How to Encrypt Long Messages without Large Size Symmetric/Asymmetric Encryption Schemes

Masashi Mitomo Kaoru Kurosawa

Department of Communication and Integrated Systems,
Tokyo Institute of Technology
2–12–1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan
mitomo@flab.fujitsu.co.jp, kurosawa@ss.titech.ac.jp

Abstract. Suppose that we wish to encrypt long messages with small overhead by a public key encryption scheme which is secure against adaptive chosen ciphertext attack (IND-CCA2). Then the previous schemes require either a large size one-way trapdoor permutation (OAEP) or both a large size symmetric encryption scheme and a small size asymmetric encryption scheme (hybrid encryption). In this paper, we show a scheme which requires only a small size asymmetric encryption scheme satisfying IND-CCA2 for our purpose. Therefore, the proposed scheme is very efficient. A hash function and a psuedorandom bit generator are used as random oracles.

Keywords: public key, chosen ciphertext attack, provable security, long message, efficient encryption/decryption.

1 Introduction

Suppose that we wish to encrypt long messages with small overhead by a public key encryption scheme which is secure against adaptive chosen ciphertext attack (IND-CCA2).

Bellare and Rogaway showed how to design a scheme satisfying IND-CCA2 from any one-way trapdoor permutation [3]. The scheme is called OAEP and it uses a hash function and a psuedorandom bit generator as random oracles. For our purpose, however, this scheme requires a large size one-way trapdoor permutation because we wish to encrypt long messages with small overhead.

On the other hand, a hybrid encryption scheme uses both a large size symmetric encryption scheme and a small size asymmetric encryption scheme. The small size asymmetric encryption scheme is used to send a secret key of the large size symmetric encryption scheme and a long message is encrypted by the large size symmetric encryption scheme.

Cramer and Shoup showed a public key encryption scheme which is secure in the sense of IND-CCA2 in the standard model by assuming the decision Diffie-Hellman assumption [4]. They briefly mentioned in their work that their scheme can be applied to hybrid usage with a secure symmetric key encryption scheme [4].

Next, Abdalla, Bellre, and Rogaway presented a more efficient hybrid encryption scheme, called DHAES, and prove that hybrid usage is secure in the IND-CCA2 sense in the random oracle model (or a strong assumption in the standard (not random oracle) model) [1]. Their scheme depends on the Diffie-Hellman key-distribution scheme.

Finally, Fujisaki and Okamoto showed a generic method to convert a secure public key encryption scheme and a secure symmetric key encryption scheme into a hybrid encryption scheme which is secure in the sense of IND-CCA2 in the random oracle model [5]. It is just required that the underlying public encryption scheme is one-way and γ-uniform. (1) Their scheme is more generic than the previous hybrid encryption schemes in a sense that the previous ones depend on some specific cryptographic assumptions. (2) However, their scheme is not efficient in decryption. Generally, the decryption algorithm outputs the message m if and only if some validity check of a ciphertext c succeeds. However, their validity check requires reencrypting the whole message m while m is usually long in hybrid encryption schemes.

To summarize, the previous schemes require either a large size one-way trapdoor permutation (OAEP) or both a large size symmetric encryption scheme and a small size asymmetric encryption scheme (hybrid encryption) if we wish to encrypt long messages with small overhead.

This paper shows that we can encrypt long messages with small overhead by using only a small size asymmetric encryption scheme satisfying IND-CCA2. Therefore, the proposed scheme is very efficient. A hash function and a psuedorandom bit generator are used as random oracles.

Our scheme is at least as generic as Fujisaki-Okamoto scheme because we can use their scheme as the underlying asymmetric encryption scheme.

Further, the decryption algorithm does not need to check that a long message is encrypted correctly.

We first prove that the following encryption scheme is secure against non-adaptive chosen ciphertext attack (IND-CCA1) if the public encryption scheme $\mathcal{E}'_{pk}$ is secure in the sense of IND-CCA1.

$$\mathcal{E}_{pk}(m) = m \oplus G(\sigma) || \mathcal{E}'_{pk}(\sigma),$$

where m is a plaintext, σ is a random seed and G is a random bit generator. In this scheme, for checking the validity of a ciphertext $\mathcal{E}_{pk}(m)$, it is enough to check the validity of only $\mathcal{E}'_{pk}(\sigma)$.

We next prove that the following encryption scheme is secure in the sense of IND-CCA2 if the public encryption scheme $\mathcal{E}'_{pk}$ is secure in the sense of IND-CCA2.

$$\mathcal{E}_{pk}(m) = (c_1, c_2),$$

where

$$c_1 = m \oplus G(\sigma), \quad c_2 = \mathcal{E}'_{pk}(\sigma \oplus H(c_1))$$

and H is a random hash function. Note that only a small size asymmetric encryption scheme $\mathcal{E}'_{pk}$ is used. Therefore, the proposed scheme is very efficient.

In particular, for checking the validity of a ciphertext $\mathcal{E}_{pk}(m) = (c_1, c_2)$, it is enough to check the validity of only $c_2 = \mathcal{E}'_{pk}(\sigma \oplus H(c_1))$. Therefore, our decryption algorithm is efficient because m is not necessary in this check as opposed to [5].

Finally, we show that the above scheme is secure in the sense of plaintext awareness if we let $c_1 = m0^k \oplus G(\sigma)$. The notion of plaintext awareness was introduced by [3]. We follow the definition of [2].

2 Definitions of security [2]

2.1 Convention

Definition 2.1 If A is a probabilistic algorithm, then $A(x_1, ...; r)$ is the result of running A on inputs $x_1, x_2, ...$and coins r. We let $y \leftarrow A(x_1, x_2, ...)$ denote the experiment of picking r at random and letting y be $A(x_1, ...; r)$. If S is a finite set then $x \leftarrow S$ is the operation of picking an element uniformly from S. If α is neither an algorithm nor a set then $x \leftarrow \alpha$ is a simple assignment statement. We say that *y can be output by* $A(x_1, x_2, ...)$ if there is some r such that $A(x_1, ...; r) = y$.

Definition 2.2 An asymmetric (public key) encryption scheme is a triple of algorithm, $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, where

- $\mathcal{K}$, the *key genenation algorithm*, is a probabilistic algorithm that takes a security parameter $k \in \mathsf{N}$(provided in unary) and returns a pair (pk,sk) of matching public and secret keys.
- $\mathcal{E}$, the *encryption algorithm*, is a probabilistic algorithm that takes a public key pk and a message $x \in \{0,1\}^*$ to produce a ciphertext y.
- $\mathcal{D}$, the *decryption algorithm*, is a deterministic algorithm which takes a secret key sk and ciphertext y to produce either a message $x \in \{0,1\}^*$ or a special symbol $\perp$ to indicate that the ciphertext was invalid.

We require that for all (pk,sk) which can be output by $\mathcal{K}(1^k)$, for all $x \in \{0,1\}^*$, and for all y that can be output by $\mathcal{E}_{pk}(x)$, we have that $\mathcal{D}_{sk}(y) = x$. We also require that $\mathcal{K}, \mathcal{E}$ and $\mathcal{D}$ can be computed in polynomial time. As the notation indecates, the keys are indicated as subscripts to the algorithms.

Recall that a function $\epsilon : \mathsf{N} \to \mathsf{R}$ is *negligible* if for every constant $c \geq 0$ there exists an integer k_c such that $\epsilon(k) \leq k^{-c}$ for all $k \geq k_c$.

2.2 Attack model

The goal of secure encryption is to preserve the privacy of messages: an adversary should not be able to learn from a ciphertext information about its plaintext beyond the length of that plaintext. We define a version of this notion, indistinguishability of encryptions(IND).

We consider an adversary $A = (A_1, A_2)$ who runs in two stages. In the find-stage A_1 is given an encryption algorithm $\mathcal{E}$ and outputs a pair x_0, x_1 of messages. It also outputs a string c which could record, for example, its history and its inputs. Now we pick at random either x_0 or x_1 (the choice made according to a bit b) and encrypt it (under $\mathcal{E}$) to get y. In the guess-stage we provide A_2 the output x_0, x_1, c of the previous stage, and y, and we ask it to guess b. (We assume wlog that $\mathcal{E}$ is include in c so that we don't need to explicity provide it again.) Since even the algorithm which always outputs a fixed bit will be right half of the time, we measure how well A is doing by $1/2$ less than the fraction of time that A correctly predicts b. We call twice this quantity the *advantage* which A has in predicting b.

We consider three types of attacks under this setup.

In a *chosen-plaintext attack* (CPA) the adversary can encrypt plaintext of her choosing. Of course, a CPA is unavoidable in the public-key setting.

In a *non-adaptive chosen ciphertext attack* (CCA1), we give A_1 (the public key and) access to a decryption oracle, but we do not allow A_2 access to a decryption oracle.

In an *adaptive chosen ciphertext attack* (CCA2), we continue to give A_1 (the public key and) access to a decryption oracle, but also give A_2 access to the same decryption oracle, with the only restriction that she cannot query the oracle on the challenge ciphertext y. This is an extremely strong attack model.

2.3 Indistinguishabile security

Definition 2.3 [IND-CPA, IND-CCA1, IND-CCA2] Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and let $A = (A_1, A_2)$ be an adversary. For $atk \in \{cpa, cca1, cca2\}$ and $k \in \mathsf{N}$, let $\mathsf{Adv}_{A,\Pi}^{ind-atk}(k) \triangleq$
$2 \cdot \Pr\Big[(pk, sk) \leftarrow \mathcal{K}(1^k); (x_0, x_1, s) \leftarrow A_1^{\mathcal{O}_1}(pk); b \leftarrow \{0,1\}; y \leftarrow \mathcal{E}_{pk}(x_b) :$
$A_2^{\mathcal{O}_2}(x_0, x_1, s, y) = b\Big] - 1$
where

If atk=cpa then $\mathcal{O}_1(\cdot) = \epsilon$ and $\mathcal{O}_2(\cdot) = \epsilon$
If atk=cca1 then $\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ and $\mathcal{O}_2(\cdot) = \epsilon$
If atk=cca2 then $\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ and $\mathcal{O}_2(\cdot) = \mathcal{D}_{sk}(\cdot)$.

We insist that A_2 does not ask its oracle to decrypt y. We say that Π is secure in the sense of IND-ATK if A being polynomial time implies that $\mathsf{Adv}_{A,\Pi}^{ind-atk}(k)$ is negiligible.

Definition 2.4 We say that A (t, ϵ)-breaks $\Pi(1^k)$ in the sense of ATK if

$$\mathsf{Adv}_{A,\Pi}^{ind-atk}(k) \geq \epsilon$$

and A runs for at most t steps. We also say that $\Pi(1^k)$ is (t, ϵ)-secure in the sense of ATK if there exists no A which (t, ϵ)-breaks $\Pi(1^k)$.

The random oracle version of this security notion is defined by allowing A to make access to a random oracle G (or G and H), which depends on Π. The probability of $\mathsf{Adv}_{A,\Pi}^{ind-atk}(k)$ is taken over the random oracle G (or G and H) as well.

3 Proposed IND-CCA1 encryption scheme

3.1 Proposed scheme

Let $\Pi' = (\mathcal{G}', \mathcal{E}', \mathcal{D}')$ be an asymmetric encryption scheme. Let $G : \{0,1\}^k \to \{0,1\}^n$ be a random bit generator. G is modeled as a random oracle.

Then we present a new asymmetric encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ which can encrypt long messages with small overhead. In the proposed scheme, $\mathcal{G} = \mathcal{G}'$ and:

- Encryption
$$\mathcal{E}_{pk}(m) = m \oplus G(r) || \mathcal{E}'_{pk}(r),$$
where r is a random number.

- Decryption
$$\mathcal{D}_{sk}(c_1||c_2) = \begin{cases} \perp & if\ \mathcal{D}'_{sk}(c_2) = \perp \\ c_1 \oplus \mathcal{G}(D'_{sk}(c_2)) & otherwise. \end{cases}$$

We will prove that Π is secure in the sense of IND-CCA1 if Π' is secure in the sense of IND-CCA1.

Let k denote the length of r and n denote the length of m.

Theorem 3.1 Suppose that there exists an adversary $A = (A_1, A_2)$ that (t, ϵ)-breaks $\Pi(1^k)$ in the sense of CCA1 with at most q_G queries to G. Then there exists an adversary $B = (B_1, B_2)$ that (t', ϵ')-breaks $\Pi'(1^k)$ in the sense of CCA1, where

$$t' \leq t + O(k), \quad \epsilon' \geq \epsilon - \frac{3q_G}{2^k}.$$

3.2 Proof of Theorem 3.1

We first show how to construct B by using A as a blackbox.

(Find stage B_1). On input a public key pk, B_1 gives pk to A_1 and runs A_1. After this, A_1 will make two kinds of oracle queries: "give me the value of G on g"(G-query) or "give me the plaintext m for a ciphertext (α, β)".

To answer Q-query, B_1 makes the query-mapping set (Q, A) as follows. Start with $Q = \phi$ and $T = \phi$. Suppose A_1 says "give me the value of G on g".

(a) If $g \notin Q$, then choose $G_g \in \{0,1\}^n$ randomly, set $G(g) = G_g$ and return G_g. Also, set $Q = Q \cup \{g\}$ and $T = T \cup (g, G_g)$.

(b) If $g \in Q$, then find $(g, G_g) \in T$ and return G_g.

Next suppose that A_1 says "give me the plaintext m for a ciphertext (α, β)". Then B_1 sends β to his decryption oracle $\mathcal{D}'_{sk}$.

(c) If $\mathcal{D}'_{sk}$ answer $\perp$, then B_1 returns $\perp$ to A_1.

(d) If $\mathcal{D}'_{sk}$ returns $g \in Q$, then B_1 finds $(g, G_g) \in T$ and returns $G_g \oplus \alpha$ to A_1.

(e) If $\mathcal{D}'_{sk}$ returns $g \notin Q$, then B_1 chooses $G_g \in \{0,1\}^n$ randomly, sets $G(g) = G_g$ and returns $G_g \oplus \alpha$ to A_1. B_1 also sets $Q = Q \cup \{g\}$ and $T = T \cup (g, G_g)$.

Finally, A_1 outputs $(m_0, m_1, info)$. Then B_1 chooses $r_0, r_1 \in \{0,1\}^k$ such that $r_0 \neq r_1$ randomly and outputs $(r_0, r_1, info')$, where

$$info' = (m_0, m_1, info).$$

(Guess stage B_2). B_2 is given $(r_0, r_1, info')$ and $\mathcal{E}'_{pk}(r_b)$, where b is a random bit.

(f) If $r_0 \in Q$ and $r_1 \notin Q$, then B_2 outputs $b' = 0$ and stops. If $r_0 \notin Q$ and $r_1 \in Q$, then B_2 outputs $b' = 1$ and stops. If $r_0 \in Q$ and $r_1 \in Q$, then B_2 outputs a random bit b' and stops.

If $r_0 \notin Q$ and $r_1 \notin Q$, then B_2 chooses $\widetilde{\alpha} \in \{0,1\}^n$ randomly, sets $\mathcal{E}_{pk}(m_b) = (\widetilde{\alpha}, \mathcal{E}_{pk}(r_b))$. B_2 next gives $(m_0, m_1, info)$ and $\mathcal{E}_{pk}(m_b)$ to A_2 and runs A_2. After this, A_2 will G-query $\tilde{r}$.

(g) If $\tilde{r} = r_0$ or r_1, then B_2 outputs b' such that $\tilde{r} = r_{b'}$ and stops.

(h) Otherwise, B_2 behaves as shown in (a) and (b). Finally A_2 outputs a bit b'. B_2 then outputs b' and stops.

Next we will estimate $\Pr(b' = b)$. It is clear that B generates the view of A correctly until B stops. Let

$$\begin{aligned} ASK_b &\stackrel{\triangle}{=} A \text{ makes } G\text{-query } r_b \text{ in the real world.} \\ ASK_{1-b} &\stackrel{\triangle}{=} A \text{ makes } G\text{-query } r_{1-b} \text{ in the real world.} \end{aligned}$$

Lemma 3.1

1. $\Pr[ASK_{1-b}] \leq q_G/2^k$.

2. $\Pr[ASK_b] \geq \epsilon$.

Lemma 3.2 For any events X and Y,

$$\Pr(X \wedge Y) \geq \Pr(X) - \Pr(\neg Y).$$

The proofs are given in Appendix.

Now define

$$ASK \quad \stackrel{\triangle}{=} \quad \text{(f) or (g) happens..}$$

Then

$$\Pr(b = b') = \Pr(b' = b \wedge ASK) + \Pr(b' = b \wedge \neg ASK) \tag{1}$$

On the first term, note that if ASK_b happens and ASK_{1-b} never happens, then ASK happens with $b' = b$. Therefore, we have

$$\begin{aligned} \Pr(b' = b \wedge ASK) &\geq \Pr(ASK_b \wedge \neg ASK_{1-b}) \\ &\geq \Pr(ASK_b) - \Pr(ASK_{1-b}) \end{aligned}$$

On the second term, note that A has no information on r_b if ASK never happens. Therefore,

$$\begin{aligned} \Pr(b' = b \wedge \neg ASK) &= \frac{1}{2}\Pr(\neg ASK) = \frac{1}{2}(1 - \Pr(ASK)) \\ &\geq \frac{1}{2}(1 - \Pr(ASK_b \vee ASK_{1-b})) \\ &\geq \frac{1}{2}(1 - \Pr(ASK_b) - \Pr(ASK_{1-b})) \end{aligned}$$

Hence,

$$\begin{aligned} \Pr(b' = b) &\geq \Pr(ASK_b) - \Pr(ASK_{1-b}) \\ &\quad + \frac{1}{2}(1 - \Pr(ASK_b) - \Pr(ASK_{1-b})) \\ &= \frac{1}{2} + \frac{1}{2}\Pr(ASK_b) - \frac{3}{2}\Pr(ASK_{1-b}) \\ 2\Pr(b' = b) - 1 &\geq \Pr(ASK_b) - 3\Pr(ASK_{1-b}) \\ &\geq \epsilon - \frac{3q_G}{2^k} \end{aligned}$$

Consequently, $\epsilon' \geq \epsilon - \frac{3q_G}{2^k}$. Finally, it is clear that $t' \leq t + O(k)$.

4 Proposed IND-CCA2 encryption scheme

4.1 Proposed scheme

Let $\Pi' = (\mathcal{G}', \mathcal{E}', \mathcal{D}')$ be an asymmetric encryption scheme. Let $G : \{0,1\}^k \to \{0,1\}^n$ be a random bit generator and $H : \{0,1\}^n \to \{0,1\}^k$ be a hash function.

Then we present a new asymmetric encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ which can encrypt long messages with small overhead.

In the proppposed scheme, $\mathcal{G} = \mathcal{G}'$ and:

- Encryption: $\mathcal{E}_{pk}(m) = (c_1, c_2)$,
  where
  $$c_1 = m \oplus G(r), \quad c_2 = \mathcal{E}'_{pk}(r \oplus H(c_1))$$
  and r is a random number.

- Decryption
  $$\mathcal{D}_{sk}(c_1||c_2) = \begin{cases} \perp & if\ \mathcal{D}'_{sk}(c_2) = \perp \\ c_1 \oplus G(\hat{r}) & otherwise, \end{cases}$$
  where $\hat{r} = \mathcal{D}_{sk}(c_2) \oplus H(c_1)$.

We will prove that Π is secure in the sense of IND-CCA2 if Π' is secure in the sense of IND-CCA2.

Let k denote the length of r and n denote the length of m.

Theorem 4.1 Suppose that there exists an adversary $A = (A_1, A_2)$ that (t, ϵ)-breaks $\Pi(1^k)$ in the sense of CCA2 with at most q_G queries to G and with at most q_D queries to $\mathcal{D}_{sk}$. Then there exists an adversary $B = (B_1, B_2)$ that (t', ϵ')-breaks $\Pi'(1^k)$ in the sense of CCA2, where

$$t' \leq t + O(k), \quad \epsilon' \geq \epsilon - \frac{5q_G}{2^k} - \frac{q_G q_D}{2^{k-2}}.$$

4.2 Proof of Theorem 4.1

We first show how to construct B by using A as a blackbox. In what follows, B simulates H randomly.

(Find stage B_1).

B_1 behaves similarly to Sec.3.2. Finally, A_1 outputs $(m_0, m_1, info)$. Then B_1 chooses $u_0, u_1 \in \{0,1\}^k$ such that $u_0 \neq u_1$ randomly and outputs $(u_0, u_1, info')$, where

$$info' = (m_0, m_1, info).$$

(Guess stage B_2).

B_2 is given $(u_0, u_1, info')$ and $\mathcal{E}'_{pk}(u_b)$, where b is a random bit. Then B_2 chooses $\widetilde{\alpha} \in \{0,1\}^n$ randomly and sets $\mathcal{E}_{pk}(m_b) = (\widetilde{\alpha}, \mathcal{E}_{pk}(u_b))$. Let

$$r_0 = u_0 \oplus H(\widetilde{\alpha}), \quad r_1 = u_1 \oplus H(\widetilde{\alpha}).$$

If $r_0 \in Q$ and $r_1 \notin Q$, then B_2 outputs $b' = 0$ and stops. If $r_0 \notin Q$ and $r_1 \in Q$, then B_2 outputs $b' = 1$ and stops. If $r_0 \in Q$ and $r_1 \in Q$, then B_2 outputs a random bit b' and stops.

If $r_0 \notin Q$ and $r_1 \notin Q$, then B_2 gives $(m_0, m_1, info)$ and $\mathcal{E}_{pk}(m_b)$ to A_2 and runs A_2. Suppose that A_2 makes G-query $\tilde{r}$. If $\tilde{r} = r_0$ or r_1, then B_2 outputs b' such that $\tilde{r} = r_{b'}$ and stops. Otherwise, B_2 behaves as shown in (a) and (b) of Sec.3.2.

Next suppose that A_2 says "give me the plaintext m for a ciphertext (α, β)". If $\beta \neq \mathcal{E}'_{pk}(u_b)$, then B_2 behaves similarly to (c),(d) and (e) of Sec.3.2 and can give $\mathcal{E}_{pk}(\alpha||\beta)$ to A_2.

(1) If $\alpha \neq \tilde{\alpha}$ and $\beta = \mathcal{E}'_{pk}(u_b)$, then B_2 cannot send β to the decryption oracle $\mathcal{D}'_{sk}$. In this case, B_2 behaves as follows. Let $X = empty$ and

$$\gamma_0 = u_0 \oplus H(\alpha), \quad \gamma_1 = u_1 \oplus H(\alpha). \tag{2}$$

(2) If $\gamma_0 \in Q$ or $\gamma_1 \in Q$, then B_2 outputs a random bit b' and stops.

(3) Otherwise, B_2 chooses $x \in \{0,1\}^n$ randomly, sets $G(\gamma_0) = G(\gamma_1) = x$ and return $m = \alpha \oplus x$. B_2 also sets $Q = Q \cup \{\gamma_0\} \cup \{\gamma_1\}$, $T = T \cup (\gamma_0, x) \cup (\gamma_1, x)$ and $X = X \cup (\gamma_0, \gamma_1)$.

Finally, A_2 outputs b'. Then B_2 outputs b'.

Now the advantage ϵ' of B is computed similarly to Sec.3.2 except for (1). First, if (2) happens, then B_2 cannot output b correctly. This probability is estimated as follows because $H(\alpha)$ is random in eq.(2).

$$\Pr(\gamma_0 \in Q \text{ or } \gamma_1 \in Q) \leq \Pr(\gamma_0 \in Q) + \Pr(\gamma_1 \in Q) \leq 2q_G/2^k = q_G/2^{k-1}.$$

$$\Pr((2) \text{ happens }) \leq q_G q_D/2^{k-1}.$$

Second, on (3), B fails to generates the view of A correctly if A makes G-queries both γ_0 and γ_1 for some $(\gamma_0, \gamma_1) \in X$. This probability is at most $q_G/2^k$ because A has no information on u_{1-b}. Hence, we have

$$\epsilon' \quad \geq \quad \epsilon - \frac{3q_G}{2^k} - \frac{q_G q_D}{2^{k-2}} - \frac{2q_G}{2^k} = \epsilon - \frac{5q_G}{2^k} - \frac{q_G q_D}{2^{k-2}}.$$

Finally, it is easy to see that $t' \leq t + O(k)$.

5 Plaintext awareness encryption scheme

5.1 Plaintext awareness

An adversary B for plaintext awareness is given a public key pk and access to the random oracle G and H. We also provide B with an oracle for $\mathcal{E}_{pk}^{G,H}$. The adversary outputs a ciphertext y. To be plaintext aware the adversary B should necessarily "know" the decryption x of its output y.

Let $\mathcal{T}_G$ denote the set of all pairs of B's queries and the corresponding answers from G, $\mathcal{T}_H$ denote the set of all pairs of B's queries and the corresponding answers from H, $\mathcal{Y}$ denote the set of all answers received as ciphertexts from $\mathcal{E}_{pk}^{G,H}(\cdot)$. y (output of B) is not in $\mathcal{Y}$. We write the experiment above as $(\mathcal{T}_G, \mathcal{T}_H, \mathcal{Y}, y) \leftarrow B^{G,H,\mathcal{E}_{pk}}(pk)$.

Definition 5.1 Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme, let B be an adversary, and let K be an algorithm (the "knoeledge extractor"). For any $k \in \mathsf{N}$ let $\mathsf{Succ}_{K,B,\Pi}^{pa}(k) \stackrel{\triangle}{=}$

$$\Pr\Big[G, H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}(1^k); (\mathcal{T}_G, \mathcal{T}_H, \mathcal{Y}, y) \leftarrow B^{G,H,\mathcal{E}_{pk}}(pk) : \\ K(\mathcal{T}_G, \mathcal{T}_H, \mathcal{Y}, y, pk) = \mathcal{D}_{sk}(y)\Big],$$

where Ω is the map family from an appropriate domain to an appropriate range. We insist that $y \notin \mathcal{Y}$; that is, B never outputs a string y which coincides with the value returned from some $\mathcal{E}_{pk}^{G,H}$-query.

We say that K is a $\lambda(k)$-extractor if K has running time polynomial in the length of its imputs and for every adversary B, $\mathsf{Succ}_{K,B,\Pi}^{pa}(k) \geq \lambda(k)$.

We say that Π is secure in the sense of PA if Π is secure in the sense of IND-CPA and there exists a $\lambda(k)$-extractor K where $1 - \lambda(k)$ is negligible.

5.2 Proposed scheme

Let $\Pi' = (\mathcal{G}', \mathcal{E}', \mathcal{D}')$ be an asymmetric encryption scheme. Let $G : \{0,1\}^k \to \{0,1\}^n$ be a random bit generator and $H : \{0,1\}^n \to \{0,1\}^k$ be a hash function. Then we present a new asymmetric encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$. In the proposed scheme, $\mathcal{G} = \mathcal{G}'$ and:

- Encryption: $\mathcal{E}_{pk}(m) = (c_1, c_2)$,
  where
  $$c_1 = m0^k \oplus G(r), \quad c_2 = \mathcal{E}'_{pk}(r \oplus H(c_1))$$
  and r is a random number.

- Decryption.
  $$\mathcal{D}_{sk}(c_1||c_2) = \begin{cases} \hat{m} & if\ \mathcal{D}'_{sk}(c_2) \neq \perp \text{ and} \\ & c_1 \oplus G(\hat{r}) = \hat{m}0^k \text{ for some } \hat{m}, \\ \perp & otherwise, \end{cases}$$
  where $\hat{r} = \mathcal{D}_{sk}(c_2) \oplus H(c_1)$.

Theorem 5.1 *If Π' is secure in the sense of PA, then Π is secure in the sense of PA.*

5.3 Proof of Theorem 5.1

We can show that Π is secure in the sense of IND-CPA if Π' is secure in the sense of IND-CPA. (The details will be given in the final paper.)

Next suppose that there exists an extractor K' for Π'. Then we will show that there exists an extractor K for Π. Let A be an adversary for Π. Suppose that
$$(\mathcal{T}_G, \mathcal{T}_H, \mathcal{T}_X, \mathcal{Y}, y) \leftarrow A^{G,H,X,\mathcal{E}_{pk}}(pk),$$
where X is the random oracle for Π' and $\mathcal{T}_X$ is the list of (query, answer) pairs on X made by A. Let $y = (c_1, c_2)$ and
$$\mathcal{Y} = ((y_{11}, y_{12}), (y_{21}, y_{22}), \cdots,).$$

We can consider an adversary B for Π' such that
$$(\mathcal{T}_X, \mathcal{Y}', c_2) \leftarrow B^{X,\mathcal{E}'_{pk}}(pk),$$

where $\mathcal{Y}' = (y_{12}, y_{22}, \cdots,)$. Indeed, B can compute $y = (c_1, c_2)$ in the same way as A does by simulating G and H by himself. For $\mathcal{Y}'$, suppose that A makes a query m_i to $\mathcal{E}_{pk}$ and obtains $(y_{i1}, yi2)$. Since $y_{i2} = \mathcal{E}'_{pk}(r_i \oplus H(y_{i2}))$ for a random number r_i, y_{i2} is a ciphertext of a random plaintext $u_i \triangleq r_i \oplus H(y_{i2})$ regardless of m_i. Therefore, B sends a random number u_i to the encryption oracle $\mathcal{E}'_{pk}$ as a plaintext and can obtain a ciphertext y_{i2}. Then from our assumption, it holds that $K'(\mathcal{T}_X, \mathcal{Y}', c_2, pk) = \mathcal{D}'_{sk}(c_2)$ with overwhelming probability.

Now K behaves as follows. $K(\mathcal{T}_G, \mathcal{T}_H, \mathcal{T}_X, \mathcal{Y}, y, pk)$ first runs $K'(\mathcal{T}_X, \mathcal{Y}', c_2, pk)$ and obtains $\mathcal{D}'_{sk}(c_2)$ with overwhelming probability. If $\mathcal{D}'_{sk}(c_2) = \perp$, then K outputs $\perp$. Otherwise, K obtains $(c_1, \mathcal{D}'_{sk}(c_2))$. Then since $(c_1, \mathcal{D}'_{sk}(c_2))$ are written as

$$c_1 = \beta \oplus G(r) \text{ and } \mathcal{D}'_{sk}(c_2) = r \oplus H(c_1)$$

for some r and β, we can show that K can output $\mathcal{D}_{sk}(c_1 || c_2)$ with overwhelming probability similarly to [3]. (The details will be given in the final paper.)

References

[1] M.Abdalla,M.Bellare and P.Rogaway. "DHAES: An encryption scheme based on the Diffie-Hellman problem ", submission to IEEE P1363.

[2] M.Bellare, A.Desai, D.Poincheval and P.Rogaway. "Relations among notions of security for public key encryption schemes ", In *Proc. of Crypto'98, Lecture Notes in Computer Science, LNCS 1462, Springer Verlag*, pages 26–45, 1998.

[3] M.Bellare and P.Rogaway. "Optimal asymmetric encryption - How to encrypt with RSA ", In *Proc. of Eurocrypt'94, Lecture Notes in Computer Science, LNCS 950, Springer Verlag*, pages 92–111, 1994.

[4] R. Cramer and V.Shoup. "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack ", In *Proc. of Crypto'98, Lecture Notes in Computer Science, LNCS 1462, Springer Verlag*, pages 13–25, 1998.

[5] E.Fujisaki and T.Okamoto. "Secure integration of asymmetric and symmetric encryption schemes ", In *Proc. of Crypto'99, Lecture Notes in Computer Science, LNCS 1666, Springer Verlag*, pages 537–554, 1999.

A Proof of lemma 3.1

1. It is clear that A has no information on r_{1-b}. Further, r_{1-b} is chosen by B_1 randomly. Therefore, the probability that B_1 chooses r_{1-b} which is queried by A is at most $q_G/2^k$. This probability is equal to $\Pr[ASK_{1-b}]$.

2. Since A (t,ϵ)-break $\mathit{\Pi}(1^n)$, we have

$$\Pr[A \text{ quesses } b \text{ correctly}] \geq \frac{1}{2} + \frac{1}{2}\epsilon$$

On the other hand, A has no information on b if A never makes G-query r_b. Therefore,

$$\begin{aligned}
& \Pr[A \text{ quesses } b \text{ correctly}] \\
= & \Pr[A \text{ quesses } b \text{ correctly}|ASK_b] \Pr[ASK_b] \\
& +\Pr[A \text{ quesses } b \text{ correctly}|\neg ASK_b] \Pr[\neg ASK_b] \\
\leq & \Pr[ASK_b] + \Pr[A \text{ quesses } b \text{ correctly}|\neg ASK_b]\Pr[\neg ASK_b] \\
= & \Pr[ASK_b] + \frac{1}{2}\Big(1 - \ Pr[ASK_b]\Big) \\
= & \frac{1}{2}\Pr[ASK_b] + \frac{1}{2}.
\end{aligned}$$

Hence,

$$\frac{1}{2} + \frac{1}{2}\epsilon \leq \frac{1}{2}\Pr[ASK_b] + \frac{1}{2}$$
$$\Pr[ASK_b] \geq \epsilon.$$

Q.E.D.

B Proof of lemma 3.2

$$\begin{aligned}
\Pr(X \wedge Y) &= \Pr(X) \cdot \Pr(Y|X) \\
&= \Pr(X)(1 - \Pr(\neg Y|X)) \\
&= \Pr(X) - \Pr(\neg Y \wedge X) \\
&\geq \Pr(X) - \Pr(\neg Y)
\end{aligned}$$

Q.E.D.