On a Variation of Kurosawa-Desmedt Encryption Scheme

Le Trieu Phong and Wakaha Ogata

January 26, 2006

Abstract

Kurosawa-Desmedt encryption scheme is a variation of Cramer-Shoup encryption schemes, which are the first practical schemes secure against adaptive chosen ciphertext attack in standard model. We introduce a variant of Kurosawa-Desmedt encryption scheme, which is not only secure against adaptive chosen ciphertext attack but also slightly more efficient than the original version.

1 Introduction

1.1 Background

The notion of *chosen-ciphertext security* was introduced by Naor and Yung [NY90] and developed by Rackoff and Simon [RS91], and Dolev, Dwork, and Naor [DDN91]. This notion is now largely considered as the "right" notion for encryption schemes.

In the random oracle model, many practical schemes secure against adaptive chosen-ciphertext attack (IND-CCA) have been proposed: OAEP+ [Sh01], SAEP [Bo01], RSA-OAEP [FOPS01] to name just a few. Although the security analysis in the random oracle model gives us a strong evidence that the schemes are secure, it does not rule out all possible attacks.

In standard model, the first practical public key cryptosystem which is provably IND-CCA secure was discovered by Cramer and Shoup [CS98]. The security of the scheme is based on the hardness of the decisional Diffie-Hellman problem. After that, in [Sh00], Shoup presented a hybrid variant of the Cramer-Shoup cryptosystem. As a hybrid scheme, the variant is very flexible since messages can be arbitrary bit strings.

In [KD04], Kurosawa and Desmedt modified the hybrid scheme presented in [Sh00], gaining a scheme which produces shorter ciphertexts and needs less exponentiations than the original one. However, their proof of security relied on the use of information theoretically secure functions KDF(key derivation function) and MAC(message authentication code), which makes the Kurosawa-Desmedt scheme as efficient as the original Cramer-Shoup for typical security parameters, as recently stated in [GS05]. In that paper, Gennaro and Shoup also presented a different proof of security for Kurosawa-Desmedt scheme, which showed that the scheme can be instantiated with any computationally secure KDF and MAC, thus extending its applicability and efficiency.

Gennaro and Shoup also raise an open question about the optimizations of the Kurosawa-Desmedt scheme in [GS05]: Can the scheme be optimized? This paper is an answer to that question.

1.2 Our Contribution

We present a variant of the Kurosawa-Desmedt scheme, which is also IND-CCA secure as the original one. Furthermore, the variant owns the following properties:

- It is optimized in the sense that all exponentiations in the decryption algorithm are with respect to the same base, hence the algorithm can be executed faster.
- It can be also instantiated with any computationally secure KDF and MAC.
- The key generation algorithm is faster, and the private key is also shorter.

2 Security against Adaptive Chosen Ciphertext Attack

For the readers' convenience, we recall here the definition of security against adaptive chosen ciphertext attack. The definition used here is essentially the same as the one in [Sh00]. The attack scenario has fourth stages as follows.

First, the key generation algorithm is run, generating the public key and private key for the cryptosystem. The adversary, of course, obtains the public key, but not the private key.

Second, the adversary makes a series of arbitrary queries to a *decryption oracle*. Each query is a ciphertext C that is decrypted by the decryption oracle, making use of the private key of the cryptosystem. The resulting decryption is given to the adversary. The adversary is free to construct the ciphertexts in an arbitrary way, namely it is *not* required to compute them using the encryption algorithm.

Third, the adversary prepares two messages m_0 and m_1 and gives these to an *encryption oracle*. The encryption oracle chooses $b \in \{0, 1\}$ at random, encrypts m_b, and gives the resulting "target" ciphertext C^* to the adversary. The adversary is free to choose m_0 and m_1 in an arbitrary way, except that if message lengths are not fixed by the cryptosystem, then these two messages must nevertheless be of the same length.

Fourth, the adversary continues to submit ciphertexts C to the decryption oracle, subject only to the restriction that $C \neq C^*$.

Just before the adversary terminates, it outputs $\hat{b} \in \{0, 1\}$, representing its "guess" of b.

The adversary's advantage in this attack scenario is defined to be the distance from 1/2 of the probability that $\hat{b} = b$.

A cryptosystem is defined to be *secure against adaptive chosen ciphertext attack* if for any efficient adversary, its advantage is negligible.

3 Kurosawa-Desmedt Scheme

3.1 Description

Kurosawa-Desmedt scheme, called KD scheme as a notational convention, was first described in [KD04]. The description used here is from [GS05]. The scheme makes use of:

- a group G of prime order q, with (random) generators g_1 and g_2. *Security assumption (DDH): Hard to distinguish (g_1^r, g_2^r) from $(g_1^r, g_2^{r'})$, where r is a random element of Z_q and r' is a random element of $Z_q \setminus \{r\}$.*

- a message authentication code MAC, which is a function that takes two inputs, a key k and message $e \in \{0,1\}^*$, and produces a "tag" $t := MAC_k(e)$.
  Security assumption: For random k, after obtaining $t^* := MAC_k(e^*)$ for (at most one) adversarially chosen e^*, it is infeasible for an adversary A_{mac} to compute a forgery pair, i.e., a pair (e, t) such that $e \neq e^*$ and $t = MAC_k(e)$.

  Define $Adv_{MAC}(A_{mac}) = \Pr(A_{mac}$ succeeds$)$. The assumption ensures that $Adv_{MAC}(A_{mac})$ is negligible for all polynomial-time adversary A_{mac}.

- a symmetric key encryption scheme, with encryption algorithm E and decryption algorithm D, such that for key K and plaintext $m \in \{0,1\}^*$, $e := E_K(m)$ is the encryption of m under K, and for key K and ciphertext $e \in \{0,1\}^*$, $m := D_K(e)$ is the decryption of e under K.
  Security assumption (semantic security): hard to distinguish $E_K(m_0)$ and $E_K(m_1)$ for randomly chosen K and adversarially chosen m_0 and m_1, where m_0 and m_1 are of equal length.

- a key derivation function KDF, such that for $v \in G$, $KDF(v) = (k, K)$, where k is a message authentication key, and K is a symmetric encryption key.
  Security assumption: hard to distinguish $KDF(v)$ and (k, K), where v, k and K are random.

  Let A_{kdf} be an 0-or-1-output algorithm that takes as input a pair of message authentication key and symmetric encryption key. Define

  $$\begin{aligned} Adv_{KDF}(A_{kdf}) &= \Pr[A_{kdf}(KDF(v)) \to 1] \\ &\quad - \Pr[A_{kdf}(k, K) \to 1]. \end{aligned}$$

  The assumption ensures that $Adv_{KDF}(A_{kdf})$ is negligible for all polynomial-time adversary A_{kdf}.

- a hash function $H : G \times G \to Z_q$.
  Security assumption (target collision resistance): given $u_1^* := g_1^r$ and $u_2^* := g_2^r$ for random $r \in Z_q$, hard to find $(u_1, u_2) \in G \times G \setminus \{(u_1^*, u_2^*)\}$ such that $H(u_1, u_2) = H(u_1^*, u_2^*)$.

Note that the key space for the message authentication code is assumed to consist of all bit strings of a given length, so that by a random key k, we mean a random bit string of appropriate length. Similarly for the symmetric encryption keys.

Note also that KDF and H may have associated keys, which are publicly known.

Key Generation: The description of the group G is generated, along with random generators g_1 and g_2 for G. Any keys for KDF and H are also generated. Then,

$$x_1, x_2, y_1, y_2 \stackrel{\$}{\leftarrow} Z_q, c \leftarrow g_1^{x_1} g_2^{x_2}, d \leftarrow g_1^{y_1} g_2^{y_2}.$$

The public key consists of the description of G, the generators g_1 and g_2, keys for KDF and H (if any), along with the group elements c and d. The private key consists of the public key, along with x_1, x_2, y_1, y_2.

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

$r \stackrel{\$}{\leftarrow} Z_q$, $u_1 \leftarrow g_1^r \in G$, $u_2 \leftarrow g_2^r \in G$,
$\alpha \leftarrow H(u_1, u_2) \in Z_q$,
$v \leftarrow c^r d^{r\alpha} \in G$, $(k, K) \leftarrow KDF(v)$,
$e \leftarrow E_K(m)$, $t \leftarrow MAC_k(e)$
output $C := (u_1, u_2, e, t)$

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

$\alpha \leftarrow H(u_1, u_2)$, $v \leftarrow u_1^{x_1+y_1\alpha} u_2^{x_2+y_2\alpha} \in G$
$(k, K) \leftarrow KDF(v)$
If $t \neq MAC_k(e)$ then output `reject`
Else
  $m \leftarrow D_K(e)$
  output m

3.2 Security of Kurosawa-Desmedt Scheme

In [KD04], Kurosawa and Desmedt proved that their scheme is secure against adaptive chosen ciphertext attack using some additional assumptions on KDF and MAC besides the assumptions described in the previous section. Recently, Gennaro and Shoup [GS05] showed that the scheme are still secure against the attack without the additional assumptions on KDF and MAC. More precisely,

Theorem 1 ([GS05]). *KD scheme is secure against adaptive chosen ciphertext attack if the assumptions on its components described in the previous section hold.*

4 Proposed Scheme

4.1 Description

The proposed scheme, called KD1 as a notational convention, makes use of the same components and security assumptions on them as KD scheme. Furthermore, the public key and the encryption algorithm of both schemes are identical. However, the key generation algorithms and decryption algorithms are different.

Key Generation: The description of the group G is generated, along with random generators g_1 for G. Any keys for KDF and H are also generated. Then,

$$\omega \xleftarrow{\$} Z_q^*, g_2 \leftarrow g_1^{\omega}$$

$$x, y \xleftarrow{\$} Z_q, c \leftarrow g_1^x, d \leftarrow g_1^y.$$

The public key consists of the description of G, the generators g_1 and g_2, keys for KDF and H (if any), along with the group elements c and d. The private key consists of the public key, along with ω, x, y.

Decryption of $C = (u_1, u_2, e, t)$:
$\alpha \leftarrow H(u_1, u_2), v \leftarrow u_1^{x+y\alpha} \in G$, $(k, K) \leftarrow KDF(v)$
If $u_2 \neq u_1^{\omega}$ or $t \neq MAC_k(e)$ then output `reject`
Else
$m \leftarrow D_K(e)$
output m

4.2 Security

Theorem 2. *KD1 is secure against adaptive chosen ciphertext attack if the DDH assumption holds for G, and the components MAC(message authentication code), E(symmetric encryption scheme), KDF (key derivation function), H(hash function) are secure.*
In particular, for all probabilistic, polynomial-time adversary A, there exist algorithms A_1 and A_2 whose resources are essentially the same as those of A such that

$$|Adv_{KD1}^{cca}(A) - Adv_{KD}^{cca}(A)| \leq Q_A(Adv_{KDF}(A_1) + Adv_{MAC}(A_2)),$$

where Q_A is the number of decryption queries made by A.

Proof. Consider a probabilistic, polynomial-time adversary A. We will begin with a game normally used to define CCA security of KD scheme.

Game 0 (attack game on KD scheme)

This game is an interactive computation between A and a simulator. Initially, the simulator runs the key generation algorithm of KD scheme, obtaining the description of G, generators g_1, g_2, keys for KDF and H (if any), along with the values $x_1, x_2, y_1, y_2 \in Z_q$ and $c, d \in G$. The simulator gives the public key to A.

During the execution of the game, the adversary A makes a number of "decryption requests." Assume these requests are $C^{(1)}, \ldots, C^{(Q_A)}$, 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 A the result. We denote by $\alpha^{(i)}$, $v^{(i)}$, $k^{(i)}$, $K^{(i)}$ the corresponding intermediate quantities computed by the decryption algorithm on input $C^{(i)}$.

The adversary may also make a single "challenge request." For such a request, the adversary submits two messages m_0, m_1 of equal length bit strings to the simulator; the simulator chooses $b \in \{0, 1\}$ at random, and encrypts m_b using the encryption algorithm of KD scheme, obtaining the "target ciphertext" $C^* = (u_1^*, u_2^*, e^*, t^*)$.

The only restriction on the adversary's requests is that after the "challenge request", subsequent decryption requests must not be the same as the target ciphertext.

At the end of the game, the adversary outputs $\hat{b} \in \{0, 1\}$.

Let T_0 be the event that $\hat{b} = b$. Advantage of the adversary with respect to KD scheme is defined as

$$Adv_{KD}^{cca}(A) = |\Pr[T_0] - \frac{1}{2}|.$$

We will consider other games, which are slightly-modified versions of Game 0. The games will be built in the order as follows:

$$\begin{tikzcd}
 & G_1 & & & \\
G_0 \arrow[ur] \arrow[dr] & & & & \\
 & G_{1'} \arrow[r] & G_{2'} \arrow[r] & G_{3'} \arrow[r] & G_{4'}
\end{tikzcd}$$

All those games are viewed as operating on the same underlying probability space, i.e., all the random variables $\texttt{Coins}$(of A), ω (in Game 1), x_1, x_2, y_1, y_2, r^* (for encrypting m_b), b take the same value in those games.

Game 1 This game is the same as Game 0, except that

- instead of being randomly chosen from G, the generator g_2 is generated as follows

$$\omega \xleftarrow{\$} Z_q, g_2 \leftarrow g_1^{\omega}.$$

- the simulator not only chooses x_1, x_2, y_1, y_2 randomly from Z_q but also puts $x := x_1 + \omega x_2$ and $y := y_1 + \omega y_2$. The values c and d are now computed as

$$c \leftarrow g_1^x, d \leftarrow g_1^y.$$

- in processing a decryption request $C = (u_1, u_2, e, t)$, the simulator proceeds as follows
  - $\alpha \leftarrow H(u_1, u_2), v \leftarrow u_1^{x+\alpha y}$
  - $(k, K) \leftarrow KDF(v)$
  - Test if $u_2 = u_1^{\omega}$ and $t = MAC_k(e)$; if this is not the case, then output `reject` and halt.
  - Output $m \leftarrow D_K(e)$

It is obvious that Game 1 is really the attack game of A against KD1. Let T_1 be the event that $\hat{b} = b$ in this game. Thus the advantage of A with respect to KD1 scheme is

$$Adv_{KD1}^{cca}(A) = |\Pr[T_1] - \frac{1}{2}|.$$

Therefore, in order to bound $|Adv_{KD1}^{cca}(A) - Adv_{KD}^{cca}(A)|$, it is sufficient to bound $|\Pr[T_1] - \Pr[T_0]|$.

Let F_1 be the event that some ciphertext is rejected in Game 1, but would have passed the test of the decryption algorithm in Game 0. Game 0 and Game 1 are then the same until F_1 occurs. Thus $T_1 \wedge \overline{F_1}$ and $T_0 \wedge \overline{F_1}$ are identical. By the Difference Lemma[1] (see [Sh05]),

$$|\Pr[T_1] - \Pr[T_0]| \leq \Pr[F_1].$$

We will use the below games $G_{1'}$, $G_{2'}$, $G_{3'}$, $G_{4'}$ to bound $\Pr[F_1]$.

[1]This lemma is called "Fundamental Lemma of Game-Playing" in [BR05]

Game $1'$ This game is the same as Game 0, except that

- instead of being randomly chosen from G, g_2 is generated by

$$\omega \xleftarrow{\$} Z_q, g_2 \leftarrow g_1^{\omega}.$$

- in processing a decryption request $C = (u_1, u_2, e, t)$, the simulator proceeds as follows
  - $\alpha \leftarrow H(u_1, u_2), v \leftarrow u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$
  - $(k, K) \leftarrow KDF(v)$
  - Test if $u_2 = u_1^{\omega}$ and $t = MAC_k(e)$; if this is not the case, then output `reject` and halt.
  - Output $m \leftarrow D_K(e)$

Note that Game 1 and Game $1'$ are identical from the viewpoint of the adversary A. In fact, if A submits a ciphertext $C = (u_1, u_2, e, t)$ with $u_2 \neq u_1^{\omega}$, then A will receive `reject` in the both games. On the other hand, if $u_2 = u_1^{\omega}$, then the value v in Game 1 and Game $1'$ is the same, which ensures that the ciphertext is identically decrypted in those games. Thus

$$\Pr[F_1] = \Pr[F_{1'}],$$

where $F_{1'}$ be the event that some ciphertext is rejected in Game $1'$, but would have passed Game 0.

Let $F_{1'}^{(j)}$ be the event that the jth ciphertext $C^{(j)}$ is rejected in Game $1'$, but would have passed the test in Game 0. Then

$$\Pr[F_{1'}] \leq Q_A \max_{1 \leq j \leq Q_A} \{\Pr[F_{1'}^{(j)}]\}.$$

Note that $F_{1'}^{(j)}$ occurs if and only if $u_2^{(j)} \neq (u_1^{(j)})^{\omega}$ and $t^{(j)} = MAC_{k^{(j)}}(e^{(j)})$, where $k^{(j)}$ is the first part of $KDF((u_1^{(j)})^{x_1+y_1\alpha^{(j)}} (u_2^{(j)})^{x_2+y_2\alpha^{(j)}})$. Our task now is to bound $\Pr[F_{1'}^{(j)}]$.

Game $2'$ This game is the same as Game $1'$, except that the simulator now proceeds a decryption request $C = (u_1, u_2, e, t)$ as follows

$D01'$:	$\alpha \leftarrow H(u_1, u_2)$
$D02'$:	If $u_2 \neq u_1^{\omega}$ then
$D03'$:	$\quad v \leftarrow u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$
$D04'$:	$\quad (k, K) \leftarrow KDF(v)$

$D05^{'}$:	Test if $t = MAC_k(e)$; if this is not the case, then output `reject` and halt.
$D06^{'}$:	$m \leftarrow D_K(e)$. Output `reject`
$D07^{'}$:	Else
$D08^{'}$:	$v \leftarrow u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$
$D09^{'}$:	$(k, K) \leftarrow KDF(v)$
$D10^{'}$:	Test if $t = MAC_k(e)$; if this is not the case, then output `reject` and halt.
$D11^{'}$:	$m \leftarrow D_K(e)$. Output m.

The change, which is purely conceptual, is that the simulator now considers two cases, $u_2 \neq u_1^{\omega}$ and $u_2 = u_1^{\omega}$ in decryption. Note that whenever line $D03^{'}$ is reached, then the output is always `reject` . Moreover, $\Pr[F_{1'}^{(j)}] = \Pr[F_{2'}^{(j)}]$, where $F_{2'}^{(j)}$ is the event that line $D06^{'}$ is executed in the jth decryption request.

Game $3^{'}$ This game is the same as Game $2^{'}$, except that we change line $D03^{'}$ as follows

$D03^{'}$: $v \xleftarrow{\$} G$

Let $F_{3'}^{(j)}$ be the event that line $D06^{'}$ is executed in the jth decryption request in Game $3^{'}$. We claim that $F_{3'}^{(j)} = F_{2'}^{(j)}$. This follows from the fact that $v = u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$, $c = g_1^{x_1+\omega x_2}$, and $d = g_1^{y_1+\omega y_2}$ are mutually independent and uniformly distributed over G if $u_2 \neq u_1^{\omega}$. In fact, if we put

$$\begin{aligned} r_1 &:= \log_{g_1} u_1, \\ r_2 &:= \log_{g_2} u_2, \end{aligned}$$

then the condition $u_2 \neq u_1^{\omega}$ implies $r_1 \neq r_2$, so the following values

$$\begin{aligned} \log_{g_1} c &= x_1 + \omega x_2, \\ \log_{g_1} d &= y_1 + \omega y_2, \\ \log_{g_1} v &= r_1(x_1 + \alpha y_1) + r_2(x_2 + \alpha y_2), \end{aligned}$$

are linearly independent. This means that v can take any value over G. Thus Game $3^{'}$ and Game $2^{'}$ are identical, and hence $F_{3'}^{(j)} = F_{2'}^{(j)}$.

Game $4^{'}$ This game is the same as Game $3^{'}$, except that we change line $D04^{'}$ as follows
$D04^{'}$: $(k, K) \xleftarrow{\$}$ "$KeySpace$"

Let $F_{4^{'}}^{(j)}$ be the event that line $D06^{'}$ is executed in the jth decryption request in Game $4^{'}$. It is clear that we can build an algorithm A_1, using similar resources to those of A, such that

$$|\Pr[F_{3^{'}}^{(j)}] - \Pr[F_{4^{'}}^{(j)}]| \leq Adv_{KDF}(A_1).$$

Note that the key k of the message authentication code in this game is completely random and is not used anywhere except as input for MAC. Thus, the probability that line $D06^{'}$ is executed in the jth decryption request must be less than the probability that an algorithm A_2 can break the MAC, i.e.,

$$\Pr[F_{4^{'}}^{(j)}] \leq Adv_{MAC}(A_2).$$

In fact, A_2 just employs A and its simulator, and returns (e, t) at $D05^{'}$ in Game $4^{'}$ whenever line $D06^{'}$ is executed. Summing up,

$$\begin{aligned}
\Pr[F_1] &= \Pr[F_{1^{'}}] \\
&\leq Q_A \max_{1 \leq j \leq Q_A} \{\Pr[F_{1^{'}}^{(j)}]\} \\
&= Q_A \max_{1 \leq j \leq Q_A} \{\Pr[F_{2^{'}}^{(j)}]\} \\
&= Q_A \max_{1 \leq j \leq Q_A} \{\Pr[F_{3^{'}}^{(j)}]\} \\
&\leq Q_A(Adv_{KDF}(A_1) + \max_{1 \leq j \leq Q_A} \{\Pr[F_{4^{'}}^{(j)}]\}) \\
&\leq Q_A(Adv_{KDF}(A_1) + Adv_{MAC}(A_2)),
\end{aligned}$$

which completes the proof.

References

[Bo01] D. Boneh. Simplified OAEP for the RSA and Rabin Functions. CRYPTO 2001, pages 275-291, 2001

[BR05] M. Bellare and P. Rogaway. The Game-Playing Technique and its Applications to Triple Encryption. Draft 2.0, 2005. Available at http://eprint.iacr.org/2004/331.pdf

[CS98] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. CRYPTO'98, pages 13-25, 1998

[DDN91] D. Dolev, C.Dwork, and M. Naor. Non-malleable cryptography. STOC'91, pages 542-552, 1991

[FOPS01] E. Fujisaki, T. Okamoto, D. Pointcheval, J. Stern. RSA-OAEP Is Secure under the RSA Assumption. CRYPTO 2001, pages 260-274, 2001

[GS05] R. Gennaro and V. Shoup. A Note on an Encryption Scheme of Kurosawa and Desmedt. manuscript, 2005. Available at http://www.shoup.net

[KD04] K.Kurosawa and Y.Desmedt. A New Paradigm of Hybrid Encryption Scheme. CRYPTO'04, pages 426-442, 2004

[NY90] M. Naor and M.Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In STOC'90, 1990

[RS91] C. Rackoff and D. Simon. Noninteractive zero-knowledge proof of knowledge and chosen ciphertext attack. In CRYPTO'91, pages 433-444, 1991

[Sh00] V. Shoup. Using Hash Function as a Hedge Against Chosen Ciphertext Attack. In EuroCrypt'00, pages 275-288, 2000

[Sh01] V. Shoup. OAEP Reconsidered. CRYPTO 2001, pp. 239-259, 2001

[Sh05] V. Shoup. Sequences of Games: a Tool for Taming Complexity in Security Proofs. manuscript, 2005. Available at www.shoup.net