Tag-KEM/DEM: A New Framework for Hybrid Encryption and A New Analysis of Kurosawa-Desmedt KEM *

Masayuki Abe[1], Rosario Gennaro[2], Kaoru Kurosawa[3] and Victor Shoup[4]

[1] NTT Information Sharing Platform Laboratories, NTT Corporation, Japan
[2] IBM T.J.Watson Research Center, USA
[3] Ibaraki University, Japan
[4] New York University, USA

Abstract. This paper presents a novel framework for generic construction of hybrid encryption schemes secure against chosen ciphertext attack. Our new framework yields new and more efficient CCA-secure schemes, and provides insightful explanations about existing schemes that do not fit into the previous frameworks. This could result in finding future improvements. Moreover, it allows immediate conversion from a class of threshold public-key encryption to a hybrid one without considerable overhead, which is not possible in the previous approaches.
Finally we present an improved security proof of the Kurosawa-Desmedt scheme, which removes the original need for information-theoretic key derivation and message authentication functions. We show that the scheme can be instantiated with any computationally secure such functions, thus extending the applicability of their paradigm, and improving its efficiency.

1 Introduction

A fundamental task of cryptography is to protect the secrecy of messages transmitted over public communication lines. For this purpose we use *encryption schemes* which use some secret information (a key) to encode a message in a way that an eavesdropper cannot decode it. However, as networks become more open and accessible, it becomes apparently clear that an adversary may not be limited to eavesdropping, but may take a more active role. She may try to interact with honest parties, by, for example, sending ciphertexts to them (possibly related to the ciphertexts

* Work done while the first author is visiting IBM T.J. Watson Research Center. The fourth author was supported by NSF grant CCR-0310297. This paper is an extended abstract combining two separate results. Proofs and detailed definitions are presented in the full versions available as [3, 20].

she intends to decrypt) and analyze their response. Such active attacks can be proven to be much more powerful and hard to combat than passive ones (see for example [6]).

To model this type of attacks, the notion of *chosen-ciphertext security* was introduced by Naor and Yung [22] and developed by Rackoff and Simon [24], and Dolev, Dwork, and Naor [17]. Security against a chosen ciphertext attack (CCA security, in short) means that, even if the adversary is allowed to query a *decryption oracle* on ciphertexts of her choosing, then she obtains no information about messages encrypted in other ciphertexts. The first CCA-secure cryptosystems were presented in [22, 24, 17], but they were quite impractical, as they rely on generic techniques for non-interactive zero-knowledge. In a breakthrough result, Cramer and Shoup in [12] presented the first truly practical CCA-secure cryptosystem, whose security is based on the hardness of the decisional Diffie-Hellman problem. This construction was generalized in [13], using a new cryptographic primitive called *projective hash functions*.

Public-key encryption schemes often limit the message space to a particular group, which can be restrictive when one wants to encrypt arbitrary messages. For this purpose *hybrid* schemes are devised, composed by two parts. First a *Key Encapsulation Mechanism* (KEM) is invoked: a random group element is encrypted and then mapped via a key derivation function into a random key K. Then a *Data Encapsulation Mechanism* is performed: the previous key K is used to encrypt the message using a symmetric encryption scheme. A formal treatment is found in [27, 14].

In order to obtain a CCA-secure hybrid encryption, it is sufficient that both KEM and DEM are CCA-secure. (Accordingly, we refer the framework of [27, 14] as CCA KEM/DEM framework in this paper). Recently in [21], Kurosawa and Desmedt introduced a hybrid encryption scheme which is a modification of the hybrid scheme presented in [25]. Their scheme is interesting from both a theoretical and a practical point of view. When one looks at it as a KEM/DEM scheme, we do not know if their KEM is CCA-secure, yet the resulting scheme is CCA-secure and more efficient than the one in [25] both in computation and bandwidth. Thus the Kurosawa-Desmedt scheme points out that to obtain CCA-secure hybrid encryption, requiring both KEM/DEM to be CCA-secure, while being a sufficient condition, may not be a necessary one, and might indeed be an overkill. There are other hybrid encryption schemes in the literature, e.g.,[5, 23], which are very efficient, mostly in the random oracle model, but do not fit to the CCA KEM/DEM framework.

Our Contribution. Prompted by the above observation, we set out to investigate another framework that yields more efficient hybrid encryption and captures a wider variety of existing schemes. Our results can be summarized as follows:

– We introduce Tag-KEM: a KEM which also takes as input a *tag*. Though such a notion is known in the literature, e.g., [27], we give an extended syntax and show, somewhat surprisingly, that if one uses a CCA-secure Tag-KEM in a novel way then it is sufficient for the DEM to be secure simply against passive attackers in order to yield CCA-secure hybrid encryption.
– We present several constructions of CCA-secure Tag-KEMs based on various combinations of assumptions.
– We show that the Tag-KEM/DEM framework provides a simple way to create threshold versions of CCA-secure hybrid encryption schemes, which is not possible in the CCA KEM/DEM framework.
– We show how several schemes in the literature can be casted in our Tag-KEM/DEM framework. Furthermore we show that some of those schemes can actually be simplified when considered as instances of our framework.
– Finally, we present an improved proof of the Kurosawa-Desmedt scheme. The original proof required the use of information-theoretic key derivation and message authentication functions. We show that any computationally secure such function suffices for the security of the scheme. The improvement is not just theoretical, but it has important practical implications as well. First of all it allows for a modular design in which any secure key derivation and MAC function can be used. Moreover our proof yields shorter security parameters and thus improved efficiency.

2 Definitions

2.1 Key Encapsulation Mechanism with Tags

In CCA KEM/DEM framework of [14], a KEM consists of three algorithms as public-key encryption does, except that the encryption algorithm takes only pk and outputs a random one-time key and its encryption. The encryption function may also take an extra string (called tag) as an input associated to every ciphertext. In our model, we divide the encryption function into two functions in such a way that the first one selects a random key and the second one encrypts the key along with a given tag. We call a KEM that meets this model a Tag-KEM. Formally:

$(pk, sk) \leftarrow \mathsf{TKEM.Gen}(1^\lambda)$: A probabilistic algorithm that generates public-key pk and private-key sk. The public-key defines spaces for tags and encapsulated keys denoted by $\mathcal{T}$ and $\mathcal{K}_K$, respectively.

$(\omega, dk) \leftarrow \mathsf{TKEM.Key}(pk)$: A probabilistic algorithm that outputs one-time key $dk \in \mathcal{K}_D$ and internal state information ω that essentially carries dk. $\mathcal{K}_D$ is the key-space of DEM.

$\psi \leftarrow \mathsf{TKEM.Enc}(\omega, \tau)$: A probabilistic algorithm that encrypts dk (embedded in ω) into ψ along with τ, where τ is called a tag.

$dk \leftarrow \mathsf{TKEM.Dec}_{sk}(\psi, \tau)$: A decryption algorithm that recovers dk from ψ and τ. For soundness, $\mathsf{TKEM.Dec}_{sk}(\psi, \tau) = dk$ must hold for any sk, dk, ψ, and τ, associated by the above three functions.

Note that, in the above syntactic definition, τ is not included in ψ and explicitly given to $\mathsf{TKEM.Dec}$. Such explicit treatment of τ has some notational advantages when we consider an adversary who tries to alter the tag without affecting to the ciphertext.

Tag-KEM is a generalization of KEM because if the tag is a fixed string, it is a KEM. Tags associated to PKE or KEM can be found in the literature (e.g. see [28, 27]), but their syntactic definition and the purpose are different from those of ours; A tag is supposed to carry an identity of the encryptor and has to be fixed before DEM key is selected in their definition. Despite the limitations, their particular implementation fits also to our model without essential modification.

The security of Tag-KEM requires that the adversary should fail to distinguish whether a given dk is the one embedded in cipehrtext (ψ, τ) or not, with adaptive access to the decryption oracle. Let $\mathcal{O}$ be the decryption oracle, $\mathsf{TKEM.Dec}_{sk}(\cdot, \cdot)$. Let A_T be a polynomial-time oracle machine that plays the following game.

[GAME.TKEM]

Step 1. $(pk, sk) \leftarrow \mathsf{TKEM.Gen}(1^\lambda)$
Step 2. $\upsilon_1 \leftarrow {A_T}^{\mathcal{O}}(pk)$
Step 3. $(\omega, dk_1) \leftarrow \mathsf{TKEM.Key}(pk)$, $dk_0 \leftarrow \mathcal{K}_D$, $\delta \leftarrow \{0, 1\}$.
Step 4. $(\tau, \upsilon_2) \leftarrow {A_T}^{\mathcal{O}}(\upsilon_1, dk_\delta)$
Step 5. $\psi \leftarrow \mathsf{TKEM.Enc}(\omega, \tau)$
Step 6. $\tilde{\delta} \leftarrow {A_T}^{\mathcal{O}}(\upsilon_2, \psi)$

In Step 6, A_T is restricted not to ask (ψ, τ) to the decryption oracle $\mathcal{O}$. Variable υ_1, υ_2 are the internal state information of the adversary. Variable dk_δ is set to either dk_0 or dk_1 according to the value of $\delta \in \{0, 1\}$. Such convention is used throughout the paper unless otherwise noted. We define $\epsilon_{\mathrm{tkem},A_T} = \left|\Pr[\tilde{\delta} = \delta] - \frac{1}{2}\right|$ and $\epsilon_{\mathrm{tkem}} = \max_{A_T}(\epsilon_{\mathrm{tkem},A_T})$ where the maximum is taken over all machines. We say that a Tag-KEM is CCA-secure if ϵ_{tkem} is negligible in λ.

2.2 Data Encapsulation Mechanism and Public-key Encryption

Data Encapsulation Mechanism (DEM). A DEM is a symmetric encryption scheme that consists of two algorithms, $\mathsf{DEM.Enc}$ and $\mathsf{DEM.Dec}$ such that $\mathsf{DEM.Enc}$ is an encryption algorithm that encrypts m into ciphertext χ by using symmetric-key $dk \in \mathcal{K}_D$ and $\mathsf{DEM.Dec}$ is a corresponding decryption algorithm that recovers message m from input ciphertext χ by using the same symmetric-key.

For our purpose, we only require DEM to be indistinguishable against passive attacks. Namely, adversary A_D chooses two same-length messages and given a ciphertext of either of the messages from the encryption oracle and decide which of the messages is encrypted. It is stressed that the ciphertext is made by a random key and the key is used only once. DEM is one-time secure if any polynomial-time adversary succeeds in distinguishing the encryption oracle's choice with probability at most $\frac{1}{2} + \epsilon_{\mathrm{dem}}$ where ϵ_{dem} is negligible in the security parameter. One-time pad is a simple example that fulfills this security notion.

Public-key Encryption (PKE). A public-key encryption scheme consists of key-generation algorithm $\mathsf{PKE.Gen}$, encryption algorithm $\mathsf{PKE.Enc}$, and decryption algorithm $\mathsf{PKE.Dec}$, which are defined in a standard way. We also define chosen ciphertext security for PKE in the standard sense. That is, the adversary chooses two messages from the message space, and is given a ciphertext of either of them from the encryption oracle. The adversary is also given access to the decrypiton oracle that will decrypt any ciphertext except for the one made by the encryption oracle. PKE is CCA secure if any polynomial-time adversary succeeds in distinguishing the encryption oracle's choice with probability at most $\frac{1}{2} + \epsilon_{\mathrm{pke}}$ where ϵ_{pke} is negligible in the security parameter.

3 Generic Construction of Hybrid PKE

In $\mathsf{GAME.TKEM}$, it is important to see that the same ψ can be asked to the decryption oracle as long as τ is different. Therefore, to conform CCA-security, the pair (ψ, τ) must be non-malleable, which means that

CCA-secure Tag-KEM provides integrity to τ. We exploit this property to protect the DEM part so as to be non-malleable.

Now in our construction of hybrid PKE, we require that Tag-KEM accepts any string as a tag, i.e., $\mathcal{T} = \{0,1\}^*$. First of all, $\mathsf{PKE.Gen}$ is the same as $\mathsf{TKEM.Gen}$; Given security parameter λ, it outputs pubic-key pk and private-key sk. Encryption and decryption functions are as follows.

Function: $\mathsf{PKE.Enc}_{pk}(m)$	**Function:** $\mathsf{PKE.Dec}(sk, c)$
$(\omega, dk) \leftarrow \mathsf{TKEM.Key}(pk)$	$(\psi, \chi) \leftarrow c$
$\chi \leftarrow \mathsf{DEM.Enc}_{dk}(m)$	$dk \leftarrow \mathsf{TKEM.Dec}_{sk}(\psi, \chi)$
$\psi \leftarrow \mathsf{TKEM.Enc}(\omega, \chi)$	$m \leftarrow \mathsf{DEM.Dec}_{dk}(\chi)$
Output $c = (\psi, \chi)$	Output m

When the length of DEM key varies depending on the length of message, like one-time pad, the syntax of Tag-KEM will be modified so that $\mathsf{TKEM.Enc}$ and $\mathsf{TKEM.Dec}$ can take necessary information.

Theorem 1. *If Tag-KEM is CCA secure and DEM is one-time secure then the Hybrid PKE scheme in Section 3 is CCA secure. In particular,* $\epsilon_{\mathrm{pke}} < 2\epsilon_{\mathrm{tkem}} + \epsilon_{\mathrm{dem}}$.

Proof. Let A_E be a polynomial-time oracle machine that launches a chosen-ciphertext attack against the above hybrid encryption scheme. Let $\mathcal{O}$ denote the decryption oracle. Call this attack $\mathsf{GAME.PKE}$.

[GAME.PKE]
Step 1. $(pk, sk) \leftarrow \mathsf{TKEM.Gen}(1^\lambda)$
Step 2. $(m_0, m_1, v) \leftarrow {A_E}^{\mathcal{O}}(pk)$
Step 3. $b \leftarrow \{0,1\}$, $(\omega, dk) \leftarrow \mathsf{TKEM.Key}(pk)$, $\chi \leftarrow \mathsf{DEM.Enc}_{dk}(m_b)$,
$\psi \leftarrow \mathsf{TKEM.Enc}(\omega, \chi)$
Step 4. $\tilde{b} \leftarrow {A_E}^{\mathcal{O}}(v, (\psi, \chi))$

Let X denote the event that $\tilde{b} = b$ happens in $\mathsf{GAME.PKE}$. The goal of this proof is to bound $\Pr[X]$. First we modify Step-3 so that $\mathsf{DEM.Enc}$ takes random key $dk^{\times}$ instead of the legitimate one generated by $\mathsf{TKEM.Key}$. Call this game $\mathsf{GAME.PKE}$'. Let X' denote the event of $\tilde{b}' = b$ in $\mathsf{GAME.PKE}$'. We claim that $|\Pr[X] - \Pr[X']| \leq 2\,\epsilon_{\mathrm{tkem}}$, which is shown by constructing A_T that attacks the underlying Tag-KEM scheme by using A_E. First A_T is given public-key pk and passes it to A_E. Given m_0 and m_1 from A_E, A_T requests dk_δ to the encryption oracle of $\mathsf{GAME.TKEM}$.

A_T then selects $b \leftarrow \{0,1\}$ and computes $\chi = \mathsf{DEM.Enc}_{dk_\delta}(m_b)$. By sending $\mathsf{TKEM.Enc}$ χ as a tag, A_T receives ψ and sends ciphertext (ψ, χ) to A_E. Every decryption query from A_E is forwarded to decryption oracle $\mathsf{TKEM.Dec}$. If $\perp$ is returned, it is forwarded to A_E. Otherwise, A_K decrypts χ by using the key given from oracle $\mathsf{TKEM.Dec}$ and pass the resulting message to A_E. When A_E outputs $\tilde{b} = b$, A_K outputs $\tilde{\delta} = 1$ meaning that dk_δ is the real key. Otherwise, if A_E outputs $\tilde{b} \neq b$, A_K outputs $\tilde{\delta} = 0$ meaning that dk_δ is random. Now observe that the view of A_E is identical to that in $\mathsf{GAME.PKE}$ when $\delta = 1$, and that in $\mathsf{GAME.PKE}$' when $\delta = 0$. Accordingly, $\Pr[\tilde{b} = b|\delta = 1] = \Pr[X]$ and $\Pr[\tilde{b} = b|\delta = 0] = \Pr[X']$. Therefore,

$$\begin{aligned}
\Pr[\tilde{\delta} = \delta] - \frac{1}{2} &= \frac{1}{2}(\Pr[\tilde{\delta} = 1|\delta = 1] - \Pr[\tilde{\delta} = 1|\delta = 0]) \\
&= \frac{1}{2}(\Pr[\tilde{b} = b|\delta = 1] - \Pr[\tilde{b} = b|\delta = 0]) \\
&= \frac{1}{2}(\Pr[X] - \Pr[X'])
\end{aligned}$$

Since $\left|\Pr[\tilde{\delta} = \delta] - \frac{1}{2}\right| \leq \epsilon_{\text{tkem}}$, we have $|\Pr[X] - \Pr[X']| \leq 2\epsilon_{\text{tkem}}$.

Next, we show that A_E playing $\mathsf{GAME.PKE}$' essentially conducts a passive attack to DEM, i.e., $\left|\Pr[X'] - \frac{1}{2}\right| \leq \epsilon_{\text{dem}}$. It is shown by constructing A_D that plays $\mathsf{GAME.DEM}$ by using A_E. A_D first generates (pk, sk) by using $\mathsf{PKE.Gen}$ and gives pk to A_E. When m_0 and m_1 are given from A_E, A_D forwards them to encryption oracle of $\mathsf{GAME.DEM}$ and receives cipehrtext χ. It then computes ψ by following $\mathsf{TKEM.Key}$ and $\mathsf{TKEM.Enc}$ by using χ as a tag, and sends $c = (\psi, \chi)$ to A_E. Note that the key chosen by the encryption oracle of $\mathsf{GAME.DEM}$ and the one embedded in ψ are independent and randomly chosen. All decryption queries are correctly processed by using sk. When A_E outputs $\tilde{b}$, A_D outputs $\tilde{\xi} = \tilde{b}$. It is now easy to see that, in this construction, $\mathsf{GAME.PKE}$' is perfectly simulated and whenever A_E wins, so does A_D. Hence $\left|\Pr[X'] - \frac{1}{2}\right| \leq \epsilon_{\text{dem}}$. The major factors of the running time of A_D is that of A_E and that for simulating the decryption oracle which grows linearly in the number of decryption queries.

In summary, we have:

$$\begin{aligned}
|(\Pr[X] - \frac{1}{2}) - (\Pr[X'] - \frac{1}{2})| &\leq 2\epsilon_{\text{tkem}} \\
\epsilon_{\text{pke}} - \epsilon_{\text{dem}} &\leq 2\epsilon_{\text{tkem}} \\
\epsilon_{\text{pke}} &\leq 2\epsilon_{\text{tkem}} + \epsilon_{\text{dem}}
\end{aligned}$$

where ϵ_{tkem} and ϵ_{dem} are assumed negligible. $\square$

4 Construction of Tag-KEM

This section develops some methods for obtaining Tag-KEM from PKE or KEM. Note that KEM is generally obtained from PKE. Hence starting from KEM is more general.

4.1 Based on PKE with Long Plaintext

When CCA-secure PKE is available, the first idea would be to encrypt the tag as a part of the plaintext together with the DEM key to encapsulate. It indeed works well if there is enough space in a plaintext. Lengthy tags would be compressed by using a hash function. We can show that a target collision-free hash function [14], which is implied by universal one-way hash function, is sufficient for this purpose.

Formally, the construction is as follows. $\mathsf{TKEM.Gen}$ is essentially the same as $\mathsf{PKE.Gen}$; It outputs (pk, sk). It also selects hash function H. (For notational simplicity, we assume that H is included in pk and sk.) $\mathsf{TKEM.Key}$ chooses random dk from $\mathcal{K}_D$. It also outputs state information $\omega = pk||dk$. The encryption and decryption functions are as follows.

Function: $\mathsf{TKEM.Enc}(\omega, \tau)$	**Function:** $\mathsf{TKEM.Dec}_{sk}(\psi, \tau)$
$(pk, dk) \leftarrow \omega$ $\tau' = H(\tau)$ $\psi = \mathsf{PKE.Enc}_{pk}(dk\|\|\tau')$ Output ψ.	$dk\|\|\tau' \leftarrow \mathsf{PKE.Dec}(sk, \psi)$ If $\tau' = H(\tau)$, return dk. Return $\perp$, otherwise.

The resulting Tag-KEM is as secure as attacking the underlying PKE or hash function. Let ϵ_{tch} be the success probability of finding a target collision for H. The following theorem holds.

Theorem 2. *If PKE is CCA-secure and H is target collision-free, the above Tag-KEM is CCA-secure. Especially,* $\epsilon_{\mathrm{tkem}} \leq \epsilon_{\mathrm{pke}} + \epsilon_{\mathrm{tch}}$.

The RSA-based simple KEM [27] can be seen as an instance of this method in the random oracle model. Applying Theorem 1 yields a hybrid PKE that is a special case of [15]. Also, similar hybrid PKE is found in legendary protocols such as [4].

4.2 Based on CCA-Secure KEM and MAC

In this section we present a CCA-secure Tag-KEM based on a CCA-secure KEM and a secure message authentication code (MAC). Here, MAC is assumed to be strongly unforgeable against one-time chosen message and unbound MAC verification attack. That is, a MAC adversary is given a MAC for an arbitrary message of its choice and attempts to create a valid message-MAC pair that is different from the observed pair. The adversary also has polynomially many access to MAC verification oracle that verifies an arbitrary pair of a message and a MAC. We say MAC is one-time secure if it satisfies this security notion. Theoretically, such a MAC is available without intractability assumptions.

The idea is to encrypt a random key K using the KEM, and derive two keys dk, mk from K. The first, dk is the actual encryption key, while mk is used to MAC the tag. The resulting MAC is appended to the ciphertext. A decryptor not only checks that the KEM decryption is correct, but also checks that the MAC on the tag, using the decrypted key mk, is correct. A formal description follows.

Construction of Tag-KEM: Let $\Pi_L = (\mathsf{KEM.Gen}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec})$ be a KEM. Let $\mathsf{MAC} = (\mathsf{MAC.Sign}, \mathsf{MAC.Ver})$ be a MAC. Let $\mathsf{KDF}_2 : \mathcal{K}_K \to \mathcal{K}_D \times \mathcal{K}_M$ be a key derivation function where $\mathcal{K}_D$ is the key-space of DEM and $\mathcal{K}_M$ is the key-space of MAC. By using these components, we construct a Tag-KEM as follows. $\mathsf{TKEM.Gen}$ is the same as $\mathsf{PKE.Gen}$; It outputs (pk, sk). $\mathsf{TKEM.Key}$ is that, given pk, it computes $(K, \phi) \leftarrow \mathsf{KEM.Enc}_{pk}()$ and $(dk, mk) \leftarrow \mathsf{KDF}_2(K)$. Then it outputs dk and state information $\omega = (mk, \phi)$. The encryption and decryption functions are as in the table below.

The security of KDF_2 requires that its output distribution is indistinguishable from uniform one over the key-spaces. By ϵ_{kdf}, we denote the maximum advantage over all polynomial-time distinguisher. If ϵ_{kdf} is negligible, we say that KDF_2 is secure. If KDF_2 requires a key, it is generated by $\mathsf{TKEM.Gen}$ and included in pk and sk.

Function: $\mathsf{TKEM.Enc}(\omega, \tau)$	**Function:** $\mathsf{TKEM.Dec}_{sk}(\psi, \tau)$
$(mk, \phi) \leftarrow \omega$ $\sigma \leftarrow \mathsf{MAC.Sign}_{mk}(\tau)$ Output $\psi = (\phi, \sigma)$	$(\phi, \sigma) \leftarrow \psi$ $K \leftarrow \mathsf{KEM.Dec}_{sk}(\phi)$ $(dk, mk) \leftarrow \mathsf{KDF}_2(K)$ If $K = \perp$ or $\mathsf{MAC.Ver}_{mk}(\sigma, \tau) \neq 1$, output $\perp$. Otherwise, output dk.

Clearly the CCA security of the KEM scheme will prevent an adversary from gaining any advantage by manipulating the KEM ciphertext. On the other hand the security of the MAC will prevent an adversary from gaining any advantage by manipulating the MAC. The following theorem holds.

Theorem 3. *If Π_L is CCA secure, MAC is one-time secure, and KDF_2 is secure then the resulting Tag-KEM is CCA secure. In particular, $\epsilon_{\text{tkem}} \leq 4\epsilon_{\text{kem}} + q_D\,\epsilon_{\text{mac}} + 5\epsilon_{\text{kdf}}$ where q_D is the maximum number of decryption queries.*

Applying Theorem 1 to the above Tag-KEM yields the same hybrid encryption scheme as in CCA KEM/DEM framework. But by analysing the same scheme in our framework, we can show that CCA KEM is an overkill. In [3], it is shown that there exists a class of KEM that is strictly weaker than CCA but suffices for this construction.

4.3 Based on KEM with Hash function

We show another approach that might be available when the underlying PKE does not have enough plaintext length as needed in Section 4.1 and/or increasing ciphertext length as in Section 4.2 is not acceptable.

If a KEM uses a hash function, probably for integrity of ciphertext or plaintext, the KEM may be converted to a Tag-KEM simply by including the tag into the hash function. This approach is correct if the hash function is involved in the scheme in a 'meaningful' way and provides 'sufficient' security. Although generic construction that follows formal version of these intuitive terms can be shown, it does not seem quite useful due to its complexity. Showing that a KEM fits to the generic framework may not be simpler than directly proving that the resulting Tag-KEM scheme is secure. Indeed, in all cases we have in mind, the security proof can be done by minor or obvious modification of that of the original KEM (or PKE). Therefore, we only show two concrete constructions of Tag-KEM based on well known encryption schemes; OAEP+ [26] and Cramer-Shoup encryption [12]. In the following, the description of the original schemes are obtained just by dropping the tag τ.

From OAEP+. Let f be a one-way trapdoor permutation. OAEP+ encrypts dk with tag τ into ciphertext ψ in the following way:

$$r' = H'(r||dk||\tau),\ s = (G(r) \oplus dk)||r',\ w = H(s) \oplus r,\ \psi = f(s||w)$$

where r and r' are random and G, H, H' are random oracles [5].

Security is argued in the same way as the original one except the case that, for challenge ciphertext (ψ, τ) the adversary finds another valid ciphertext (ψ, τ'). Since ψ uniquely identifies r, r' and K, (ψ, τ') is valid only if $H'(r||dk||\tau) = H'(r||dk||\tau')$ holds. When H' outputs a k_1-bit string, such an event happens with probability at most $q_{H'}\, 2^{-k_1}$ where $q_{H'}$ is the maximum number of queries to H'. Based on this observation, we define game $\mathsf{GAME.0'}$ where decryption oracle returns $\perp$ for all queries that differs only in the tag part with the challenge ciphertext. The rest of the security proof is done in the same way as in the original paper [26] except for obvious modifications. Accordingly, only $q_{H'}\, 2^{-k_1}$ is an extra reduction cost to that of OAEP+.

From Cramer-Shoup Encryption. A Tag-KEM scheme based on Cramer-Shoup encryption over a multiplicative group, say G_q, of prime order q is the following. A private-key is $(x_1, x_2, y_1, y_2, z_1, z_2) \in Z_q$ and the public-key is $g_1, g_2 \leftarrow G_q^2$, and $c = g_1^{x_1} g_2^{x_2}$, $d = g_1^{y_1} g_2^{y_2}$, $h = g_1^{z_1} g_2^{z_2}$. The encryption function yields $dk = h^r$ where r is random, and ciphertext (u_1, u_2, v) such that

$$u_1 = g_1^r,\ u_2 = g_2^r,\ \alpha = H(u_1||u_2||\tau),\ v = c^r d^{\alpha r}$$

where H is a hash function. Decryption first checks if $v \stackrel{?}{=} u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$ and then recovers $dk = u_1^{z_1} u_2^{z_2}$. Applying Theorem 1 results in the hybrid PKE briefly mentioned in [12].

In contrast to [12] where H can be Target Collision Free, we need slightly stronger assumption to prove the security in our framework, which nevertheless has little practical impact. We say that H is *Random Prefix Collision-Free* if any adversary wins the following game with at most negligible probability. The adversary is first given H and outputs τ and then given random x and finally outputs x' and τ' such that $H(x||\tau) = H(x'||\tau')$. We can prove that the above scheme is secure Tag-KEM when H is random prefix collision free.

It holds that (Collision-Free) $\Rightarrow$ (Random Prefix Collision-Free) $\Rightarrow$ (Target Collision-Free). Hence it is reasonable to use cryptographic hash functions like SHA-1 which can be assumed collision-free. Nevertheless, we stress that random prefix collision-freeness may not necessarily be equivalent to collision-free because, for example, it is not clear how to perform a birthday attack in the above game (if the randomness of x affects to the output). Theoretically, we do not know constructions of random prefix collision-free hash functions from target collision-free or universal one-way hash functions, thus we resort to strong collision-freeness. The only drawback is that this requires a longer output (about twice as much because the birthday paradox applies here), but that does not affect our construction.

4.4 Based on ID-based PKE

An ID-based encryption scheme is selective-ID secure when it is secure against chosen ciphertext and chosen ID attacks provided that the target ID is committed at the beginning and the ID must not be included in any decryption query. It is shown in [10] that selective-ID ID-based encryption schemes (sIBE in short) can be strengthened to a full CCA secure ones by using one-time signature. Then, according to CCA KEM/DEM framework, an ID-based hybrid encryption scheme can be obtained by combining it with a CCA secure DEM. We show that the conversion from sIBE to full IBE also yields a Tag-KEM. Accordingly, the DEM part can be simplified to be a one-time secure DEM. The resulting scheme yields shorter ciphertexts than before.

Let $(\mathsf{SIG.Gen}, \mathsf{SIG.Sign}, \mathsf{SIG.Ver})$ be a one-time signature scheme where $\mathsf{SIG.Gen}$ is a key generation algorithm, $\mathsf{SIG.Sign}$ is a signature generation algorithm, and $\mathsf{SIG.Ver}$ is a signature verification algorithm. Let $\mathsf{sIBE.Enc}(pk, \mathrm{ID}, m)$ be the encryption function of an sIBE. Then, we construct a Tag-KEM scheme as follows: It encrypts (pk, dk) and τ into ciphertext $\psi = (vk, \phi, \sigma)$ where

$$(vk, sk) \leftarrow \mathsf{SIG.Gen}(1^\lambda),\ \phi \leftarrow \mathsf{sIBE.Enc}(pk, vk, dk),\ \sigma = \mathsf{SIG.Sign}(sk, \phi || \tau).$$

Including τ into the message to be signed provides integrity to the tag without affecting the security of the original scheme. Indeed, the security proof is almost the same as in [10] with obvious modification. The reduction cost does not change, either. One can extend the above Tag-KEM to ID-based Tag-KEM in the same way starting from a 2nd-level ID Encryption function that takes two ID's. (A given ID is assigned to the first ID and vk is assigned to the second ID.) For efficient implementations of sIBE based on standard cryptographic assumptions, we refer to [7].

In [8], Boneh and Katz improved the efficiency of [10] by replacing the one-time signature with commitment scheme (using hash function) and MAC. Part of their scheme can also be seen as a Tag-KEM.

5 Applications

5.1 Threshold Hybrid PKE

Designing a threshold hybrid PKE is not a trivial task. Even though threshold PKE is available, it is not clear how it can be extended to hybrid threshold PKE. By following CCA KEM/DEM framework, one will suffer from sharing KDF and $\mathsf{MAC.Ver}$, which are often implemented by number-theoretically unstructured primitives. Although these tasks are feasible using generic techniques from multi-party computation, we are focusing on efficient and practical solutions.

Since Tag-KEM/DEM framework allows the DEM part to be CPA, it immediately yields a threshold hybrid PKE once a shared Tag-KEM is available. Decrypting the DEM part is a local task. By defining CCA security for threshold PKE and DEM as in [28, 18], we can translate and prove Theorem 1 in the threshold setting. Accordingly, one can concentrate on constructing threshold Tag-KEM. A threshold KEM or PKE can be converted into a threshold Tag-KEM by following the construction in Section 4.3 or 4.1 without considerable overheads.

Threshold Cramer-Shoup encryption, secure against static adversaries, is shown in [1, 9], and the conversion technique in Section 4.3 (or result of section 4.1 with larger security parameter) can be used to obtain a threshold Cramer-Shoup Tag-KEM. Accordingly, by following the threshold version of Theorem 1, one can have a secure threshold hybrid encryption scheme in the standard model. Adaptive security can be achieved as well based on the adaptively secure threshold Cramer-Shoup encryption of [2].

5.2 Refined Fujisaki-Okamoto Conversion and More

We revisit the Fujisaki-Okamoto conversion [19] that provides secure construction of hybrid encryption in the random oracle model. By fitting their scheme into Tag-KEM/DEM framework, we can see that one of their assumptions can be eliminated and a refined version is obtained without loss of efficiency.

Let $\mathsf{PKE.Enc}_{pk}(\cdot\,;\cdot)$ be public-key encryption function where the last argument denotes a random coin used in the function. Fujisaki-Okamoto conversion combines PKE and DEM by using two random oracles, H and G, as follows:

$$\psi \leftarrow \mathsf{PKE.Enc}_{pk}(K; H(K||m)), \chi \leftarrow \mathsf{DEM.Enc}_{G(K)}(m).$$

A ciphertext is (ψ, χ). The resulting hybrid PKE is CCA-secure if PKE is one-way and DEM is one-time secure and $\mathsf{DEM.Enc}$ is a bijection between ciphertexts and messages for every fixed key.

Now one can observe that $\mathsf{PKE.Enc}_{pk}(K; H(K||\tau))$ works as a Tag-KEM encryption function that encapsulates DEM key $G(K)$. Then, according to Tag-KEM/DEM framework, we have slightly modified hybrid encryption:

$$\psi \leftarrow \mathsf{PKE.Enc}_{pk}(K; H(K||\chi)), \chi \leftarrow \mathsf{DEM.Enc}_{G(K)}(m)$$

which does not require $\mathsf{DEM.Enc}$ to be a bijection.

Similar observation applies to Bellare-Rogaway scheme [5], which is a special case of Fujisaki-Okamoto construction, and REACT-RSA [23].

5.3 Revisiting RCCA-secure PKE

This section revisits RCCA-secure PKE in [11] and show that their construction of CCA-secure hybrid PKE from RCCA-secure PKE can be improved by following our Tag-KEM/DEM framework.

The notion of RCCA-secure PKE is introduced in [11]. RCCA is a variant of CCA where the decryption oracle returns a special nonce 'test' when it receives a ciphertext that yields one of the questioned message, m_0 and m_1. Accordingly, even if the adversary can tweak the challenge ciphertext without affecting the embedded plaintext (such a feature is called benign-malleability [27]), sending it to the decryption oracle will give no advantage to the adversary in determining which of the questioned messages is hidden there. 'R' stands for 'replayable' in this sense. RCCA-security is a strict relaxation of CCA-security and proven useful for several cryptographic tasks, though, currently, there is no known instance of RCCA-secure PKE that is more efficient than known CCA-secure ones.

In [11], it is shown that combining RCCA-secure PKE and CCA-secure symmetric encryption can yield CCA-secure hybrid PKE. Suppose that a CCA-secure symmetric encryption is made by combining passively secure DEM and one-time MAC. Then, their construction is summarized as follows. Given message m, output ciphertext (ϕ, χ, σ) such that;

$$\phi \leftarrow \mathsf{PKE.Enc}_{pk}(dk||mk),\ \chi \leftarrow \mathsf{DEM.Enc}_{dk}(m||\phi),\ \sigma \leftarrow \mathsf{MAC.Sign}_{mk}(\chi)$$

where dk and mk, are chosen randomly from appropriate domains. It is stressed that ϕ is encrypted by DEM and this double-encryption structure is essential in their security proof. Due to this special structure, the construction does not fit to Tag-KEM/DEM framework. Below, we show a slightly more efficient variant that avoids double encryption and fits to Tag-KEM/DEM framework.

$$\phi \leftarrow \mathsf{PKE.Enc}_{pk}(dk||mk),\ \chi \leftarrow \mathsf{DEM.Enc}_{dk}(m),\ \sigma \leftarrow \mathsf{MAC.Sign}_{mk}(\chi||\phi)$$

Intuitively, applying MAC to ϕ offsets the benign-malleability of ϕ. The modified scheme yields shorter ciphertexts.

From the above, we derive a Tag-KEM scheme which is summarized as follows.

$$(K,\phi) \leftarrow \mathsf{KEM.Enc}_{pk}(),\ (dk,mk) \leftarrow KDF_2(K),\ \sigma \leftarrow \mathsf{MAC.Sign}_{mk}(\tau||\phi)$$

It can be seen as a variant of the construction shown in Section 4.2; MAC is applied to $\tau||\phi$ rather than to τ.

By defining RCCA-security for KEM in the same way as that for PKE, the following theorem can be proven.

Theorem 4. *If KEM is RCCA-secure, MAC is one-time secure, and DEM is secure, the above Tag-KEM is CCA-secure. Especially,* $\epsilon_{\text{tkem}} \leq 2\epsilon_{\text{rkem}} + (q_D+3)\epsilon_{\text{kdf}} + \frac{q_D}{2}\epsilon_{\text{mac}}$

According to Theorem 1, the modified hybrid PKE is CCA-secure. This uncovers the superfluousness of the double-encryption in the original construction and obtains a more efficient scheme.

6 New Proof for Kurosawa-Desmedt Scheme

Let us briefly recall the Kurosawa-Desmedt scheme from [21]. The group G, the hash function H and the public and secret key are as in the Cramer-Shoup scheme described earlier. It also uses a key derivation function KDF, such that for $v \in G$, $\mathsf{KDF}(v) = (k,K)$, where k is a message authentication key, and K is a symmetric encryption key.

Encryption of $m \in \{0,1\}^*$:

$$\begin{aligned}
&r \leftarrow \mathbb{Z}_q,\ u_1 \leftarrow g_1^r \in G,\ u_2 \leftarrow g_2^r \in G,\ \alpha \leftarrow H(u_1,u_2) \in \mathbb{Z}_q\\
&v \leftarrow c^r d^{r\alpha} \in G,\ (k,K) \leftarrow \mathsf{KDF}(v),\ e \leftarrow E_K(m),\ t \leftarrow \mathsf{MAC}_k(e)\\
&\text{output } C := (u_1,u_2,e,t)
\end{aligned}$$

Decryption of $C = (u_1,u_2,e,t)$:

$$\begin{aligned}
&\alpha \leftarrow H(u_1,u_2) \in \mathbb{Z}_q,\ v \leftarrow u_1^{x_1+y_1\alpha}u_2^{x_2+y_2\alpha} \in G,\ (k,K) \leftarrow \mathsf{KDF}(v)\\
&\text{if } t \neq \mathsf{MAC}_k(e) \text{ then } \textit{reject}\\
&\text{else output } m \leftarrow D_K(e)
\end{aligned}$$

It is possible to formalize this scheme as a Tag-KEM protocol. Indeed we can consider (u_1, u_2, t) as the Tag-KEM part (where (u_1, u_2) is the proper KEM part, e is the tag and t is a MAC on it), while e is the one-time DEM. This analysis seems identical to the one in Section 4.2, but here the basic KEM is not known to be CCA secure, so we can't invoke Theorem 3, and a proof specifically for this case is required.

The proof of security in [21] requires the MAC and KDF functions to be *information-theoretically secure*, i.e. if $v \in G$ is random, then at least the first component k of the output of $\mathsf{KDF}(v)$ should be (statistically close to) uniform; also for all e and t, if k is chosen at random, then $\Pr[\mathsf{MAC}_k(e) = t]$ is negligible. Our new proof of security, relaxes the above assumptions as follows: (i) if $v \in G$ is random, then at least the first component k of the output of $\mathsf{KDF}(v)$ should be computationally indistinguishable from uniform; (ii) the MAC function should be unforgeable. As we pointed out in the introduction this has a significant practical impact on the scheme.

Our proof shows that the Tag-KEM described above is CCA-secure. Using Theorem 1 we get that the hybrid scheme is CCA-secure as well.

Game 0. We start the proof by defining a game, called *Game 0*, which is an interactive computation between an *adversary* and a *simulator*. This game is simply the usual game used to define CCA security for Tag-KEM, in which the simulator provides the adversary's environment.

Initially, the simulator runs the key generation algorithm, obtaining the description of G, generators g_1 and g_2, keys for KDF and H (if any), along with the values $x_1, x_2, y_1, y_2 \in \mathbb{Z}_q$ and $c, d \in G$. The simulator gives the public key to the adversary.

During the execution of the game, the adversary makes a number of "decryption requests." Assume these requests are $C^{(1)}, \ldots, C^{(Q)}$, where $C^{(i)} = (u_1^{(i)}, u_2^{(i)}, e^{(i)}, t^{(i)})$. For each such request, the simulator decrypts the given ciphertext, and gives the adversary the result. We denote by $\alpha^{(i)}$, $v^{(i)}$, $k^{(i)}$, and $K^{(i)}$ the corresponding intermediate quantities computed by the decryption algorithm on input $C^{(i)}$. The oracle returns $K^{(i)}$ to the adversary.

The adversary may also make a single "challenge request." When such request is issued, the Tag-KEM encryption oracle generates $u_1 = g_1^r, u_2 = g_2^r, \alpha = H(u_1, u_2), v = c^r d^{r\alpha}$ and sets $(k_1, K_1) = \mathsf{KDF}(v)$. It also generates K_0 at random, and a random bit δ. The value K_δ is returned to the adversary who then produces a tag e and receives back (u_1, u_2, t) where $t = \mathsf{MAC}_{k_1}(e)$.

The only restriction on the adversary's requests is that after it makes a challenge request, subsequent decryption requests must be different from (u_1, u_2, e, t). At the end of the game, the adversary outputs $\hat{\delta} \in \{0, 1\}$.

Let X_0 be the event that $\hat{\delta} = \delta$. Security means that $|\Pr[X_0] - 1/2|$ should be negligible.

We prove this by considering other games, *Game 1*, *Game 2*, etc. These games will be quite similar to Game 0 in their overall structure, and will only differ from Game 0 in terms of how the simulator works. However, in each game, there will be well defined bits $\hat{\delta}$ and δ, so that in Game i, we always define X_i to the event that $\hat{\delta} = \delta$ in that game. All of these games should be viewed as operating on the same underlying probability space.

Before moving on, we make a couple of additional assumptions about the internal structure of Game 0 that will be convenient down the road. First, the simulator computes v as $(u_1)^{x_1+y_1\alpha}(u_2)^{x_2+y_2\alpha}$. This change is purely conceptual, since v has the same value either way. Second, we assume that g_2 is computed as $g_2 := g_1^w$ for $w \in_R \mathbb{Z}_q^*$. Second, we assume that the quantities r, u_1, u_2, α, v, k, and K_0, K_1 are computed at the very start of the game (they do not depend on values provided later by the adversary, so this can be done).

Game 1. This is the same as Game 0, except for the following differences. If the adversary ever submits $C^{(i)}$ for decryption with $(u_1^{(i)}, u_2^{(i)}) \neq (u_1, u_2)$ and $\alpha^{(i)} = \alpha$, the simulator *rejects* the given ciphertext.

In Game 1, the simulator may reject ciphertexts that would not have been rejected in Game 0. Let us call **Rejection Rule 0** the rule by which ciphertexts are rejected as in the ordinary decryption algorithm (i.e., the message authentication tags do not match). Let us call **Rejection Rule 1** this new rejection rule, introduced in Game 1.

Let F_1 be the event that the simulator applies Rejection Rule 1 in Game 1 to a ciphertext to which Rejection Rule 0 does not apply. Because Game 0 and Game 1 proceed identically until the this event occurs, we have

$$|\Pr[X_0] - \Pr[X_1]| \leq \Pr[F_1] \quad \text{and} \quad \Pr[F_1] \leq \epsilon_{\text{tcr}}, \tag{1}$$

where ϵ_{tcr} is the success probability that one can find a collision in H using resources similar to those of the given adversary. By assumption, ϵ_{tcr} is negligible.

Game 2. Now generate u_2 as $g_2^{r'}$ where $r' \in_R \mathbb{Z}_q$. We have

$$|\Pr[X_2] - \Pr[X_3]| \leq \epsilon_{\text{ddh}}, \tag{2}$$

where ϵ_{ddh} is the advantage with which one can solve the DDH problem, using resources similar to those of the given adversary. By assumption, ϵ_{ddh} is negligible.

Game 3. In this game, the simulator makes use of the value $w \in \mathbb{Z}_q$, where $g_2 = g_1^w$. The simulator did not need to make explicit use of this value in previous games. Indeed, we could not have used the DDH assumption if the simulator had to use w. However, we are now finished with the DDH assumption, and so the simulator is free to make use of w in this and subsequent games.

Game 3 is the same as Game 2, except that we introduce a new **Rejection Rule 2**: in responding to decryption requests, the simulator *rejects* any ciphertext $C^{(i)}$ such that $(u_1^{(i)})^w \neq u_2^{(i)}$, which is equivalent to saying that $\log_{g_1} u_1^{(i)} \neq \log_{g_2} u_2^{(i)}$.

Define F_4 to be the event that a ciphertext is rejected during Game 3 using Rejection Rule 2 to which Rejection Rules 0 and 1 are not applicable.

Clearly, we have

$$|\Pr[X_3] - \Pr[X_4]| \leq \Pr[F_4], \tag{3}$$

and we want to show that $\Pr[F_4]$ is negligible.

We postpone this until later. This is the step that allows us to avoid a circular argument in the original Kurosawa-Desmedt proof and forced them to make the information theoretic assumptions. Instead of attempting to bound $\Pr[F_4]$ right now, we shall patiently wait until Game 5, where it will be much easier. However, at this point we augment Game 3 just slightly: the simulator chooses $j \in \{1, \ldots, Q\}$, and we define F_4' to be the event that in Game 3, Rejection Rules 0 and 1 do not apply to $C^{(j)}$, but Rejection Rule 2 does apply to $C^{(j)}$. Clearly,

$$\Pr[F_4] \leq Q \Pr[F_4'], \tag{4}$$

and so it suffices to show that $\Pr[F_4']$ is negligible.

Game 4. Moving from Game 3 to Game 4 is a bit involved technically, yet the basic idea is *exactly* the same as that underlying the analysis in [12] of the original Cramer-Shoup encryption scheme. To motivate Game 4, we begin with some observations about Game 3. Let $x := x_1 + wx_2$ and $y := y_1 + wy_2$. Then we have $c = g_1^x$ and $d = g_1^y$. Also, for $i = 1, \ldots, Q$, if $\log_{g_1} u_1^{(i)} = \log_{g_2} u_2^{(i)}$ $v^{(i)} = u_1^{x+y\alpha^{(i)}}$. Moreover, v is uniformly distributed over G, independently of x and y. Further, if $\alpha^{(j)} \neq \alpha$ and

$\log_{g_1} u_1^{(j)} \neq \log_{g_2} u_2^{(j)}$ then $v^{(j)}$ is uniformly distributed over G, independently of x, y, and v. These observations follow from simple linear algebra considerations, as in [12].

Based on these observations, in Game 4, we compute a number of quantities in a different, but equivalent, manner. Let $\bar{x}, \bar{y}$ be random elements of $\mathbb{Z}_q$, and let $\bar{v}_1, \bar{v}_2$ be random elements of G. Let $(\bar{k}_i, \bar{K}_i) := \mathsf{KDF}(\bar{v}_i)$.

The key generation algorithm is modified as follows: $c \leftarrow g_1^{\bar{x}}, d \leftarrow g_1^{\bar{y}}$. The values k_1 and K_1 are set equal to $(\bar{k}_1, \bar{K}_1)$.

In processing decryption requests, for a given $C^{(i)}$ that is not subject to Rejections Rules 1 or 2, the value $v^{(i)}$ is computed as $(u_1^{(i)})^{\bar{x}+\bar{y}\alpha^{(i)}}$. Finally, we define the event F_5' to be the event in Game 4 that $C^{(j)}$ is subject to Rejection Rule 2, $C^{(j)}$ is not subject to Rejection Rule 1, and

- $(u_1^{(j)}, u_2^{(j)}) = (u_1^*, u_2^*)$ and $t^{(j)} = \mathsf{MAC}_{\bar{k}_1}(e^{(j)})$, or
- $(u_1^{(j)}, u_2^{(j)}) \neq (u_1^*, u_2^*)$ and $t^{(j)} = \mathsf{MAC}_{\bar{k}_2}(e^{(j)})$.

Note that the values $x_1, x_2, y_1, y_2, v^*, v^{(j)}$ are not used in Game 4.

We claim that

$$\Pr[X_4] = \Pr[X_5] \text{ and } \Pr[F_4'] = \Pr[F_5']. \tag{5}$$

This follows from the observations above — we have simply replaced one set of random variables by another set with same joint distribution.

It is perhaps helpful at this point to state how Game 4 works, starting from scratch:

- The simulator generates the description of G, along with a random generator g_1, and any keys for KDF and H. It computes $w, r, r', \bar{x}, \bar{y} \in_R \mathbb{Z}_q^*$, $g_2 := g_1^w$, $c := g_1^{\bar{x}}$, $d := g_1^{\bar{y}}$, $u_1 := g_1^r$, $u_2 := g_1^{wr'}$, $\bar{v}_1, \bar{v}_2 \in_R G$, $(\bar{k}_i, \bar{K}_i) \leftarrow \mathsf{KDF}(\bar{v}_i)$ and $j \in_R [1..Q]$.
  The simulator gives the description of G, the generators g_1 and g_2, keys for KDF and H (if any), along with c and d to the adversary.
- In processing a decryption request $C^{(i)} = (u_1^{(i)}, u_2^{(i)}, e^{(i)}, t^{(i)})$, the simulator first checks if $(u_1^{(i)})^w \neq u_2^{(i)}$; if so, the ciphertext is rejected. Otherwise, the simulator computes $\alpha^{(i)} := H(u_1^{(i)}, u_2^{(i)})$ and checks if $(u_1^{(i)}, u_2^{(i)}) \neq (u_1, u_2)$ and $\alpha^{(i)} = \alpha$; if so, the ciphertext is rejected. Otherwise, the simulator computes $v^{(i)}$ as $u_1^{\bar{x}+\bar{y}\alpha^{(i)}}$ and $(k^{(i)}, K^{(i)}) \leftarrow \mathsf{KDF}(v^{(i)})$. It then tests if $t^{(i)} = \mathrm{mac}_{k^{(i)}}(e^{(i)})$; if not, the ciphertext is rejected. Otherwise, the simulator returns $D_{K^{(i)}}(e^{(i)})$ to the adversary.

– In processing the challenge request, the simulator sets $K_1 = \bar{K}_1$, then chooses a random key K_0 and a random bit δ and gives K_δ to the adversary who responds with a tag e. Now the simulator computes $t \leftarrow \mathsf{MAC}_{\bar{k}_1}(e)$, and gives $C := (u_1, u_2, t)$ to the adversary.

Note that the values j and $\bar{v}_2$ (and the derived values $\bar{k}_2$ and $\bar{K}_2$) are not used in this game, other than to define the event F'_5.

Game 5. This is the same as Game 4, except that instead of applying KDF to derive the keys $\bar{k}_1, \bar{K}_1, \bar{k}_2, \bar{K}_2$, these keys are simply generated at random. Define the event F'_6 in Game 5 in the same way as it was defined in Game 4.

It is easy to see that

$$|\Pr[X_5] - \Pr[X_6]| \leq 2\epsilon_{\text{kdf}} \text{ and } |\Pr[F'_5] - \Pr[F'_6]| \leq 2\epsilon_{\text{kdf}}, \tag{6}$$

where ϵ_{kdf} is the advantage of distinguishing the output of the KDF from a random key pair, using resources similar to those of the given adversary. The factor of 2 comes from applying a standard "hybrid" argument to the two KDF outputs to be distinguished in moving from Game 4 to Game 5. By assumption, ϵ_{kdf} is negligible.

We claim that

$$\Pr[X_6] = 1/2 \tag{7}$$

This follows by construction — note that the key $\bar{K}_1$ in Game 5 is random, and is not used at all in the game, other than to define K_1. Therefore, conditioned on either $\delta = 0$ or $\delta = 1$, the adversary's view has the same conditional distribution; from this, it follows that the distribution of δ is independent of the adversary's view.

We also claim that

$$\Pr[F'_6] \leq 2\epsilon_{\text{mac}}, \tag{8}$$

where ϵ_{mac} is the probability of breaking the message authentication code, using resources similar to those of the given adversary. This also follows by construction — one has to make a simple "hybrid" argument to account for the fact that we are breaking one out of two message authentication schemes (one keyed with $\bar{k}_1$ and the other keyed with $\bar{k}_2$, whence the factor of 2). By assumption, ϵ_{mac} is negligible.

We are now in a position to complete the proof of security. By using Eqs. (4), (5), (6), (8), we get

$$\Pr[F_4] \leq Q(2\epsilon_{\text{mac}} + 2\epsilon_{\text{kdf}}). \tag{9}$$

Finally, combining (1), (2), (3), (5), (6), (7), and (9), we have:

$$|\Pr[X_0] - 1/2| \leq \epsilon_{\text{tcr}} + \epsilon_{\text{ddh}} + 2\epsilon_{\text{kdf}} + Q(2\epsilon_{\text{mac}} + 2\epsilon_{\text{kdf}}). \quad (10)$$

By assumption, the right-hand side of (10) is negligible, which finishes the proof.

Acknowledgments

The authors would like to thank Hugo Krawczyk, Shai Halevi, Mario Di Raimondo, Yevgeniy Dodis and Eiichiro Fujisaki for valuable discussion.

References

1. M. Abe. Robust distributed multiplication without interaction. In *CRYPTO '99*, LNCS 1666, pages 130–147. Springer-Verlag, 1999.
2. M. Abe and S. Fehr. Adaptively secure feldman VSS and applications to universally-composable threshold cryptography. IACR ePrint Archive 2004/119, 2004. Preliminary version apepars in *CRYPTO '04*, LNCS 3152, pages 317–334. Springer-Verlag, 2004.
3. M. Abe, R. Gennaro and K. Kurosawa. Tag-KEM/DEM: A new framework for hybrid encryption. IACR ePrint Archive 2005/027, 2005.
4. M. Bellare, J. Garay, R. Hauser, A. Herzberg, H. Krawczyk, M. Steiner, G. Tsudic, E. Van Herreweghen and M. Waidner. Design, implementation and Deployment of the *i*KP secure electronic payment system. *IEEE JSAC*, vol. 18, No. 4, April 2000.
5. M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *1st ACM CCCS*, pages 62–73. Association for Computing Machinery, 1993.
6. D. Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In *CRYPTO '98*, LNCS 1462, pages 1–12. Springer-Verlag, 1998.
7. D. Boneh and X. Boyen. Efficient Selective-ID Secure Identity Based Encryption. In *EUROCRYPT '04*, LNCS 3027, pages 223–238. Springer-Verlag, 2004.
8. D. Boneh and J. Katz. Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. IACR ePrint archive, 2004/261, 2004.
9. R. Canetti and S. Goldwasser. An efficient threshold public key cryptosystem secure against adaptive chosen ciphertext attack. In *EUROCRYPT '99*, LNCS 1592, pages 90–106. Springer-Verlag, 1999.
10. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT 2004*, LNCS 3027, pages 207–222. Springer-Verlag, 2004.
11. R. Canetti, H. Krawczyk, and J. Nielsen. Relaxing chosen-ciphertext security. IACR ePrint archive, 2003/174, 2003.
12. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO '98*, LNCS 1462, pages 13–25. Springer-Verlag, 1998.

13. R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT '02*, LNCS 2332, pages 45–64. Springer-Verlag, 2002.
14. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.
15. Y. Dodis and J. An. Concealment and Its Applications to Authenticated Encryption. In *EUROCRYPT '03*, LNCS 2656, pages 312–329, Springer-Verlag, 2003.
16. Y. Dodis, R. Gennaro, J. Haastad, H. Krawczyk, and T. Rabin. Randomness extraction and key derivation using the CBC, Cascade and HMAC modes. In *CRYPTO '04*, LNCS 3152, pages 494–510. Springer-Verlag, 2004.
17. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *23rd STOC*, pages 542–552, New York City, 1991.
18. P. Fouque and D. Pointcheval. Threshold cryptosystems secure against chosen-ciphertext attacks. In *Asiacrypt 2001*, LNCS 2248, pages 351–368. Springer-Verlag, 2001.
19. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO '99*, LNCS 1666, pages 537–554. Springer-Verlag, 1999.
20. R. Gennaro and V. Shoup. A note on an encryption scheme of Kurosawa and Desmedt. IACR ePrint archive, 2004/194, 2004.
21. K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. In *CRYPTO 2004*, LNCS 3152, pages 426–442. Springer-Verlag, 2004.
22. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd STOC*, pages 427–437, 1990.
23. T. Okamoto and D. Pointcheval. REACT: Rapid enhanced-security asymmetric cryptosystem transform. In *RSA '2001*, LNCS, Springer-Verlag, 2001.
24. C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO '91*, LNCS 576, pages 433–444. Springer-Verlag, 1992.
25. V. Shoup. Using hash functions as a hedge against chosen ciphertext attack. In *EUROCRYPT 2000*, LNCS 1807, pages 275–288. Springer-Verlag, 2000.
26. V. Shoup. OAEP reconsidered. In *CRYPTO 2001*, LNCS 2139, pages 239–259. Springer-Verlag, 2001.
27. V. Shoup. ISO 18033-2: An emerging standard for public-key encryption (committee draft). Available at `http://shoup.net/iso/`, June 3 2004.
28. V. Shoup and R. Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. In *EUROCRYPT '98*, LNCS 1403, pages 1–16. Springer-Verlag, 1998.