The Kurosawa-Desmedt Key Encapsulation is not Chosen-Ciphertext Secure

Javier Herranz Dennis Hofheinz Eike Kiltz

September 13, 2006

CWI Amsterdam, The Netherlands
{herranz,hofheinz,kiltz}@cwi.nl

Abstract

At CRYPTO 2004, Kurosawa and Desmedt presented a hybrid public-key encryption scheme that is chosen-ciphertext secure in the standard model. Until now it was unknown if the key-encapsulation part of the Kurosawa-Desmedt scheme by itself is still chosen-ciphertext secure or not. In this short note we answer this question to the negative, namely we present a simple chosen-ciphertext attack on the Kurosawa-Desmedt key encapsulation mechanism.

1 Introduction

Hybrid public-key encryption [4] consists of a key-encapsulation (KEM) part and a data encapsulation (DEM) part. For the hybrid scheme to be chosen-ciphertext secure it is sufficient that both the KEM and the DEM part are chosen-ciphertext secure [4]. Kurosawa and Desmedt [5] propose a very efficient public-key encryption scheme that, as a full hybrid encryption scheme is chosen-ciphertext secure (under the DDH assumption in the standard model). Due to its great flexibility and other reasons, in practise one always prefers a secure KEM over a full encryption scheme. However, until now it was unknown if the KEM part alone is still chosen-ciphertext secure or not. Whereas in [5] the authors claim that this is most likely not the case, in [2] this is referred to as an open problem. In this note we settle this question by giving a chosen-ciphertext attack on the KEM part of the Kurosawa-Desmedt encryption scheme. In fact our attack even holds in a stronger security setting, i.e. it breaks the chosen-plaintext non-malleability of the KD-KEM. Our results show that (under the DDH assumption) there exists a hybrid KEM/DEM encryption scheme that is chosen-ciphertext secure whereas the KEM part alone is not.

We stress that our results do not affect the security of the original Kurosawa-Desmedt hybrid public-key encryption scheme.

2 Key Encapsulation Mechnisms

We recall the definition of a *key encapsulation mechanism* (KEM). A KEM $\mathcal{KEM} = (\mathsf{Kg}, \mathsf{Encaps}, \mathsf{Decaps})$ with key-space $\mathsf{KeySp}(k)$ consists of three polynomial-time algorithms. Via $(pk, sk) \leftarrow \mathsf{Kg}(1^k)$ the randomized key-generation algorithm produces keys for security parameter $k \in \mathbb{N}$; via $(K, C) \leftarrow \mathsf{Encaps}(1^k, pk)$ a key $K \in \mathsf{KeySp}(k)$ together with a ciphertext C is created; via $K \leftarrow \mathsf{Decaps}(sk, C)$ the possessor of secret key sk decrypts ciphertext C to get back a key. For consistency, we require that for all $k \in \mathbb{N}$, and all $(K, C) \leftarrow \mathsf{Encaps}(1^k, pk)$ we have

$\Pr[\,\mathsf{Decaps}(sk, C) = K\,] = 1$, where the probability is taken over the choice of $(pk, sk) \leftarrow \mathsf{Kg}(1^k)$, and the coins of all the algorithms in the expression above.

The formal security of a KEM against chosen-ciphertext attacks is defined as follows.

Definition 2.1 The following experiment is associated to an adversary $\mathcal{A}$:

$$
\begin{array}{l}
\textbf{Experiment } \mathbf{Exp}^{\textit{kem-ind-cca}}_{\mathcal{KEM},\mathcal{A}}(k) \\
\quad (pk, sk) \leftarrow \mathsf{Kg}(1^k) \\
\quad K_0 \leftarrow \mathsf{KeySp}(k)\,;\ (K_1, C) \leftarrow \mathsf{Encaps}(pk) \\
\quad \delta \leftarrow \{0,1\}\,;\ K \leftarrow K_\delta \\
\quad \delta' \leftarrow \mathcal{A}^{\mathit{Decaps}(\cdot)}(pk, K, C) \\
\quad \text{If } \delta \neq \delta' \text{ then return 0 else return 1}
\end{array}
$$

where the oracle $\mathit{Decaps}(C)$ returns $K \leftarrow \mathsf{Decaps}(sk, C)$ with the restriction that $\mathcal{A}$ is not allowed to query $\mathit{Decaps}(\cdot)$ on the target ciphertext C. The advantage of $\mathcal{A}$ in the experiment is defined as

$$
\mathbf{Adv}^{\textit{kem-ind-cca}}_{\mathcal{KEM},\mathcal{A}}(k) \;=\; \left|\Pr\left[\,\mathbf{Exp}^{\textit{kem-ind-cca}}_{\mathcal{KEM},\mathcal{A}}(k) = 1\,\right] - \frac{1}{2}\right|
$$

A key encapsulation mechanism $\mathcal{KEM}$ is said to be *indistinguishable against chosen-ciphertext attacks* if the advantage function $\mathbf{Adv}^{\textit{kem-ind-cca}}_{\mathcal{KEM},\mathcal{A}}(k)$ is a negligible function in k for all polynomial-time adversaries $\mathcal{A}$.

3 Kurosawa-Desmedt Key Encapsulation

3.1 The KD Key Encapsulation Mechanism

Let $\mathbb{G}$ be a group of prime order p and let g_1, g_2 be two public generators of $\mathbb{G}$. Let $\mathsf{TCR} : \mathbb{G}\times\mathbb{G} \to \mathbb{Z}_p$ be a hash function. In the original paper a target collision-resistant hash function is proposed (whose definition can be looked up in [5]). However, the particular choice of the hash function TCR is independent of our attack. The key encapsulation part of the Kurosawa-Desmedt scheme is as follows:

$\mathsf{Kg}(1^k)$	$\mathsf{Encaps}(pk)$	$\mathsf{Decaps}(sk, C)$
$x_1, x_2, y_1, y_2 \leftarrow \mathbb{Z}_p^*$	$r \leftarrow \mathbb{Z}_p^*\,;\ c_1 \leftarrow g_1^r\,;\ c_2 \leftarrow g_2^r$	Parse C as (c_1, c_2)
$h_1 \leftarrow g_1^{x_1} g_2^{x_2}\,;\ h_2 \leftarrow g_1^{y_1} g_2^{y_2}$	$C \leftarrow (c_1, c_2) \in \mathbb{G}^2$	$t \leftarrow \mathsf{TCR}(C)$
$pk \leftarrow (h_1, h_2)$	$t \leftarrow \mathsf{TCR}(C)$	$K \leftarrow c_1^{tx_1+y_1} c_2^{tx_2+y_2}$
$sk \leftarrow (x_1, x_2, y_1, y_2)$	$K \leftarrow h_1^{tr} h_2^r \in \mathbb{G}$	Return K
Return (pk, sk)	Return (C, K)	

3.2 A chosen-ciphertext attack

We now describe a chosen-ciphertext attack on the Kurosawa-Desmedt KEM. Let K be the challenge key and $C = (c_1, c_2)$ be the challenge ciphertext an attacker receives in the chosen-ciphertext security game from Definition 2.1. The attacker picks two random integers $a, b \in \mathbb{Z}_p^*$ and computes the two ciphertexts

$$
C_a = (c_1^a, c_2^a), \quad C_b = (c_1^b, c_2^b)\,. \tag{1}
$$

Let $t = \mathsf{TCR}(C)$, $t_a = \mathsf{TCR}(C_a)$, and $t_b = \mathsf{TCR}(C_b)$. Note that the target collision resistance of TCR implies $t_a \neq t_b$ with overwhelming probability, so that we may simply assume $t_a \neq t_b$ here.

Now the attacker makes two queries to the decapsulation oracle:

$$K_a \leftarrow \mathit{Decaps}(C_a), \quad K_b \leftarrow \mathit{Decaps}(C_b)\ . \tag{2}$$

With K_a and K_b the correct challenge key (with respect to C) can be reconstructed as

$$\tilde{K} \leftarrow (K_a^{t_b/a} K_b^{-t_a/b})^{\frac{1}{t_b - t_a}} \cdot (K_a^{-1/a} K_b^{1/b})^{\frac{t}{t_b - t_a}}. \tag{3}$$

The attacker returns 1 if this key $\tilde{K}$ equals the challenge key K and 0 otherwise. This completes the description of the attack.

We claim that the above described attack successfully breaks the chosen-ciphertext security of the Kurosawa-Desmedt KEM with (maximal) advantage 1/2. Namely, we show that the key $\tilde{K}$ computed by the adversary in (3) always equals the key corresponding to the challenge chiphertext C which is defined as $\mathsf{Decaps}(sk, C) = c_1^{tx_1+y_1} c_2^{tx_2+y_2}$. This can be verified using the following easy claim:

Claim 3.1 $K_a^{t_b/a} K_b^{-t_a/b} = (c_1^{y_1} c_2^{y_2})^{t_b - t_a}$ and $K_a^{-1/a} K_b^{1/b} = (c_1^{x_1} c_2^{x_2})^{t_b - t_a}$.

Proof of Claim 3.1: Note that by definition of the decapsulation algorithm we have $K_a^{1/a} = ((c_1^a)^{t_a x_1 + y_1} (c_2^a)^{t_a x_2 + y_2})^{1/a} = c_1^{t_a x_1 + y_1} c_2^{t_a x_2 + y_2}$ and $K_b^{1/b} = c_1^{t_b x_1 + y_1} c_2^{t_b x_2 + y_2}$. Consequently,

$$\begin{aligned} K_a^{t_b/a} K_b^{-t_a/b} &= (c_1^{t_a x_1 + y_1} c_2^{t_a x_2 + y_2})^{t_b} (c_1^{t_b x_1 + y_1} c_2^{t_b x_2 + y_2})^{-t_a} \\ &= c_1^{t_a t_b x_1 + t_b y_1 - t_a t_b x_1 - t_a y_1} c_2^{t_a t_b x_2 + t_b y_2 - t_a t_b x_2 - t_a y_2} \\ &= (c_1^{y_1} c_2^{y_2})^{t_b - t_a}\ , \end{aligned}$$

and

$$\begin{aligned} K_a^{-1/a} K_b^{1/b} &= (c_1^{-t_a x_1 - y_1} c_2^{-t_a x_2 - y_2}) c_1^{t_b x_1 + y_1} c_2^{t_b x_2 + y_2} \\ &= c_1^{-t_a x_1 + t_b x_1} c_2^{-t_a x_2 + t_b x_2} \\ &= (c_1^{x_1} c_2^{x_2})^{t_b - t_a}\ . \end{aligned}$$

∎

Remark 3.2 This attack is also successful against a variant of the Kurosawa-Desmedt KEM where ciphertext are checked for consistency in the decapsulation algorithm, i.e. it is checked if $\log_{g_1} c_1 = \log_{g_2} c_2$. Such a check can be implemented by verifying if $c_1^{\omega} = c_2$, where $\omega = \log_{g_1} g_2$ which can be made part of sk. In our attack the two queried ciphertexts from (2) are obviously both consistent.

Remark 3.3 Our attack reconstructs the original challenge session key and therefore the KEM is not even one-way chosen-ciphertext secure.

Remark 3.4 Our attack in fact even breaks the chosen-plaintext (CPA) non-malleability of the Kurosawa-Desmedt KEM, i.e. the KEM is not NM-CPA. In a non-malleability attack an adversary is considered to be successful if she can come up with a vector of ciphertexts such that the respective decapsulated session keys of those ciphertexts are *meaningfully related* to the (unknown) session key of the challenge ciphertext. In the attack, given the challenge ciphertext C the adversary simply outputs the ciphertexts C_a and C_b as in (1) and defines the relation over $K = \tilde{K}, K_a, K_b$ as in (3). Note that this is a chosen-plaintext attack since the adversary never queries the decryption oracle. On the other hand it is easy to show that the Kurosawa-Desmedt KEM is indistinguishable under chosen-plaintext attacks (IND-CPA) under the DDH assumption.

Remark 3.5 In the Kurosawa-Desmedt public-key encryption scheme the symmetric key K is additionally hashed using a key-derivation function $\mathsf{KDF} : \mathbb{G} \to \{0,1\}^k$. We now show that even if one consideres this hash function as part of the Kurosawa Desmedt KEM then our attack still applies. The point is that the hash function KDF only has to satisfy relatively weak security properties, namely $\mathsf{KDF}(K)$ has to be uniformly distributed over $\{0,1\}^k$ if K is uniformly distributed over $\mathbb{G}$. In particular, a hash function that is efficiently invertible may satisfy this property. In that case the attacker can reconstruct K from $\mathsf{KDF}(K)$ and run the attack as described above.

Surely, if we model KDF as a random oracle [3] or if we are willing to base security on a much stronger assumption like the *Oracle Diffie-Hellman Assumption* [1] (which is an interactive assumption between the hash function KDF and the group) then the hashed version of the Kurosawa-Desmedt KEM indeed can be proved chosen-ciphertext secure.

References

[1] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In David Naccache, editor, *Topics in Cryptology – CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 143–158, San Francisco, CA, USA, April 8–12, 2001. Springer-Verlag, Berlin, Germany. 4

[2] Masayuki Abe, Rosario Gennaro, Kaoru Kurosawa, and Victor Shoup. Tag-KEM/DEM: A new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 128–146, Aarhus, Denmark, May 22–26, 2005. Springer-Verlag, Berlin, Germany. 1

[3] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press. 4

[4] Ronald Cramer and Victor 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. 1

[5] Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 426–442, Santa Barbara, CA, USA, August 15–19, 2004. Springer-Verlag, Berlin, Germany. 1, 2