New Paradigm for Practical Cryptosystems without Random Oracles

Tatsuaki Okamoto†

Abstract

This paper introduces a new paradigm for making various types of cryptographic primitives such as authenticated key exchange and key encapsulation without random oracles under three assumptions: the decisional Diffie-Hellman assumption, target collision resistant hash functions, and a class of pseudorandom functions. It describes a new two-pass authenticated key exchange (AKE) protocol (based on the public key infrastructure model) that is comparable in efficiency to the most efficient of the existing protocols and secure (under these assumptions), whereas existing efficient two-pass AKE protocols are secure in the random oracle model. This protocol is shown to be secure in the (currently) strongest security definition, the extended Canetti-Krawczyk (eCK) security definition. This paper also describes a key encapsulation mechanism (KEM) that is secure against adaptive chosen ciphertext attacks (i.e., CCA-secure) under these assumptions and almost as efficient as the Kurosawa-Desmedt KEM. The schemes presented this paper are validity-check-free, which implies that combining them with validity-check-free symmetric encryption (data encryption mechanism) will yield validity-check-free (e.g., free of message authentication code) CCA-secure hybrid encryption.

1. Introduction

The concept of public-key cryptosystems was introduced by Diffie and Hellman in 1976 to solve the key agreement problem over insecure networks (like the Internet). The standard security notion of a public-key cryptosystem (encryption) is security against adaptive chosen ciphertext attacks (i.e., CCA-security), and there are two major methodologies for designing practical CCA-secure public-key encryption: one is based on the random oracle model and the other on the standard model. A CCA-secure scheme in the standard model provides a real security guarantee, whereas a CCA-secure scheme in the random oracle model is guaranteed to be secure under unrealistic idealization of a hash function as an ideal random function. One of the most important topics for the last ten years has been to design a truly practical public-key cryptosystems in the standard model[*1].

The most common paradigm for designing practical public-key cryptosystems that are secure in the standard model is to combine a trapdoor function (e.g., Diffie-Hellman or RSA (Rivest, Shamir, Adleman) function) and target collision resistance (TCR) hash functions, where the security is proven under a trapdoor function assumption (e.g., DDH (decisional Diffie-Hellman) or strong RSA assumption) and the TCR hash function assumption [3]–[5]. This paper introduces a new paradigm for designing practical public-key cryptosystems, where a class of *pseudorandom functions* (PRFs) PRFs with pairwise-independent random sources (πPRFs), is used in addition to a trapdoor function (Diffie-Hellman function) and TCR hash function.

The concept of a PRF was introduced by Goldreich, Goldwasser, and Micali [6]. The PRF has been shown to exist if and only if a one-way function exists [6], [7]. Therefore, the existence of a PRF is one of the weakest assumptions, so it is one of the most funda-

† NTT Information Sharing Platform Laboratories
Musashino-shi, 180-8585 Japan
Email: okamoto.tatsuaki@lab.ntt.co.jp

[*1] For a general explanation of the standard model and random oracle model, see for example [1], [2].

mental primitives in cryptography[*2].

Since a TCR hash function (and the slightly more general concept of the universal one-way hash function) has also been shown to exist if and only if a one-way function exists [9], [10], the TCR hash function and PRF are on the same level as (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 part of the input (or a keyed hash function) can be assumed to be a PRF (and a πPRF).

Authenticated key exchange (AKE) protocols have been extensively studied to enhance the security of the Diffie-Hellman (DH) key exchange protocol, which was proposed in 1976, because the DH protocol is not secure against the man-in-the-middle attack[11]–[17].

This paper presents a two-pass AKE protocol that has the following properties.

1. It is comparable in efficiency to MQV [16], HMQV [14], and CMQV [17] (the scheme's message size for one party is that of MQV plus the size of three group elements, and the computational complexity for a session of this scheme is around 3.7 group exponentiations, while that of MQV is around 2.2 group exponentiations).
2. The assumption and model for its proof of security are three assumptions (DDH, TCR hash function, and πPRF) and the 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 [15].
4. Its security proof reduction efficiency is better than those of previous protocols in the random oracle model.

This paper also presents a *CCA-secure* (i.e., secure against adaptive chosen ciphertext attacks) key encapsulation mechanism (KEM) under these assumptions that is almost as efficient as the Kurosawa-Desmedt KEM [5].

The schemes presented in this paper are validity-check-free, which implies validity-check-free (e.g., free of message authentication code (MAC-free)) CCA-secure hybrid encryption if they are combined with validity-check-free CCA-secure symmetric encryption (data encryption mechanism (DEM)). Therefore, their ciphertexts can be decrypted with no validity-check.

2. Preliminaries

2.1 Notation

$\mathbb{N}$ 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 $k > n$.

If 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. If a is a value, $A(x) \rightarrow a$ denotes the event that A outputs a on input x. If A is a set, $y \overset{\mathsf{U}}{\leftarrow} A$ denotes that y is uniformly selected from A. If A is a value, $y \leftarrow A$ denotes that y is set to A.

In this paper, the underlying machines are considered to be uniform Turing machines, but the results can easily be extended to non-uniform Turing machines.

2.2 Decisional Diffie-Hellman (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 groups $\mathbb{G}$ 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

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

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

2.3 Pseudo-random Function (PRF)

Let $k \in \mathbb{N}$ be a security parameter. A 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 PRFs 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 ele-

[*2] For a general explanation of cryptographic primitives, see for example [8].

ment 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
$\mathrm{AdvPRF}_{\mathsf{F},\mathcal{A}}(k) \leftarrow |\Pr[\mathcal{A}^{F}(1^k,\mathcal{D},\mathcal{R}) \rightarrow 1] - \Pr[\mathcal{A}^{RF}(1^k,\mathcal{D},\mathcal{R}) \rightarrow 1]|$,
where $\Sigma \overset{R}{\leftarrow} \mathsf{Seed}_k$, $\sigma \overset{U}{\leftarrow} \Sigma$, $\mathcal{D} \overset{R}{\leftarrow} \mathsf{Dom}_k$, $\mathcal{R} \overset{R}{\leftarrow} \mathsf{Rng}_k$, $F \leftarrow \mathsf{F}_{\sigma}^{k,\Sigma,\mathcal{D},\mathcal{R}}$, and $RF : \mathcal{D} \rightarrow \mathcal{R}$ is a truly random function $(\forall \rho \in \mathcal{D}\ RF(\rho) \overset{U}{\leftarrow} \mathcal{R})$.

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

2.4 Pseudo-random function with pairwise-independent random sources (πPRF)

Here, we introduce a specific class of PRFs, πPRFs.

Let $k \in \mathbb{N}$ be a security parameter and F be a PRF family associated with $\{\mathsf{Seed}_k\}_{k\in\mathbb{N}}$, $\{\mathsf{Dom}_k\}_{k\in\mathbb{N}}$, and $\{\mathsf{Rng}_k\}_{k\in\mathbb{N}}$.

We then define a πPRF family for F.

Let $\Sigma \overset{R}{\leftarrow} \mathsf{Seed}_k$, $\mathcal{D} \overset{R}{\leftarrow} \mathsf{Dom}_k$, $\mathcal{R} \overset{R}{\leftarrow} \mathsf{Rng}_k$, and $RF: \mathcal{D} \rightarrow \mathcal{R}$ be a truly random function $(\forall \rho \in \mathcal{D}\ RF(\rho) \overset{U}{\leftarrow} \mathcal{R})$.

Let I_Σ be a set of indices regarding Σ such that there exists a deterministic polynomial-time algorithm, f_Σ: $I_\Sigma \rightarrow \Sigma$, that on input $i \in I_\Sigma$, outputs $\sigma_i \in \Sigma$.

Let $\sigma_{i_0}, \sigma_{i_1}, \ldots, \sigma_{i_{t(k)}}$ be random variables indexed by I_Σ, where $i_j \in I_\Sigma$ $(j = 0, 1, \ldots, t(k))$ and $t(k)$ is a polynomial of k. Let σ_{i_0} be pairwisely independent from other variables, $\sigma_{i_1}, \ldots, \sigma_{i_{t(k)}}$ and each variable be uniformly distributed over Σ. That is, for any pair of $(\sigma_{i_0}, \sigma_{i_j})$ $(j = 1, \ldots, t(k))$, for any $(x, y) \in \Sigma^2$, $\Pr[\sigma_{i_0} \rightarrow x \wedge \sigma_{i_j} \rightarrow y] = \Pr[\sigma_{i_0} \rightarrow x] \cdot \Pr[\sigma_{i_j} \rightarrow y] = 1/|\Sigma|^2$.

Let $\mathcal{A}^{F,\,I_\Sigma}$ be a probabilistic polynomial-time machine $\mathcal{A}$ that queries $q_j \in \mathcal{D}$ along with $i_j \in I_\Sigma$ to F and receives the reply $\mathsf{F}_{\sigma_{i_j}}^{k,\,\Sigma,\,\mathcal{D},\,\mathcal{R}}(q_j)$ for each $j = 0, 1, \ldots, t(k)$.

Let $\mathcal{A}^{RF,\,I_\Sigma}$ be the same as $\mathcal{A}^{F,\,I_\Sigma}$ except that $\mathsf{F}_{\sigma_{i_j}}^{k,\,\Sigma,\,\mathcal{D},\,\mathcal{R}}(q_0)$ is replaced by $RF(q_0)$.

For all k, we define
$\mathrm{Adv\pi PRF}_{\mathsf{F},\,I_\Sigma,\,\mathcal{A}}(k) \leftarrow |\Pr[\mathcal{A}^{F,\,I_\Sigma}(1^k, \mathcal{D}, \mathcal{R}) \rightarrow 1] - \Pr[\mathcal{A}^{RF,\,I_\Sigma}(1^k, \mathcal{D}, \mathcal{R}) \rightarrow 1]|$.

F is a πPRF family with index $\{(I_\Sigma, f_\Sigma)\}_{\Sigma \in \mathsf{Seed}_k,\, k\in\mathbb{N}}$ if for any probabilistic polynomial-time adversary $\mathcal{A}$, $\mathrm{Adv\pi PRF}_{\mathsf{F},\,I_\Sigma,\,\mathcal{A}}(k)$ is negligible in k.

2.5 Target collision resistant (TCR) hash function

Let $k \in \mathbb{N}$ be a security parameter. A 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 of 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, $h \overset{R}{\leftarrow} \mathsf{KH}_k$, $\mathcal{D} \overset{R}{\leftarrow} \mathsf{Dom}_k$, and $\mathcal{R} \overset{R}{\leftarrow} \mathsf{Rng}_k$, 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
$\mathrm{AdvTCR}_{\mathsf{H},\mathcal{A}}(k) \leftarrow \Pr[\rho \in \mathcal{D} \wedge \rho \neq \rho^* \wedge \mathsf{H}_h^{k,\mathcal{D},\mathcal{R}}(\rho) = \mathsf{H}_h^{k,\mathcal{D},\mathcal{R}}(\rho^*) | \rho \overset{R}{\leftarrow} \mathcal{A}(1^k, \rho^*, h, \mathcal{D}, \mathcal{R})]$,
where $\mathcal{D} \overset{R}{\leftarrow} \mathsf{Dom}_k$, $\mathcal{R} \overset{R}{\leftarrow} \mathsf{Rng}_k$, $\rho^* \overset{U}{\leftarrow} \mathcal{D}$ and $h \overset{R}{\leftarrow} \mathsf{KH}_k$. H is a TCR hash function family if for any probabilistic polynomial-time adversary $\mathcal{A}$, $\mathrm{AdvTCR}_{\mathsf{H},\mathcal{A}}(k)$ is negligible in k.

2.6 PKI-based authenticated key exchange (AKE) and eCK security definition

This section outlines the eCK security definition [16] for two-pass AKE protocols based on the public key infrastructure (PKI) model and follows the description in [17].

The eCK definition assumes that there are n parties, which are modeled as probabilistic polynomial-time Turing machines. We assume that these parties have agreed some common parameters (common reference strings) in the AKE protocol before the protocol is started. The parameter selection mechanism is out of the 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. $\hat{A}$ (or $\hat{B}$) denotes the static public key A (or B) of party $\mathcal{A}$ ($\mathcal{B}$) together with a certificate. The certifying authority (CA) is not assumed to require parties to prove possession of their static private keys, but the CA is required to verify 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, it is called 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 $\mathcal{A}$ and $\mathcal{B}$ are said to be the owner and peer of the session, respectively. 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 to be 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 Send (*message*), thereby controlling the activation of sessions. In order to capture private information that may leak, adversary $\mathcal{M}$ is allowed the following queries:

- EphemeralKeyReveal (sid) The adversary obtains the ephemeral private key associated with session sid.
- SessionKeyReveal (sid) The adversary obtains the session key for session sid, provided that the session has a session key.
- StaticKeyReveal (pid) The adversary learns the static private key of party pid.
- EstablishParty (pid) This query makes it possible for the adversary to register a static public key on behalf of a party. In this way, the adversary completely controls that party.

If a party pid is established by EstablishParty (pid) query issued by adversary $\mathcal{M}$, then we call the party *dishonest*. If a party is not dishonest, we call it *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 Test (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 $R^* \overset{\cup}{\leftarrow} \{0, 1\}^{|K^*|}$. The adversary wins the game if he correctly guesses whether the key is random or not. To define the game, we need the notion of a *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 $\overline{\text{sid}}$ be the session identifier of the matching session of sid, if it exists. We define session sid to be *fresh* if none of the following conditions hold:

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

Now we are ready to present the eCK security notion.

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

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

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 an indication of protocol failure).
- For any probabilistic polynomial-time adversary $\mathcal{M}$, $\text{AdvAKE}_{\mathcal{M}}(k)$ is negligible in k.

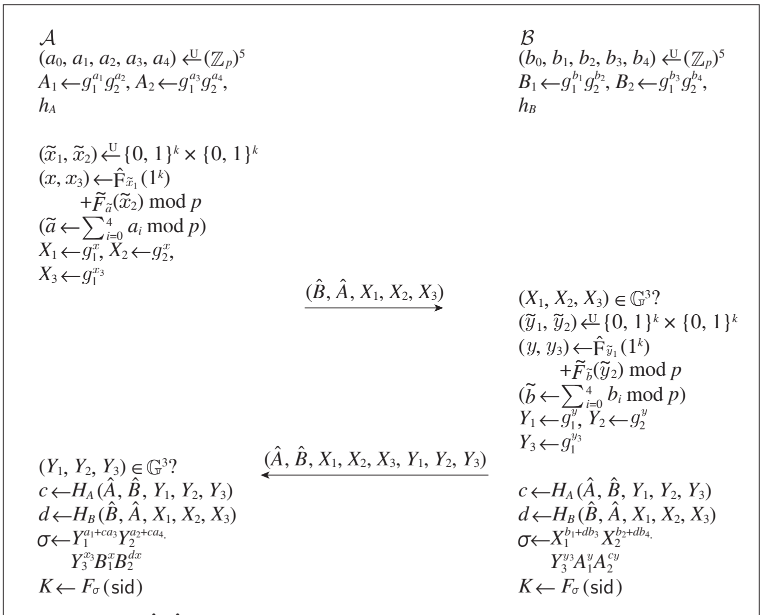
This security definition is stronger than the original form of Canetti-Krawczyk security [11] and it simultaneously captures all the known desirable security properties for authenticated key exchange including resistance to key-compromise impersonation attacks, weak perfect forward secrecy, and resilience to the leakage of ephemeral private keys.

2.7 Key encapsulation mechanism (KEM)

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

1. K, the key generation algorithm, is a probabilistic polynomial-time 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 probabilistic polynomial-time 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).

For all (pk, sk) output by key generation algorithm K and for all (K^*, C^*) output by key encryption algo-


$\mathcal{A}$
$(a_0, a_1, a_2, a_3, a_4) \overset{\mathrm{U}}{\leftarrow} (\mathbb{Z}_p)^5$
$A_1 \leftarrow g_1^{a_1} g_2^{a_2}, A_2 \leftarrow g_1^{a_3} g_2^{a_4},$
h_A
$\mathcal{B}$
$(b_0, b_1, b_2, b_3, b_4) \overset{\mathrm{U}}{\leftarrow} (\mathbb{Z}_p)^5$
$B_1 \leftarrow g_1^{b_1} g_2^{b_2}, B_2 \leftarrow g_1^{b_3} g_2^{b_4},$
h_B
$(\tilde{x}_1, \tilde{x}_2) \overset{\mathrm{U}}{\leftarrow} \{0, 1\}^k \times \{0, 1\}^k$
$(x, x_3) \leftarrow \hat{\mathrm{F}}_{\tilde{x}_1}(1^k)$
$+\tilde{F}_{\tilde{a}}(\tilde{x}_2) \bmod p$
$(\tilde{a} \leftarrow \sum_{i=0}^{4} a_i \bmod p)$
$X_1 \leftarrow g_1^x, X_2 \leftarrow g_2^x,$
$X_3 \leftarrow g_1^{x_3}$
$(\hat{B}, \hat{A}, X_1, X_2, X_3)$
$(X_1, X_2, X_3) \in \mathbb{G}^3?$
$(\tilde{y}_1, \tilde{y}_2) \overset{\mathrm{U}}{\leftarrow} \{0, 1\}^k \times \{0, 1\}^k$
$(y, y_3) \leftarrow \hat{\mathrm{F}}_{\tilde{y}_1}(1^k)$
$+\tilde{F}_{\tilde{b}}(\tilde{y}_2) \bmod p$
$(\tilde{b} \leftarrow \sum_{i=0}^{4} b_i \bmod p)$
$Y_1 \leftarrow g_1^y, Y_2 \leftarrow g_2^y$
$Y_3 \leftarrow g_1^{y_3}$
$(\hat{A}, \hat{B}, X_1, X_2, X_3, Y_1, Y_2, Y_3)$
$(Y_1, Y_2, Y_3) \in \mathbb{G}^3?$
$c \leftarrow H_A(\hat{A}, \hat{B}, Y_1, Y_2, Y_3)$
$d \leftarrow H_B(\hat{B}, \hat{A}, X_1, X_2, X_3)$
$\sigma \leftarrow Y_1^{a_1+ca_3} Y_2^{a_2+ca_4}.$
$Y_3^{x_3} B_1^x B_2^{dx}$
$K \leftarrow F_\sigma(\mathsf{sid})$
$c \leftarrow H_A(\hat{A}, \hat{B}, Y_1, Y_2, Y_3)$
$d \leftarrow H_B(\hat{B}, \hat{A}, X_1, X_2, X_3)$
$\sigma \leftarrow X_1^{b_1+db_3} X_2^{b_2+db_4}.$
$Y_3^{y_3} A_1^y A_2^{cy}$
$K \leftarrow F_\sigma(\mathsf{sid})$

Here, $\mathsf{sid} \leftarrow (\hat{A}, \hat{B}, X_1, X_2, X_3, Y_1, Y_2, Y_3)$. and $(A_1, A_2, B_1, B_2) \in \mathbb{G}^4$ is confirmed indirectly through the certificates.

Fig. 1. New AKE.

rithm $\mathsf{E}(pk)$, $\mathsf{D}(sk, C^*) = K^*$ 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 $\mathcal{A}$. The attack game (IND-CCA2 game) used to define security against adaptive chosen ciphertext attacks (IND-CCA2) is described below.

1. The challenger $\mathcal{C}$ generates a pair of keys $(pk, sk) \overset{\mathrm{R}}{\leftarrow} \mathsf{K}(1^k)$ 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 C), and DO returns $\mathsf{D}_{sk}(C_i)$ to $\mathcal{A}$.
3. $\mathcal{A}$ submits the encryption query to C. The encryption oracle EO in C selects $b^* \overset{\mathrm{U}}{\leftarrow} \{0, 1\}$ and computes $(C^*, K^*) \leftarrow \mathsf{E}(pk)$ and returns (C^*, K^*) to $\mathcal{A}$ if $b^* = 0$ and (C^*, K^*) if $b^* = 1$, where $\mathrm{R}^* \overset{\mathrm{U}}{\leftarrow} \{0, 1\}^{|K^*|}$ (C^* is called the *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 $\mathcal{A}$, $\mathrm{AdvKEM}_{\mathcal{A}}^{\text{IND-CCA2}}(k) \leftarrow |\Pr[b = b^*] - 1/2|$ 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 adversary $\mathcal{A}$, $\mathrm{AdvKEM}_{\mathcal{A}}^{\text{IND-CCA2}}(k)$ is negligible in k.

3. New AKE protocol

3.1 Protocol

Let $k \in \mathbb{N}$ be a security parameter, let $\mathbb{G} \overset{\mathrm{U}}{\leftarrow} \{\mathbb{G}\}_k$ be a group with security parameter k, and let $(g_1, g_2) \overset{\mathrm{U}}{\leftarrow} \mathbb{G}^2$, where the order of $\mathbb{G}$ is prime p and $|p| = k$. Let H be a TCR hash function family, $\hat{\mathsf{F}}$ and $\tilde{\mathsf{F}}$ be PRF families and F be a πPRF family.

$(\mathbb{G}, g_1, g_2)$, H, F, $\tilde{\mathsf{F}}$, and $\hat{\mathsf{F}}$ are the system parameters common among all users of this AKE protocol (although $\tilde{\mathsf{F}}$ and $\hat{\mathsf{F}}$ can be set privately by each party). We assume that the system parameters are selected by a trusted third party.

Party $\mathcal{A}$'s static private key is $(a_1, a_2, a_3, a_4) \overset{\mathrm{U}}{\leftarrow} (\mathbb{Z}_p)^5$ and $\mathcal{A}$'s static public key is $A_1 \leftarrow g_1^{a_1} g_2^{a_2}$, $A_2 \leftarrow g_1^{a_3} g_2^{a_4}$. Here, $h_A \overset{\mathrm{R}}{\leftarrow} \mathsf{KH}_k$ indexes a TCR hash function $H_A \leftarrow \mathsf{H}_{h_A}^{k, D_H, \mathrm{R}_H}$, where $D_H \leftarrow \Pi_k \times \mathbb{G}^4$, $\mathcal{R}_H \leftarrow \mathbb{Z}_p$, and Π_k denotes the space of possible certificates for static public keys.

Similarly, Party $\mathcal{B}$'s static private key is $(b_0, b_1, b_2,$

$b_3, b_4) \overset{\mathsf{U}}{\leftarrow} (\mathbb{Z}_p)^5$ and $\mathcal{B}$'s static public key is $B_1 \leftarrow g_1^{b_1} g_2^{b_2}$, $B_2 \leftarrow g_1^{b_3} g_2^{b_4}$. Here, $h_B \overset{\mathsf{R}}{\leftarrow} \mathsf{KH}_k$ indexes a TCR hash function $H_B \leftarrow \mathsf{H}_{hB}^{k, D_H, \mathrm{R_H}}$.

$\mathcal{A}$ and $\mathcal{B}$ set πPRF and PRFs $F \leftarrow \mathsf{F}^{k, \Sigma_\mathsf{F}, \mathcal{D}_\mathsf{F}, \mathcal{R}_\mathsf{F}}$, $\tilde{F} \leftarrow \tilde{F}^{k, \Sigma_{\tilde{F}}, \mathcal{D}_{\tilde{F}}, \mathcal{R}_{\tilde{F}}}$ and $\hat{F} \leftarrow \hat{F}^{k, \Sigma_{\hat{F}}, \mathcal{D}_{\hat{F}}, \mathcal{R}_{\hat{F}}}$, where $\Sigma_\mathsf{F} \leftarrow \mathbb{G}$, $\mathcal{D}_\mathsf{F} \leftarrow (\Pi_k)^2 \times \mathbb{G}^{10}$, $\mathcal{R}_\mathsf{F} \leftarrow \{0, 1\}^k$, $\Sigma_{\tilde{\mathsf{F}}} \leftarrow \mathbb{Z}_p$, $\mathcal{D}_{\tilde{\mathsf{F}}} \leftarrow \{0, 1\}^k$, $\mathcal{R}_{\tilde{\mathsf{F}}} \leftarrow (\mathbb{Z}_p)^2$, $\Sigma_{\hat{\mathsf{F}}} \leftarrow \{0, 1\}^k$, $\mathcal{D}_{\hat{\mathsf{F}}} \leftarrow \{0, 1\}^k$, and $\mathcal{R}_{\hat{\mathsf{F}}} \leftarrow (\mathbb{Z}_p)^2$.

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

1. Select an ephemeral private key $(\tilde{x}_1, \tilde{x}_2) \overset{\mathsf{U}}{\leftarrow} \{0, 1\}^k \times \{0, 1\}^k$.
2. Compute $\tilde{a} \leftarrow \sum_{i=0}^{4} a_i \bmod p$ $(x, x_3) \leftarrow \hat{F}_{\tilde{x}_1}(1^k) + \tilde{F}_{\tilde{a}}(\tilde{x}_2) \bmod p$ (as two-dimensional vectors) and the ephemeral public key $(X_1 \leftarrow g_1^x, X_2 \leftarrow g_2^x, X_3 \leftarrow g_1^{x_3})$. Note that the value of (x, x_3) (and $\tilde{a}$) is only computed in a computation process of the ephemeral public key from ephemeral and static private keys.
3. Erase (x, x_3) and the whole computation history of the ephemeral public key.
4. Send $(\hat{B}, \hat{A}, X_1, X_2, X_3)$ to $\mathcal{B}$.

Upon receiving $(\hat{B}, \hat{A}, X_1, X_2, X_3)$, party $\mathcal{B}$ verifies that $(X_1, X_2, X_3) \in \mathbb{G}^3$. If so, perform the following procedure.

1. Select an ephemeral private key $(\tilde{y}_1, \tilde{y}_2) \overset{\mathsf{U}}{\leftarrow} \{0, 1\}^k \times \{0, 1\}^k$.
2. Compute $\tilde{b} \leftarrow \sum_{i=0}^{4} b_i \bmod p$ $(y, y_3) \leftarrow \hat{F}_{\tilde{y}_1}(1^k) + \tilde{F}_{\tilde{b}}(\tilde{y}_2) \bmod p$ (as two-dimensional vectors) and the ephemeral public key $(Y_1 \leftarrow g_1^y, Y_2 \leftarrow g_2^y, Y_3 \leftarrow g_1^{y_3})$
3. Erase (y, y_3) and the whole computation history of the ephemeral public key.
4. Send $(\hat{A}, \hat{B}, X_1, X_2, X_3, Y_1, Y_2, Y_3)$ to $\mathcal{A}$.

Upon receiving$(\hat{A}, \hat{B}, X_1, X_2, X_3, Y_1, Y_2, Y_3)$, party $\mathcal{A}$ checks if he sent $(\hat{B}, \hat{A}, X_1, X_2, X_3)$ to $\mathcal{B}$. If so, $\mathcal{A}$ verifies that $(Y_1, Y_2, Y_3) \in \mathbb{G}^3$.

To compute the session key, $\mathcal{A}$ computes $\sigma_A \leftarrow Y_1^{a_1+ca_3} Y_2^{a_2+ca_4} Y_3^{x_3} B_1^x B_2^{dx}$, and $\mathcal{B}$ computes $\sigma_B \leftarrow X_1^{b_1+db_3} X_2^{b_2+db_4} X_3^{y_3} A_1^y A_2^{cy}$, where $c \leftarrow H_A(\hat{A}, \hat{B}, Y_1, Y_2, Y_3)$ and $d \leftarrow H_B(\hat{B}, \hat{A}, X_1, X_2, X_3)$.

If they are correctly computed, then $\sigma \leftarrow \sigma_A (= \sigma_B)$. The session key is $K \leftarrow F_\sigma$ (sid), where sid $\leftarrow (\hat{A}, \hat{B}, X_1, X_2, X_3, Y_1, Y_2, Y_3)$.

3.2 Security

Theorem 1.

The new AKE protocol is secure (in the sense of Definition 2) if the DDH assumption holds for $\{\mathbb{G}\}_{k \in \mathbb{N}}$, where H is a TCR hash function family, $\tilde{\mathsf{F}}$ and $\hat{\mathsf{F}}$ are PRF families and F is a πPRF family with index $\{(I_\mathbb{G}, f_\mathbb{G})\}_{\mathbb{G} \in \{\mathbb{G}\}_k, k \in \mathbb{N}}$, where $I_\mathbb{G} \leftarrow \{(V, W, d) | (V, W, d) \in \mathbb{G}^2 \times \mathbb{Z}_p\}$ and $f_\mathbb{G} : (V, W, d) \mapsto V^{r_1+dr_2} W$ with $(r_1, r_2) \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^2$.

4. New KEM scheme

4.1 Scheme

This section shows 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 $sk \leftarrow (x_1, x_2, y_1, y_2) \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^4$.

Public key: $g_1 \overset{\mathsf{U}}{\leftarrow} \mathbb{G}$, $g_2 \overset{\mathsf{U}}{\leftarrow} \mathbb{G}$, $z \leftarrow g_1^{x_1} g_2^{x_2}$, $w \leftarrow g_1^{y_1} g_2^{y_2}$, $H \leftarrow \mathsf{H}_h^{k, \mathcal{D}_H, \mathcal{R}_H}$ and $F \leftarrow \mathsf{F}^{k, \Sigma_\mathsf{F}, \mathcal{D}_\mathsf{F}, \mathcal{R}_\mathsf{F}}$, where $h \overset{\mathsf{R}}{\leftarrow} \mathsf{KH}_k$, $\mathcal{D}_H \leftarrow \{pk\} \times \mathbb{G}^2$ (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 $r \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$ and compute

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

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

4.2 CCA security

Theorem 2.

The new KEM scheme is IND-CCA2-secure if the DDH assumption holds for $\{\mathbb{G}\}_{k \in \mathbb{N}}$, if H is a TCR hash function family, and if F is a πPRF family with index $\{(I_\mathbb{G}, f_\mathbb{G})\}_{\mathbb{G} \in \{\mathbb{G}\}_k, k \in \mathbb{N}}$, where $I_\mathbb{G} \leftarrow \{(V, W, d) | (V, W, d) \in \mathbb{G}^2 \times \mathbb{Z}_p\}$ and $f_\mathbb{G} : (V, W, d) \mapsto V^{r_1+dr_2} W$ with $(r_1, r_2) \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p^2$.

5. Concluding remarks

This paper introduced a new paradigm for making various types of cryptographic primitives such as authenticated key exchange and key encapsulation without random oracles under the three standard assumptions: the DDH assumption, TCR hash functions, and πPRFs. These schemes are secure without random oracles and almost as efficient as the most efficient schemes secure in the random oracle model. Therefore, they are good candidates for replacing practical schemes in the random oracle model and will have many practical applications[*3] [18].

References

[1] http://en.wikipedia.org/wiki/Standard_Model_%28cryptography%29

[2] http://en.wikipedia.org/wiki/Random_oracle_model

[3] M. Abe, R. Gennaro, K. Kurosawa, and V. Shoup, "Tag-KEM/DEM: A New Framework for Hybrid Encryption and New Analysis of Kurosawa-Desmedt KEM," Adv. in Cryptology, Eurocrypt 2005, Lecture Notes in Computer Science, Vol. 3494, pp. 128–146, 2005.

[4] R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack," SIAM Journal on Computing, Vol. 33, No. 1, pp. 167–226, 2003.

[5] K. Kurosawa and Y. Desmedt, "A New Paradigm of Hybrid Encryption Scheme," Advances in Cryptology, Crypto 2004, Lecture Notes in Computer Science,Vol. 3152, Springer-Verlag, pp. 426–442, 2004.

[6] O. Goldreich, S. Goldwasser, and S. Micali, "How to Construct Random Functions." Journal of the ACM, Vol. 33, No. 4, pp. 792–807, 1986.

[7] J. Hastad, R. Impagliazzo, L. Levin, and M. Luby, "A Pseudorandom Generator from any One-way Function," SIAM Journal on Computing, Vol. 28, No. 4, pp. 1364–1396, 1999.

[8] http://en.wikipedia.org/wiki/Cryptographic_primitive

[9] M. Naor and M. Yung, "Universal one-way hash functions and their cryptographic applications," Proceedings of the 21st Annual ACM Symposium on Theory of Computing, pp. 33–43, 1989.

[10] J. Rompel, "One-way functions are necessary and sufficient for secure signatures. Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, pp. 387–394, 1990.

[11] R. Canetti, and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," Advances in Cryptology, Eurocrypt 2001, Lecture Notes in Computer Science, Vol. 2045, 2001.
http://eprint.iacr.org/2001/040.

[12] A. Menezes, "Another look at HMQV," Journal of Mathematical Cryptology 1, pp. 148–175, 2007.

[13] T. Matsumoto, Y. Takashima, and H. Imai, "On Seeking Smart Public-key Distribution Systems," Transactions of the IECE of Japan, E69:99–106, 1986.

[14] H. Krawczyk, "HMQV: A high-performance secure Diffie-Hellman protocol," Advances in Cryptology, Crypto 2005, Lecture Notes in Computer Science, Vol. 3621, 2005.
http://eprint.iacr.org/2005/176.

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

[16] L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone, "An efficient protocol for authenticated key agreement," Designs, Codes and Cryptography 28, pp. 119–134, 2003.

[17] B. Ustaoglu, "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

[18] T. Okamoto, "Authenticated Key Exchange and Key Encapsulation without Random Oracles," http:/eprint.iacr.org/2007/473

Tatsuaki Okamoto
Research Fellow, Okamoto Research Laboratory, NTT Information Sharing Platform Laboratories.

He received the B.E., M.E., and Dr.Eng. degrees from University of Tokyo, Tokyo, in 1976, 1978, and 1988, respectively. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan and International Association for Cryptologic Research.

[*3] Detailed descriptions and proofs of the theorems are given in [18].