Kurosawa-Desmedt KEM is not CCA2-secure

Seung Geol Choi [a], Javier Herranz [b], Dennis Hofheinz [c], Jung Yeon Hwang [d], Eike Kiltz [c,1], Dong Hoon Lee [d,2], Moti Yung [a]

[a] *Department of Computer Science, Columbia University, USA*

[b] *IIIA-CSIC, Bellaterra, Spain*

[c] *CWI Amsterdam, The Netherlands*

[d] *Graduate School of Information Management and Security, Korea University, Seoul, Korea*

Abstract

At CRYPTO 2004, Kurosawa and Desmedt presented a new hybrid encryption scheme that is CCA2-secure in the standard model. Until now it was unknown if the key encapsulation part of the Kurosawa-Desmedt scheme by itself is still CCA2-secure or not. In this note we answer this question to the negative, namely we present a simple CCA2 attack on the Kurosawa-Desmedt key encapsulation mechanism. Our attack further supports the design paradigm of Kurosawa and Desmedt to build CCA2-secure hybrid encryption from weak key encapsulation.

Key words: Cryptography, Hybrid encryption, Key encapsulation mechanism

1 Introduction

A hybrid encryption (HE) scheme is a public-key encryption method which consists of a key encapsulation mechanism (KEM) and a data encapsulation

Email addresses: sc2506@cs.columbia.edu (Seung Geol Choi), jherranz@iiia.csic.es (Javier Herranz), hofheinz@cwi.nl (Dennis Hofheinz), videmot@korea.ac.kr (Jung Yeon Hwang), kiltz@cwi.nl (Eike Kiltz), donghlee@korea.ac.kr (Dong Hoon Lee), moti@cs.columbia.edu (Moti Yung).

[1] Supported by the research program Sentinels. Sentinels is being financed by Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.

[2] Supported by the Brain Korea 21 Project.

mechanism (DEM) [8,16]: KEM encrypts a key for a symmetric encryption scheme under a public key and DEM encrypts a message under this key. Typically public-key encryption schemes (in particular HE schemes) are required to achieve CCA2-security; that is, a ciphertext should reveal no meaningful information about the original message, even to an active adversary who can probe a decryption oracle with chosen ciphertexts, both before and after the challenge ciphertext is available [9,15]. For construction of a CCA2-secure hybrid encryption scheme, the problem of security association between the KEM and DEM portions is not straightforward. Naturally, a good starting point is to consider the case where both the KEM and the DEM are CCA2-secure and, indeed, this combination was shown to be CCA2-secure [8]. The original reasonable belief [8] was that in a HE scheme, the KEM must be CCA2-secure. In fact, to achieve the CCA2-security of a hybrid encryption scheme, most research [8,3] uses the CCA2-security of the underlying KEM. Recently, in order to improve efficiency, Kurosawa and Desmedt [14] proposed a hybrid encryption scheme based on a *weak* KEM, that does not seem to achieve the level of CCA2-security [14]. Gennaro and Shoup extended the applicability of the idea by showing that the Kurosawa-Desmedt scheme with only *computationally secure* key derivation and message authentication functions is CCA2-secure [3,11]. The security of the Kurosawa-Desmedt KEM was later shown to follow the weaker security notions of *CCCA security* [12] and *LCCA security* [3] (which is defined with respect to a certain predicate). But until today it was not known whether the Kurosawa-Desmedt KEM is CCA2-secure or not [14,3].

In this paper we show that the KEM part of the Kurosawa-Desmedt hybrid encryption scheme is not CCA2-secure. More concretely, we present an attack that makes two queries to the KEM decapsulation oracle and reconstruct the original challenge key, hence breaking CCA2-security of the Kurosawa-Desmedt KEM. This provides an answer to the open question of [14,3]. We further show how to extend our attack to the generalized hash-free variant of the KD scheme based on hash proof systems. We stress that our results do not affect the security of the original Kurosawa-Desmedt hybrid encryption schemes. In contrary, they support the Kurosawa-Desmedt design paradigm of building efficient HE schemes from KEMs with (strictly) weaker security properties than CCA2-security.

2 Key Encapsulation Mechanisms

We briefly recall the definition of a key encapsulation mechanism (KEM) [8]. A KEM consists of the following three algorithms: a key-pair generation algorithm $\mathsf{KEM.KGen}$, a key encryption algorithm $\mathsf{KEM.Enc}$, a key decapsulation algorithm $\mathsf{KEM.Dec}$,

- $(pk, sk) \leftarrow \mathsf{KEM.KGen}(1^\lambda)$. A probabilistic polynomial-time (PPT) algorithm that on input security parameter λ, outputs a public/private key pair (pk, sk). A key space $\mathcal{K}$ is specified in pk.
- $(k_s, C) \leftarrow \mathsf{KEM.Enc}(1^\lambda, pk)$. A PPT encapsulation algorithm that on input pk, outputs a ciphertext C.
- $k_s \leftarrow \mathsf{KEM.Dec}(1^\lambda, sk, C)$. A deterministic decapsulation algorithm that on input a private key sk and a ciphertext C, outputs k_s.

We require that a KEM $(\mathsf{KEM.KGen}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec})$ satisfies the consistency property: $\forall\lambda \in \mathbb{N}, \forall(pk, sk) \leftarrow \mathsf{KEM.KGen}(1^\lambda), (k_s, C) \leftarrow \mathsf{KEM.Enc}(1^\lambda, pk)$, $k_s \leftarrow \mathsf{KEM.Dec}(1^\lambda, sk, C)$.

Next we define the IND-CCA2 security for a KEM via the following game with an adversary $\mathcal{A}$:

$$
\begin{aligned}
\textit{Initialization}: &\quad (pk, sk) \leftarrow \mathsf{KEM.KGen}(1^\lambda) \\
\textit{Preprocessing}: &\quad \eta_1 \leftarrow \mathcal{A}^{\mathsf{KEM.Dec}(sk,\cdot)}(pk) \\
\textit{Challenge/Response}: &\quad (k_s^{(0)}, C) \leftarrow \mathsf{KEM.Enc}(pk),\ k_s^{(1)} \xleftarrow{R} \mathcal{K};\ b \xleftarrow{R} \{0,1\} \\
\textit{Postprocessing}: &\quad \eta_2 \leftarrow \mathcal{A}^{\mathsf{KEM.Dec}(sk,\cdot)}(\eta_1, k_s^{(b)}, C) \\
\textit{Guess}: &\quad b' \leftarrow \mathcal{A}(\eta_2)
\end{aligned}
$$

In the above game variables η_1, η_2 are state information of the adversary. In the *Postprocessing* phase $\mathcal{A}$ is not allowed to issue the challenge ciphertext C to the decapsulation oracle $\mathsf{KEM.Dec}(sk, \cdot)$.

We define $\mathsf{Adv}^{\mathsf{IND-CCA2}}_{\mathcal{A},\mathsf{KEM}}(q_{dec}, t) = |\Pr[b = b'] - 1/2|$, where q_{dec} is the maximum number of decapsulation queries and t is the running time of $\mathcal{A}$. We also define that, for any q_{dec} and t, $\mathsf{Adv}^{\mathsf{IND-CCA2}}_{\mathsf{KEM}}(q_{dec}, t) = \max_{\mathcal{A}}[\mathsf{Adv}^{\mathsf{IND-CCA2}}_{\mathcal{A},\mathsf{KEM}}(q_{dec}, t)]$ where the maximum is taken over all $\mathcal{A}$. We say that KEM is CCA2-secure if $\mathsf{Adv}^{\mathsf{IND-CCA2}}_{\mathsf{KEM}}(q_{dec}, t)$ is negligible.

3 Kurosawa-Desmedt Key Encapsulation Mechanisms

In this section, let us briefly recall Kurosawa-Desmedt KEMs [14]. For more details, refer to [14].

3.1 Basic Kurosawa-Desmedt KEM

Let G be a commutative group of prime order q and g_1, g_2 be random generators of G. This scheme uses a *target collision resistant* hash function H.[3]

- $\mathsf{KEM.KGen}$. Choose $x_1, x_2, y_1, y_2 \in \mathbb{Z}_q$ at random and compute $c = g_1^{x_1} g_2^{x_2}, d = g_1^{y_1} g_2^{y_2}$. The public key is $pk = (g_1, g_2, c, d)$ and the private key is $sk = (x_1, x_2, y_1, y_2)$.

- $\mathsf{KEM.Enc}$. Given a public key $pk = (g_1, g_2, c, d)$, choose $r \in \mathbb{Z}_q$ at random and compute
$$u_1 = g_1^r, \ u_2 = g_2^r$$
and a key $v = c^r d^{r\alpha}$ where $\alpha = \mathsf{H}(u_1, u_2)$. Then output (u_1, u_2) as the KEM-ciphertext and v as the key.

- $\mathsf{KEM.Dec}$. Given a ciphertext (u_1, u_2) and a private key $sk = (x_1, x_2, y_1, y_2)$, return the key
$$v = u_1^{x_1 + y_1 \alpha} \cdot u_2^{x_2 + y_2 \alpha},$$
where $\alpha = \mathsf{H}(u_1, u_2)$.

3.2 KEM of Kurosawa-Desmedt Scheme Based on the Variant of HPS

We briefly review the KEM part of the modification of the Kurosawa-Desmedt scheme, based on a variant of Hash Proof Systems [7]. We denote this KEM as *HPS-based Kurosawa-Desmedt KEM*.

- Let G be a commutative group of prime order q. This scheme uses a public injective function $\Gamma : G^2 \to \mathbb{Z}_q^n$ for some (sufficiently large) n.

- $\mathsf{KEM.KGen}$. Generate a group G and two generators g_1, g_2 of G at random. Select $i_0, \ldots, i_n, j_0, \ldots, j_n \in \mathbb{Z}_q$ at random. Compute $s_t = g_1^{i_t} g_2^{j_t}$ for $0 \leq t \leq n$. The public key is $pk = (g_0, g_1, s_0, s_1, \ldots, s_n)$ and the secret key is $sk = (i_0, \ldots, i_n, j_0, \ldots, j_n)$.

- $\mathsf{KEM.Enc}$. Choose $r \in \mathbb{Z}_q$ at random and compute the values $u_1 = g_1^r$ and $u_2 = g_2^r$, and the key $v = (s_0 s_1^{a_1} \ldots s_n^{a_n})^r$, where $\Gamma(u_1, u_2) = (a_1, \ldots, a_n)$. Output (u_1, u_2) as the KEM-ciphertext and v as the key.

[3] Our attack does not exploit any specific property of H so we may even assume it is an ideal hash function like a random oracle [5].

- KEM.Dec. Given a ciphertext $(u_1 = g_1^r,\ u_2 = g_2^r)$ and a private key sk, compute the key v as follows.

$$v = u_1^{i_0+a_1 i_1+\ldots+a_n i_n} u_2^{j_0+a_1 j_1+\ldots+a_n j_n},$$

where $\Gamma(u_1, u_2) = (a_1, \ldots, a_n)$.

4 The CCA2 Attack on the Kurosawa-Desmedt KEMs

In this section we prove that the Kurosawa-Desmedt KEMs are not CCA2-secure.

Lemma 1. The KEM part in the Kurosawa-Desmedt hybrid encryption scheme is not CCA2-secure. In particular, we present an efficient adversary that reconstructs the real key in the adaptive chosen ciphertext attack game making two queries to the decapsulation oracle.

Proof. Let $(u_1 = g_1^r, u_2 = g_2^r)$ be the challenge ciphertext given to the adversary $\mathcal{A}$. Now $\mathcal{A}$ uses the following algorithm that computes the real key $v = c^r d^{r\alpha}$, where $\alpha = \mathsf{H}(u_1, u_2)$.

- First $\mathcal{A}$ chooses two distinct $w_1, w_2 \in \mathbb{Z}_q^* \backslash \{1\}$. Using the challenge ciphertext ψ=(u_1, u_2), $\mathcal{A}$ computes two ciphertexts ψ_i=$(u_1^{w_i} = g_1^{rw_i}, u_2^{w_i} = g_2^{rw_i})$ such that $\alpha_2 - \alpha_1 \neq 0$, where $\alpha_i = \mathsf{H}(u_1^{w_i}, u_2^{w_i})$ for i=1,2. If the condition does not hold, $\mathcal{A}$ repeats the previous step to obtain such ciphertexts. (We note that if H is target collision-resistant such ciphertexts are generated the first time with overwhelming probability.)
- $\mathcal{A}$ submits the ciphertexts ψ_1, ψ_2 to the decapsulation oracle and gets back two corresponding key values v_i=$(u_1^{w_i})^{x_1+y_1\alpha_i} \cdot (u_2^{w_i})^{x_2+y_2\alpha_i}$. Since $\psi_i \neq \psi$, the decapsulation oracle does not reject the queries.
- $\mathcal{A}$ computes $w_i^{-1} \bmod q$ and $T_i = v_i^{w_i^{-1}}$ for $i = 1, 2$. If $\alpha = \alpha_i$ then $\mathcal{A}$ returns T_i. Otherwise, $\mathcal{A}$ computes and returns $T_2 \cdot (T_2/T_1)^{(\alpha_2-\alpha_1)^{-1}(\alpha-\alpha_2)}$.

We analyze the above attack algorithm. For $i = 1, 2$, we have

$$\begin{aligned} T_i &= v_i^{w_i^{-1}} \\ &= ((u_1^{w_i})^{x_1+y_1\alpha_i} \cdot (u_2^{w_i})^{x_2+y_2\alpha_i})^{w_i^{-1}} \\ &= u_1^{x_1+y_1\alpha_i} \cdot u_2^{x_2+y_2\alpha_i} \\ &= (g_1^{x_1} g_2^{x_2})^r \cdot (g_1^{y_1} g_2^{y_2})^{r\alpha_i} = c^r d^{r\alpha_i} \ . \end{aligned}$$

If $\alpha = \alpha_i$ then T_i is obviously equal to the real key v. Otherwise, we have

$$T_2 \cdot (T_2/T_1)^{(\alpha_2-\alpha_1)^{-1}(\alpha-\alpha_2)} = c^r d^{r\alpha_2} \cdot (c^r d^{r\alpha_2}/c^r d^{r\alpha_1})^{(\alpha_2-\alpha_1)^{-1}(\alpha-\alpha_2)}$$
$$= c^r d^{r\alpha} ,$$

which is equal to the real key v. $\square$

By extending the above attack idea we can show that the HPS-based Kurosawa-Desmedt KEM (described in Section 3.2) its variant based on is not CCA2-secure, that is, it completely reveals a plaintext (i.e., a key) under adaptive chosen ciphertext attack. In this case, to mount such an attack, we need more elaborated techniques to construct a set of $n+1$ independent equations for $n+1$ variables.

Lemma 2. The HPS-based Kurosawa-Desmedt KEM is not CCA2-secure. In particular, we present an efficient adversary that reconstructs the real key in the adaptive chosen ciphertext attack game making $n+1$ queries to the decapsulation oracle.

Proof. Let $(u_1 = g_1^r, u_2 = g_2^r)$ be the challenge given to the adversary $\mathcal{A}$. Now $\mathcal{A}$ uses the following algorithm to reconstruct the real key corresponding to the challenge ciphertext.

- First $\mathcal{A}$ chooses distinct $w_i \in \mathbb{Z}_q^* \backslash \{1\}$ for $i = 1, \ldots, n+1$. Using the challenge ciphertext $\psi=(u_1, u_2)$, $\mathcal{A}$ computes n ciphertexts $\psi_i=(u_1^{w_i} = g_1^{rw_i}, u_2^{w_i} = g_2^{rw_i})$ satisfying

$$M = \begin{pmatrix} 1 & a_{11} & \ldots & a_{1j} & \ldots & a_{1n} \\ \vdots & \vdots & \ddots & & & \vdots \\ 1 & a_{i1} & & a_{ij} & & a_{in} \\ \vdots & \vdots & & & \ddots & \vdots \\ 1 & a_{(n+1)1} & \ldots & a_{(n+1)j} & \ldots & a_{(n+1)n} \end{pmatrix}, \quad |M| \neq 0 . \tag{1}$$

  where $(a_{i1}, \ldots, a_{in}) = \Gamma(u_1^{w_i}, u_2^{w_i})$ for $i = 1, \ldots, n+1$. If the condition does not hold, $\mathcal{A}$ repeats the previous step to obtain such ciphertexts. (We assume that if input values to Γ are random then such ciphertexts are generated with meaningful probability.)

- $\mathcal{A}$ submits the ciphertexts ψ_i to a decapsulation oracle and gets back its corresponding key values $v_i=(u_1^{w_i})^{i_0+a_{i1}i_1+\ldots+a_{in}i_n} \cdot (u_2^{w_i})^{j_0+a_{i1}j_1+\ldots+a_{jn}j_n}$. Since $\psi_i \neq \psi$ the decapsulation queries are valid in the chosen ciphertext attack game.
- $\mathcal{A}$ computes $w_i^{-1} \bmod Q$ and $T_i=v_i^{w_i^{-1}}$ for $i = 1, \ldots, n+1$. Then

$$T_i = v_i^{w_i^{-1}} = ((u_1^{w_i})^{i_0+a_{i1}i_1+\ldots+a_{in}i_n} \cdot (u_2^{w_i})^{j_0+a_{i1}j_1+\ldots+a_{jn}j_n})^{w_i^{-1}}$$
$$= u_1^{i_0+a_{i1}i_1+\ldots+a_{in}i_n} \cdot u_2^{j_0+a_{i1}j_1+\ldots+a_{jn}j_n}$$

$$= s_0^r (s_1^r)^{a_{i1}} \ldots (s_n^r)^{a_{in}} \text{ (in a projective hash system).}$$

- The goal of $\mathcal{A}$ is to compute the original key

$$v = s_0^r (s_1^r)^{a_1} \ldots (s_n^r)^{a_n},$$

where $\Gamma(u_1, u_2)=(a_1, \ldots, a_n)$. We note that, if $\mathcal{A}$ could compute all s_i^r then $\mathcal{A}$ could compute v by using the public information $\Gamma(u_1, u_2)=(a_1, \ldots, a_n)$. Next we shall show that $\mathcal{A}$ can compute all s_i^r. Conceptually $\mathcal{A}$ has $n+1$ independent equations for $n+1$ variables $z_i = \log s_i^r$ as follows:

$$\log T_i = z_1 + a_{i1} z_1 + \ldots + a_{in} z_n.$$

By assumption the determinant $|M|$ of the coefficient matrix M of this system of equations is not zero. Using *Gaussian Elimination method*, $\mathcal{A}$ transforms the matrix M to $(n+1) \times (n+1)$ identity matrix. At the same time, the adversary $\mathcal{A}$ applies the "same" operation steps to the values $T_i = s_0^r (s_1^r)^{a_{i1}} \ldots (s_n^r)^{a_{in}}$, i.e., replacing addition and multiplication with multiplication and exponentiation, respectively. If the first row of M is transformed to $(1, 0, \ldots, 0)$ then $\mathcal{A}$ obtains s_0^r. Similarly, whenever the i-th row of M is transformed to $(0, \ldots, i, \ldots, 0)$ the adversary $\mathcal{A}$ obtains s_i^r. Hence $\mathcal{A}$ knows all the s_i^r values and can therefore compute v.

5 Discussion

Remark. In fact our attack already breaks chosen-plaintext (CPA) non-malleability of the Kurosawa-Desmedt KEM, i.e., the KEM is not NM-CPA [9,4]. In a non-malleability attack an adversary is considered to be successful if it can come up with a vector of ciphertexts such that the respective decapsulated session keys of those ciphertexts are *meaningfully related* to the (unknown) key of the challenge ciphertext. In the attack, given the challenge ciphertext ψ the adversary simply outputs the ciphertexts ψ_1 and ψ_2 and defines the following relation $R(v, v_1, v_2)$ over the respective (hidden) keys v, v_1, v_2:

$$v = T_2 \cdot (T_2/T_1)^{(\alpha_2-\alpha_1)^{-1}(\alpha-\alpha_2)} \quad \text{where } T_i = v_i^{w_i^{-1}} .$$

Note that this is a chosen-plaintext attack since the adversary never queries the decapsulation 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. Our attacks are also successful against a variant of the Kurosawa-Desmedt KEM where ciphertexts are checked for consistency in the decapsulation algorithm, i.e., it is checked if $\log_{g_1} u_1 = \log_{g_2} u_2$. Such a check can be

implemented by verifying if $u_1^{\omega} = u_2$, where $\omega = \log_{g_1} g_2$ which can be made part of sk. In our attack the queried ciphertexts are obviously consistent.

Remark. In the Kurosawa-Desmedt hybrid encryption scheme the symmetric key v is additionally hashed using a key-derivation function $K : G \to \{0,1\}^k$. We now show that even if one considers this hash function as part of the Kurosawa-Desmedt KEM then our attack may still apply (depending on the concrete hash function used). The point is that in the security requirements of [14] the key-derivation function K only has to satisfy relatively weak security properties, namely $K(v)$ has to be uniformly distributed over $\{0,1\}^k$ given that v is uniformly distributed over G. In particular, a hash function that is efficiently invertible may satisfy this property. In that case the attacker can reconstruct v from $K(v)$ and run the attack as described above. Concretely, in certain cryptographically relevant elliptic curve groups a representation of an element in G requires $k = 2\lambda$ bits, where λ is the security parameter. In that case the identity function $K : G \to \{0,1\}^k$ fulfills the security requirements of [14] and is clearly invertible. Therefore, the minimum requirements of the key-derivation function from [14] cannot protect against our proposed attack.

Surely, if we model K as a random oracle [5] 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 K and the group G) then the hashed version of the Kurosawa-Desmedt KEM indeed can be proved chosen-ciphertext secure.

Remark. In [12] it was shown that the Kurosawa-Desmedt KEM is *CCCA* secure. Furthermore, in [2] it was shown that the Kurosawa-Desmedt KEM is *LCCA* secure with respect to a certain predicate for which *ciphertext soundness* holds. Both security notions are weaker than CCA2 security, but this does not imply that the Kurosawa-Desmedt KEM is not CCA2-secure.

Remark. Hofheinz and Kiltz [12] proposed a "dual version" of the Kurosawa-Desmedt KEM which, as a hybrid encryption scheme, can also be proved CCA2-secure. We remark that our attack does not carry over to show that their KEM is not CCA2-secure. Indeed, it was shown in [13] that the KEM part is indistinguishable under CCA2 attacks under the non-standard *Gap Hashed Diffie-Hellman* assumption. (Or, one-way under CCA2 attacks under the Gap Diffie-Hellman assumption.) It remains an open problem to prove CCA2-security of the dual KD-KEM from [12] under the DDH assumption, or to show the impossibility of such a proof.

6 Conclusion

We showed that the KEM part of the Kurosawa-Desmedt hybrid encryption scheme [14] is not CCA2-secure, that is, not secure against adaptively chosen ciphertext attack.

References

[1] M. Abdalla, M. Bellare and P. Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. Proc. of CT-RSA'01, LNCS 2020, Springer-Verlag, pp. 143-158, 2001.

[2] M. Abe, R. Gennaro and K. Kurosawa. Tag-KEM/DEM : a new framework for hybrid encryption. Cryptology ePrint Archive, Report 2005/027, 2005. http://eprint.iacr.org/.

[3] M. Abe, R. Gennaro, K. Kurosawa and V. Shoup. Tag-KEM/DEM: a new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM. Proc. of Eurocrypt'05, LNCS 3494, Springer-Verlag, pp. 128-146, 2005.

[4] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. Relations among notions of security for public-key encryption schemes. Proc. of CRYPTO'98, LNCS 1462, Springer-Verlag, pp. 26-45, 1998.

[5] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. Proc of ACM CCS'93, ACM Press, pp. 62-73, 1993.

[6] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against chosen ciphertext attack. Proc. of CRYPTO'98, LNCS 1462, Springer-Verlag, pp. 13-25, 1998.

[7] R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. Proc. of Eurocrypt'02, LNCS 2332, Springer-Verlag, pp. 45-64, 2002.

[8] 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), pp. 167-226, 2003.

[9] D. Dolev, C. Dwork and M. Naor. Non-malleable cryptography, SIAM Journal on Computing, 30(2), pp. 391-437, 2000.

[10] S. Goldwasser and S. Micali. Probabilistic encryption. J. Computer System Sciences 28(2), pp. 270 -299, 1984

[11] R. Gennaro and V. Shoup. A note on an encryption scheme of Kurosawa and Desmedt. Cryptology ePrint Archive, Report 2004/194, 2005. http://eprint.iacr.org/.

[12] D. Hofheinz and E. Kiltz. Secure hybrid encryption from weakened key encapsulation. Proc. of CRYPTO'07, LNCS 4622, Springer-Verlag, pp. 553–571, 2007.

[13] E. Kiltz. Chosen-ciphertext secure key-encapsulation based on Gap Hashed Diffie-Hellman. Proc. of PKC'07, LNCS 4450, Springer-Verlag, pp. 282–297, 2007.

[14] K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. Proc. of CRYPTO'04, LNCS 3152, Springer-Verlag, pp. 426-442, 2004.

[15] C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. Proc. of CRYPTO'91, LNCS 576, Springer-Verlag, pp. 433-444, 1992.

[16] V. Shoup. Using hash functions as a hedge against chosen ciphertext attack. Proc. of Eurocrypt'00, LNCS 1807, Springer-Verlag, pp. 275-278, 2000.