Annual ACM Symposium on Theory of Computing, pp.33-43 (1989).
13. Rompel, J., One-way functions are necessary and sufficient for secure signatures. In Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, pp.387-394 (1990).
14. Ustaoglu, B., Obtaining a secure and efficient key agreement protocol from (H)MQV and NAXOS, Cryptology ePrint Archive, Report 2007/123, 2007, http://eprint.iacr.org/2007/123.