Chosen Ciphertext Security with Optimal Ciphertext Overhead

Masayuki Abe[1], Eike Kiltz[⋆2] and Tatsuaki Okamoto[1]

[1] NTT Information Sharing Platform Laboratories, NTT Corporation, Japan
[2] CWI Amsterdam, The Netherlands

Abstract. Every public-key encryption scheme has to incorporate a certain amount of randomness into its ciphertexts to provide semantic security against chosen ciphertext attacks (IND-CCA). The difference between the length of a ciphertext and the embedded message is called the *ciphertext overhead*. While a generic brute-force adversary running in 2^t steps gives a theoretical lower bound of t bits on the ciphertext overhead for IND-CPA security, the best known IND-CCA secure schemes demand roughly $2t$ bits even in the random oracle model. Is the t-bit gap essential for achieving IND-CCA security?

We close the gap by proposing an IND-CCA secure scheme whose ciphertext overhead matches the generic lower bound up to a small constant. Our scheme uses a variation of a four-round Feistel network in the random oracle model and hence belongs to the family of OAEP-based schemes. Maybe of independent interest is a new efficient method to encrypt long messages exceeding the length of the permutation while retaining the minimal overhead.

1 Introduction

1.1 Background

Motivation. Ever since Goldwasser and Micali introduced the concept of "probabilistic encryption" [16] it is well understood that every public-key encryption scheme has to incorporate a certain amount of randomness into their ciphertexts in order to achieve semantic security. Thus a ciphertext c must be longer than the embedded message m and the difference $\ell_{\mathsf{oh}} := |c| - |m|$ is called the *ciphertext overhead*. In order to achieve stronger security properties, the ciphertext overhead tends to be even larger due to the use of extended randomness or extra integrity

[⋆] Supported by the research program Sentinels (`http://www.sentinels.nl`). Sentinels is being financed by Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.

checking mechanisms. In this paper we are asking for the minimal possible ciphertext overhead to protect against adaptive chosen ciphertext attacks (IND-CCA security).

A Generic Lower Bound. A ciphertext overhead of ℓ_{oh} bits means that at most ℓ_{oh} bits of randomness can be incorporated into a ciphertext. A brute-force adversary in the IND-CPA experiment can exhaustively search for the randomness used for the challenge ciphertext. After encrypting one of the challenge messages up to 2^t times, it has an advantage of $\Omega(2^t/2^{\ell_{\mathsf{oh}}})$. Requiring the advantage to be smaller than $2^{-\varepsilon}$ (and ignoring small additive constants), it must hold that

$$\ell_{\mathsf{oh}} \;\geq\; t + \varepsilon \;.$$

Accordingly, $t + \varepsilon$ bits are a lower bound on the ciphertext overhead with respect to adversaries running in 2^t steps and having a success probability of at most $2^{-\varepsilon}$, by counting encryption as one step. (We refer to Section 2 for a more formal treatment.) We say that the ciphertext overhead is *optimal* if it matches the lower bound up to a (small) constant term, i.e., if $\ell_{\mathsf{oh}} \leq t + \varepsilon + O(1)$. Since every IND-CPA adversary is also an IND-CCA adversary, the above lower bound also applies to IND-CCA secure schemes.

For a number of schemes the ciphertext overhead primarily depends on the size of the underlying number-theoretic primitive, which often suffers from more sophisticated attacks. For example, ciphertexts of ElGamal-type schemes contain at least one group element of overhead which must be longer than $2t + \varepsilon$ bits due to the generic square-root bounds on the discrete-logarithm problem. Hence, the ciphertext overhead of such schemes can never match the generic lower bound.

Upper Bounds from Existing Schemes. Among the cryptosystems based on trapdoor permutations, there are ones whose ciphertext overhead is essentially independent of the size of the underlying permutation. We focus on such schemes for the rest of the paper. An example with optimal ciphertext overhead is the basic version of OAEP [4], which omits the zero padding and therefore only offers IND-CPA security. Considering IND-CCA security, however, OAEP loses its optimal ciphertext overhead as exemplified in Section 2.2. On the other hand, concrete security proofs for existing schemes provide upper bounds on the ciphertext overhead with which the desired level of security is attained. Table 1 summarizes the ciphertext overhead of existing schemes. Its content is discussed in the rest of this section.

Scheme	Ciphertext Overhead	Assumption on TDP	#Feistel rounds
OAEP [4, 15]	$\ell_{\mathsf{oh}} \leq 3t + 2\varepsilon$	SPD-OW	2
OAEP+ [25]	$\ell_{\mathsf{oh}} \leq 3t + 2\varepsilon$	OW	2
PSS-E [10]	$\ell_{\mathsf{oh}} \leq 2t + 2\varepsilon$	SPD-OW	2
PSP2 S-Pad [14]	$\ell_{\mathsf{oh}} \leq 2t + 2\varepsilon$	OW	4
OAEP-3R [23]	$\ell_{\mathsf{oh}} \leq 2t + \varepsilon$	OW	3
OAEP-4X (ours)	$\ell_{\mathsf{oh}} = t + \varepsilon$	OW	4

Table 1. Upper bounds on the ciphertext overhead (up to small additive constants) in OAEP variants for $(2^{\varepsilon}, 2^{-t})$-adversaries. The lower bound is $\ell_{\mathsf{oh}} \geq t + \varepsilon$. OW: one-wayness. SPD-OW: set partial domain one-wayness.

IND-CCA Security via Validity Checking. As in OAEP, a common approach [25, 19, 21, 10, 20, 14] to achieve IND-CCA security is to attach a deterministic *validity string* (such as zero-padding or a hash of the message, etc) to the message (or the ciphertext) so that decryption can verify and reject almost all invalid ciphertexts. The ciphertext overhead is thus determined by the size of the randomness and the validity string. OAEP and the schemes in [25, 19] require randomness of $2t+\varepsilon$ bits plus a validity string of $t+\varepsilon$ bits. (See Section 2.2 for details on how to compute these values.) Their ciphertext overhead is thus $\ell_{\mathsf{oh}} = 3t+2\varepsilon$. The schemes in [10, 14] have a better security reduction and achieve $\ell_{\mathsf{oh}} = 2t + 2\varepsilon$, which seems the best one can expect as long as encryption incorporates a validity string into the ciphertexts.

Validity-free Encryption. A considerable step towards minimizing the ciphertext overhead was the *validity-free* approach introduced by Phan and Pointcheval [22, 23]. In their scheme (called 3-round OAEP) decryption never rejects but returns a randomly looking message if a given ciphertext was not properly created with the encryption algorithm. Since no validity string is needed, the ciphertext overhead only depends on the randomness. As we shall discuss later, their security reduction however forces the ciphertext overhead to be $\ell_{\mathsf{oh}} = k_r = 2t + \varepsilon$ bits because of a "quadratic term" $q_h q_d / 2^{k_r}$ that appears in the success probability of their reduction. A more recent scheme in [13] suffers from the same problem. In summary, these schemes successfully eliminate the validity string but instead demand an extended randomness to prove IND-CCA security.

Encrypting long messages. The problem of getting optimal overhead becomes even more difficult when considering longer messages. Notice that all above schemes limit the messages to the size of the permutation minus the overhead. To encrypt long inputs, [4, 17] suggest to stretch the width of the Feistel network to cover the entire message and apply the permutation only to a part of the output. But no general and formal treatment has been given to this methodology and it is unclear if and how it affects the ciphertext overhead. Furthermore, for schemes that use several Feistel rounds, this approach is expensive in computation as every internal hash function has to deal with a long input or output. A number of methods for constructing hybrid encryption are available (e.g., [12, 8, 9, 1, 6]), but they all increase the ciphertext overhead mainly because a one-time session-key is being encrypted.

1.2 Our Contribution

Our main contribution is an IND-CCA-secure public-key encryption scheme with optimal ciphertext overhead based on arbitrary family of trapdoor one-way permutation in the random oracle model. We follow the validity-free approach of 3-round OAEP [22] but instead use a 4-round Feistel network. (See Figure 1 in Section 3 for a diagram.) We stress that the essential difference is not the increased number of rounds; it is rather the way we bind the message to the randomness in the first round of the Feistel network while most of OAEP variants separately input the message and the randomness. (See Section 1.3 for more intuition.)

Our contribution is mostly theoretical; Our scheme demonstrates that lower and upper bounds on the ciphertext overhead with respect to IND-CCA security can match up to a small additive constant in the random oracle model. The design approach that binds the message to the randomness and the security proof may be of technical interest, too. In practice, when implemented with an 1024-bit RSA permutation (80-bit security), our scheme encrypts 943-bit and longer messages while it is 863 bits for a known best scheme, which is at most 9% increase of the message space. Though such a t-bit saving may have limited practical impact in general, the scheme could find applications with edgy requirements in bandwidth.

We also introduce a novel method to securely combine simple passively secure symmetric encryption with the Feistel network to encrypt long messages while retaining the optimal ciphertext overhead. While the construction is interesting in that it suggests a new variant of a KEM that allows partial message recovery, it is interesting also in a theoretical sense as it illustrates the difference in the properties of the round functions in a 4-round Feistel network as it will be discussed later.

1.3 Technical Overview

Achieving Optimal Overhead. We explain the technical details in 3-round OAEP that seem to make it difficult to prove an optimal ciphertext overhead. The extended randomness of size $k_r \geq 2t + \varepsilon$ stems from a quadratic term $q_h\, q_d / 2^{k_r}$ in the success probability of the security reduction. Since an adversary running in time 2^t can make at most $q_h \leq 2^t$ hash oracle queries and $q_d \leq 2^t$ decryption queries, we must assume that $q_h\, q_d \approx (2^t)^2$. Requiring $q_h\, q_d / 2^{k_r} \leq 2^{-\varepsilon}$ results in $k_r \geq 2t + \varepsilon$.

Where does this quadratic loss in the reduction actually come from? In the security proof, every time the simulated decryption oracle receives a ciphertext that was not legitimately generated by asking the random oracles, it returns a random plaintext. Later, it patches the hash table for the simulated randomness so that the hash output looks consistent. The patching fails if the randomness has already been asked to the random oracle. This happens with probability at most $q_h / 2^{k_r}$ since there are at most q_h hash queries. Throughout the attack, there are at most q_d decryption queries and hence the error probability of the patching is bounded by $q_h\, q_d / 2^{k_r}$.

Our main technical contribution is to provide a security analysis for our scheme where only linear terms of the form $q_h / 2^{k_r}$ or $q_d / 2^{k_r}$ appear. We overcome the problem observed in 3-round OAEP by feeding the randomness *together* with a part of the input message (say m_1) into the hash function, i.e., by computing $H_1(r \,\|\, m_1)$. This link between the randomness and the message allows the reduction to partition hash queries by m_1 and therefore reducing the error probability in patching the hash table to $q_{h,m_1} / 2^{k_r}$, where q_{h,m_1} is the number of hash queries with respect to m_1. By summing up the probabilities for all m_1 returned from the decryption oracle, the error probability is bounded by $\sum_{m_1} q_{h,m_1} / 2^{k_r} \leq q_h / 2^{k_r}$. The quadratic term is thus eliminated. The fourth round of the Feistel network is then needed to cover m_1.

Encrypting Long Messages. In order to encrypt long messages exceeding the size of the permutation (while retaining the optimal overhead), we incorporate the idea of the Tag-KEM/DEM framework [1] that allows to use a simple passively secure length-preserving symmetric cipher. The exceeding part of the message is encrypted with the symmetric cipher whose key is derived from the randomness used in the asymmetric part of encryption. The symmetric part is then tied to the asymmetric part of the ciphertext by feeding it back into one of the hash function used in the Feistel network. Conceptually, our approach is similar to Tag-

KEMs with partial ciphertext recovery [6] but in our case the message can be directly recovered. Namely, the main part of our construction can be used as *a Tag-KEM with partial message recovery.*

A concrete technical difficulty is how and where to include the feedback from the symmetric part. Including it in the F-function (random oracle) in every round of the 4-round Feistel network should work but may be redundant. Is it then secure if the feedback is given only to one of the F-functions? Which one? [24] showed that the inner two rounds have different properties than the outer two ones. Does that also apply to our case? Our result shows that it is sufficient to give the feedback to one of the inner two hash functions. We remark that when including the feedback only in the outer hash functions then either our security proof does no longer hold or there is a concrete attack. We refer to Section 3.3 for further details.

1.4 Related Work

In Other Models. [22] constructed a simple scheme with optimal ciphertext overhead in the ideal full-domain permutation model. Looking at the construction and the security proof, however, one can see that the model is very strong and has little difference from idealizing the encryption function itself. Recently it is shown that ideal full-domain permutation can be constructed using random oracles [11] but the reduction is very costly and a *tight reduction* needed to retain the optimal overhead is highly unlikely. Note that [22] could only present a non-optimal scheme in the random oracle model, which shows the difficulty of achieving the optimality.

For Short Messages. Schemes based on general one-way permutations can never offer the optimal overhead for messages shorter than the size of the permutation. For the state of art in this issue, we refer to [2] which presents a scheme that offers non-optimal but $\ell_{\mathsf{oh}} \geq 2t+\varepsilon$ that is currently the shortest overhead for messages of arbitrary (small) length. It is left as another open problem to construct a scheme with optimal overhead for arbitrary message size.

2 Lower Bound of Ciphertext Overhead

We follow the standard definition of public-key encryption $\mathsf{PKE} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ and indistinguishability against chosen plaintext attacks (IND-CPA) and adaptive chosen ciphertext attacks (IND-CCA). For formal definitions, we refer to the full version [3].

2.1 General Argument

Let $\mathsf{PKE} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme and let $\mathcal{M}$ and $\mathcal{R}$ be the message and randomness space associated to a public-key pk. For $(pk, sk) \leftarrow \mathcal{G}(1^k)$ and $M \in \mathcal{M}$, let $C(M)$ denote the set of ciphertexts that recover message M. The ciphertext overhead ℓ^k_{oh} with respect to k is defined by $\ell^k_{\mathsf{oh}} = |\mathcal{E}_{pk}(M; r)| - |M|$. To obtain a simple form of the lower bound, we restrict ourselves to PKE where ℓ^k_{oh} is a fixed positive constant for any $pk \in \mathcal{G}(1^k)$, $M \in \mathcal{M}$ and $r \in \mathcal{R}$.

Let A be an adversary that runs in 2^t steps and breaks the semantic (IND-CPA) security of PKE with advantage at most $2^{-\varepsilon}$. To study the relation between the adversary's ability and the ciphertext overhead, we treat t, ε independently from k and represent the bounds of the ciphertext overhead as a function $\ell^k_{\mathsf{oh}}(t, \varepsilon)$. In the following argument, we count every encryption as one step. A launches the following attack.

1. Given pk generated by $(pk, sk) \leftarrow \mathcal{G}(1^k)$, pick arbitrary M_0 and M_1 of the same length from $\mathcal{M}$. Send (M_0, M_1) to the challenger and receive $c^* = \mathcal{E}_{pk}(M_b)$ where $b \leftarrow \{0, 1\}$.
2. Repeat the following up to 2^t times.
   - $r \leftarrow \mathcal{R}$, $c = \mathcal{E}_{pk}(M_0; r)$.
   - If $c = c^*$, output $\tilde{b} = 0$ and stop.
3. Output $\tilde{b} = 1$.

For a string c, let $p(c)$ denote the probability that $c = \mathcal{E}_{pk}(M_0; r)$ happens for uniformly chosen r. Similarly, let $p'(pk)$ denote the probability that pk is selected by $\mathcal{G}(1^k)$. The advantage of adversary A in breaking the semantic security with respect to pk is

$$\begin{aligned}\mathbf{Adv}_{\mathsf{A},pk} &= |\Pr[\tilde{b} = 0 \,|\, b = 0] - \Pr[\tilde{b} = 0 \,|\, b = 1]| \\ &= \Pr[\tilde{b} = 0 \,|\, b = 0] - 0 \\ &= \sum_{c \in C(M_0)} p(c)(1 - (1 - p(c))^{2^t}). \end{aligned} \tag{1}$$

Let η be the min-entropy with respect to the ciphertexts in $C(M_0)$ in bits. Since $p(c) \geq \frac{1}{2^\eta}$ for any $c \in C(M_0)$,

$$\mathbf{Adv}_{\mathsf{A},pk} \geq \sum_{c \in C(M_0)} p(c)(1 - (1 - \frac{1}{2^\eta})^{2^t}) \geq \frac{2^t}{2^\eta} - \frac{2^t - 1}{2^{2\eta}}. \tag{2}$$

Since $\eta \leq \ell^k_{\mathsf{oh}}$, we have

$$\begin{aligned}
\mathbf{Adv}_{\mathsf{A}}(k) &= \sum_{pk \in \mathcal{G}(k)} p'(pk) \cdot \mathbf{Adv}_{\mathsf{A},pk} \\
&\geq \sum_{pk \in \mathcal{G}(k)} p'(pk) \cdot \left(\frac{2^t}{2^{\ell^k_{\mathsf{oh}}}} - \frac{2^t - 1}{2^{2\ell^k_{\mathsf{oh}}}} \right) \\
&\geq \frac{1}{2} \cdot \frac{2^t}{2^{\ell^k_{\mathsf{oh}}}} . \qquad (3)
\end{aligned}$$

Since we require $\mathbf{Adv}_{\mathsf{A}}(k) \leq 2^{-\varepsilon}$, it holds that $2^{-\varepsilon} \geq \frac{1}{2} \cdot \frac{2^t}{2^{\ell^k_{\mathsf{oh}}}}$ for $t, \varepsilon \geq 1$. Thus we have the lower bound:

$$\ell^k_{\mathsf{oh}}(t, \varepsilon) \geq t + \varepsilon - 1 \ . \qquad (4)$$

If $c \leftarrow \mathcal{E}_{pk}(M; r)$ is bijective with respect to c and r, the adversary can search r one by one without duplication and the advantage for this case is $\mathbf{Adv}_{\mathsf{A},pk} = \frac{2^t}{2^\eta}$, which results in $\ell^k_{\mathsf{oh}}(t, \varepsilon) \geq t + \varepsilon$.

In the above discussion we used the simplified argument to count one encryption as one single time unit. More generally, one should count each fundamental cryptographic operation (such as hashing, group operation, etc.) as one step. Hence the value 2^t is understood as the total number of times the adversary performs the fundamental cryptographic operations. A precise assessment is possible by incorporating an adequate scaling factor that represent the exact number of steps (depending on the computational model).

2.2 Example : Ciphertext Overhead of OAEP

OAEP includes randomness of size k_r and zero-padding of size k_v. These parameters define the ciphertext overhead as $\ell_{\mathsf{oh}} = k_r + k_v$. Together with the size of permutation, n, they are provided as a security parameter $k = (n, k_r, k_v)$. According to [15, Th. 1], the advantage of an adversary A against the IND-CCA security of OAEP, making up to q decryption and hash queries is upper bounded by

$$\mathbf{Adv}^{\text{cca}}_{\mathsf{A}}(k) \leq \epsilon_{\text{spd}}(n) + \frac{c\,q^2}{2^{k_r}} + \frac{c' q}{2^{k_v}} \ , \qquad (5)$$

where $\epsilon_{\text{spd}}(n)$ is the probability of breaking set partial one-wayness of the underlying trapdoor permutation of size n, and $c, c' \geq 1$ are two (small) constants.

Consider an $(2^t, 2^{-\varepsilon})$ adversary that can make at most $q \leq 2^t$ oracle queries. Since parameter n can be chosen essentially independently from k_r and k_v, we can safely assume that $\epsilon_{\mathrm{spd}}(n)$ is small enough. Assuming $\epsilon_{\mathrm{spd}}(n) \leq c'' 2^{-\varepsilon}$ with a constant $0 < c'' \leq \frac{1}{2}$ for concreteness, each of the remaining two terms in (5) must be smaller than $2^{-\varepsilon} - \epsilon_{\mathrm{spd}}(n) \geq (1 - c'') 2^{-\varepsilon}$. Namely,

$$\frac{c\, 2^{2t}}{2^{k_r}} \leq (1 - c'')\, 2^{-\varepsilon} \quad \text{and} \quad \frac{c' 2^t}{2^{k_v}} \leq (1 - c'')\, 2^{-\varepsilon} \tag{6}$$

must hold. Accordingly, in order to attain the desired security level, it is *sufficient* to choose

$$k_r = 2t + \varepsilon \quad \text{and} \quad k_v = t + \varepsilon \tag{7}$$

plus some small positive constants. As a result, the ciphertext overhead of OAEP is upper bounded by

$$k_r + k_v = 3t + 2\varepsilon + O(1). \tag{8}$$

3 Proposed Scheme

3.1 Description

Our construction requires a symmetric-key encryption scheme $\mathsf{SE}_{k_e} = (\mathsf{E}, \mathsf{D})$ and a trapdoor permutation family $\mathcal{P}_n$ as building blocks. The symmetric encryption scheme SE must be length-preserving and passively secure (indistinguishable against passive attacks), and the trapdoor permutation family must be one-way. For formal definitions, we refer to the full version [3].

Let (n, k_e, k_r) be a set of security parameters where n represents the bit-length of the trapdoor permutation, k_e is the key size of the symmetric-key encryption, and k_r is the size of randomness incorporated into the ciphertext. The proposed scheme $\mathsf{PKE} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is the following. See also Figure 1 for a diagram of encryption.

Key Generation $\mathcal{G}$: Given a security parameter $k = (n, k_e, k_r)$ for $n \geq 6k_r$, set parameters k_{m_1} and k_{m_2} so that

$$k_{m_1} \geq 2k_r, \quad k_{m_2} \geq 3k_r, \quad n = k_r + k_{m_1} + k_{m_2} \tag{9}$$

are fulfilled. Then select $(f, f^{-1}) \leftarrow \mathcal{P}_n$ (the trapdoor permutation generator) and hash functions G and H_i for $i = 1, 2, 3, 4$ such that

$$\begin{aligned} &G : \{0,1\}^{k_r + k_{m_1}} \to \{0,1\}^{k_e}, && H_1 : \{0,1\}^{k_r + k_{m_1}} \to \{0,1\}^{k_{m_2}}, \\ &H_2 : \{0,1\}^{k_{m_2}} \to \{0,1\}^{k_r + k_{m_1}}, && H_3 : \{0,1\}^{*} \to \{0,1\}^{k_{m_2}}, \\ &H_4 : \{0,1\}^{k_{m_2}} \to \{0,1\}^{k_r + k_{m_1}}. \end{aligned}$$

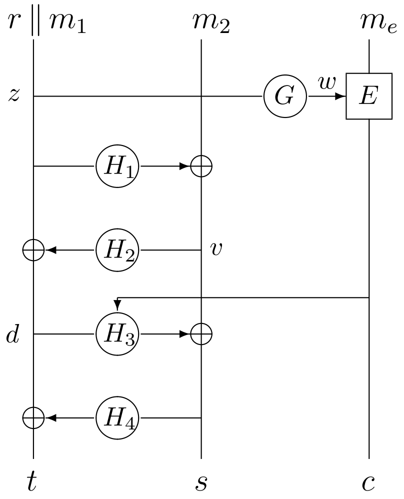

Fig. 1. The diagram of (a part of) encryption. Input message is $m = m_1 \parallel m_2 \parallel m_e \in \{0,1\}^{k_{m_1}} \times \{0,1\}^{k_{m_2}} \times \{0,1\}^*$ and the randomness is $r \in \{0,1\}^{k_r}$. The actual ciphertext is (u, c) where $u = f(t \parallel s)$.

The private-key is f^{-1}. The public-key includes f, SE_{k_e}, and the hash functions with associated parameters.

Encryption $\mathcal{E}$: Given a plaintext $m \in \{0,1\}^*$, first chop it into three blocks, m_1, m_2, and m_e such that

$$m = m_1 \parallel m_2 \parallel m_e \in \{0,1\}^{k_{m_1}} \times \{0,1\}^{k_{m_2}} \times \{0,1\}^*.$$

Then choose random $r \leftarrow \{0,1\}^{k_r}$ and compute

$$\begin{array}{llll} z = r \parallel m_1, & w = G(z), & c = \mathsf{E}_w(m_e), & \\ h_1 = H_1(z), & v = h_1 \oplus m_2, & h_2 = H_2(v), & d = h_2 \oplus z, \\ h_3 = H_3(d \parallel c), & s = h_3 \oplus v, & h_4 = H_4(s), & t = h_4 \oplus d, \end{array}$$

and $u = f(t \parallel s)$. The ciphertext is $(u, c) \in \{0,1\}^n \times \{0,1\}^*$.

Decryption $\mathcal{D}$: Given a ciphertext $(u, c) \in \{0,1\}^n \times \{0,1\}^{k_e}$, compute $y = f^{-1}(u)$ and parse y as $y = t \parallel s \in \{0,1\}^{k_r + k_{m_1}} \times \{0,1\}^{k_{m_2}}$. Then compute the following values:

$$\begin{array}{llll} h_4 = H_4(s), & d = h_4 \oplus t, & h_3 = H_3(d \parallel c), & v = h_3 \oplus s, \\ h_2 = H_2(v), & z = h_2 \oplus d, & h_1 = H_1(z), & m_2 = h_1 \oplus v, \\ w = G(z), & m_e = \mathsf{D}_w(c), & & \end{array}$$

and parse $z = r \parallel m_1 \in \{0,1\}^{k_r} \times \{0,1\}^{k_{m_1}}$. The output is $m_1 \parallel m_2 \parallel m_e$.

3.2 Security and Optimality

The following theorems hold for PKE described in the previous section. A proof sketch is in Section 4 and the complete proof is in [3].

Theorem 1 (Chosen Ciphertext Security). *Suppose* A *is an adversary that runs in time* τ *with at most* q_h *hash queries and* q_d *decryption queries. Then there exist an adversaries* B *that runs in time at most* $\tau + O(q_h^2)$ *and an adversary* C *that runs in time at most* $\tau + O(1)$ *with*

$$\mathbf{Adv}_{\mathsf{A}}^{\text{cca}}(k) \leq \mathbf{Adv}_{\mathsf{C},\mathsf{SE}}^{\text{ind-pa}}(k_e) + 2\mathbf{Adv}_{\mathsf{B},\mathcal{P}}^{\text{owp}}(n) + O(\frac{q_h + q_d}{2^{k_r}}) \ .$$

Note that the number of hash queries includes the ones made through the decryption queries. In an asymptotic sense, Theorem 1 states that the above scheme is semantically secure against adaptive chosen message attacks in the random oracle model if the trapdoor permutation $\mathcal{P}$ is one-way and SE is passively secure.

As it is the case for most OAEP variants, our security reduction includes a quadratic factor q_h^2 in the running time of the adversary against the one-way permutation. It results in demanding larger n which increases the minimal length of the message the scheme can encrypt attaining the optimal overhead. The approach from [19, 14] helps achieving a linear running time if desired.

Theorem 2 (Optimality in Ciphertext Overhead). *If* $\mathbf{Adv}_{\mathsf{C},\mathsf{SE}}^{\text{ind-pa}}(k_e) + 2\mathbf{Adv}_{\mathsf{B},\mathcal{P}}^{\text{owp}}(n) \leq 2^{-(\varepsilon+1)}$ *holds for all adversaries* C *and* B *running in time* 2^t*, then* $k_r = \ell_{\mathsf{oh}} = t + \varepsilon + 4$ *is sufficient for messages of size equal or larger than* $n - k_r$ *bits.*

Note that parameters k_e and n are independent of the overhead and can be set arbitrary to fulfill the condition.

3.3 Notes on Variations

Why not 3 rounds? Consider the 3-round version of our scheme obtained by removing H_4 and simply letting $t = d$. We show that the 3-round version is not simulatable, at least with the technique that constructs a plaintext extractor from the queries to the random oracles. Since the following argument holds regardless of the presence of the extended part c, let us ignore it.

Suppose that the adversary creates two ciphertexts u and u' by randomly choosing t, s, t' and computing $s' = H_3(t) \oplus s \oplus H_3(t')$, $u = f(t \,\|\, s)$, and $u' = f(t' \parallel s')$. Since $H_3(t) \oplus s = H_3(t') \oplus s'$, decrypting u and u' yield the same v. However, such a relation between u and u' can not be detected by the simulator since $H_2(v)$ is not asked. Accordingly the decryption oracle must return random $m_1 \parallel m_2$ and ${m_1}' \parallel {m_2}'$ to answer to the queries on u and u', respectively. Then the adversary asks $H_2(v)$ and obtains h_2. For consistency, it must hold that $h_2 = (r \parallel m_1) \oplus t = (r' \parallel {m_1}') \oplus t'$. However, since m_1 and ${m_1}'$ are randomly chosen before the simulator sees t and t', such a relation can be fulfilled only by chance. The adversary can notice the inconsistency by checking the relation and the simulation should fail.

Including c into a hash other than H_3. We discuss on the variants that includes c into one of the hash functions rather than H_3. In summary, only the inner two hash functions, H_2 and H_3, are the right choice.

- Case of $H_1(z \parallel c)$. This is clearly a wrong choice since (u^*, c^*) and (u^*, c) yield the same m_1.
- Case of $H_2(v \parallel c)$. It is possible to modify the proof of Theorem 1 to show that this variant is also secure.
- Case of $H_4(s \parallel c)$. For this case, we can show that a (powerful) adversary can distinguish the simulation from the reality. The underlying idea is that, given a challenge ciphertext (u^*, c^*), the adversary builds a ciphertext (u, c) that yields the same plaintext without making queries to H_3. Suppose that the adversary finds (t^*, s^*). It obtains $h_4^* = H_4(s^* \parallel c^*)$ and $d^* = h_4^* \oplus t^*$. It then selects arbitrary c and asks $h_4 = H_4(s^* \parallel c)$. Note that c must be different from c^*. It further computes $t = d^* \oplus h_4$ and $u = f(t \parallel s^*)$. Observe that (u, c) recovers d^* and v^* since $d = t \oplus H_4(s^* \parallel c) = d^* \oplus h_4 \oplus H_4(s^* \parallel c) = d^* \oplus h_4 \oplus h_4 = d^*$ and $v = s^* \oplus H_3(d) = s^* \oplus H_3(d^*) = v^*$. Therefore, the selected challenge message is returned if (u, c) is asked to the real decryption oracle. However, since $H_3(d^*)$ has only been defined implicitly and was never directly asked by the adversary, the simulated decryption oracle cannot detect such a case and returns a random message which is noticed by the adversary.

4 Proofs

4.1 Proof of Theorem 1 (sketch)

We proceed in games. Let X_i denote the event that adversary A outputs $\tilde{b} = b$ in Game i.

Game 0. The original CCA game. By definition, we have

$$\Pr[X_0] = \frac{1}{2} \cdot \mathbf{Adv}^{\mathrm{cca}}_{\mathsf{A}}(k) + \frac{1}{2}. \tag{10}$$

Game 1. Modify the challenge oracle so that it returns random u^* that is independent from the challenge messages as follows.

```
Challenge Oracle $(M_0, M_1)$.
C.1  Choose $u^* \leftarrow \{0,1\}^n$.
C.2  Choose $b \leftarrow \{0,1\}$ and split $M_b$ into ${m_1}^*$, ${m_2}^*$ and ${m_e}^*$, accordingly.
     Then choose $w^* \leftarrow \{0,1\}^{k_e}$ and compute $c^* = \mathsf{E}_{w^*}({m_e}^*)$.
C.3  Return $(u^*, c^*)$.
```

For u^*, c^* and w^*, let $(t^*, s^*, d^*, v^*, z^*, h_4^*, h_3^*, h_2^*, h_1^*)$ be a consistent internal state. Let AskH_3^+ denote an event such that $(d^* \parallel c^*)$ is asked to H_3 *after* s^* is asked to H_4. The following bound can be shown.

$$|\Pr[X_0] - \Pr[X_1]| \leq \frac{q_g}{2^{k_r}} + \frac{q_{h_1}}{2^{k_r}} + \frac{q_{h_2}}{2^{k_{m_2}}} + \frac{q_{h_3}}{2^{k_r + k_{m_1}}} + \Pr[\mathsf{AskH}_3^+] \tag{11}$$

It is straightforward to see that distinguishing b breaks the passive security of the symmetric encryption since only the symmetric part is related to b in Game 1. We thus have

$$\Pr[X_1] \leq \frac{1}{2} + \frac{1}{2} \cdot \mathbf{Adv}^{\mathrm{ind\text{-}pa}}_{\mathsf{C},\mathsf{SE}}(k_e)\ , \tag{12}$$

for some suitable adversary C that has similar running time as A.

To bound $\Pr[\mathsf{AskH}_3^+]$, we initiate a new series of sub-games starting from Game 1. In the following games, each random oracle X is simulated with an independent list L_X that is initially empty. When X is first asked on fresh input a, output b is uniformly selected and (a, b) is stored in L_X. If a has been asked before, the corresponding b is read from L_X and returned. By $(a, [b]) \in L_X$, we mean that table L_X includes an entry whose first element is a. If such entry exists, the second element is denoted by b. List L_X is *consistent* for oracle X if every input a is unique in L_X. By $F_{1.i}$ we denote the same event in the following sub-games Game 1.i.

Game 1.0 This game is the same as Game 1. Since this is just a change of notation, we have

$$\Pr[\mathsf{AskH}_3^+] = \Pr[F_{1.0}]\ . \tag{13}$$

Game 1.1 The game is modified so that it *immediately* stops at the moment AskH_3^+ happens. To capture event AskH_3^+, hash oracle H_3 is modified so that it checks whether the query $d \,\|\, c$ equals the value $d^* \,\|\, c^*$ by searching L_{H4} for corresponding s^*.

```
Hash Oracle $H_3(d \| c)$.
A.1 If $(d \| c, [h_3]) \in L_{H3}$, return $h_3$.
A.2 Choose $h_3 \leftarrow \{0,1\}^{k_{m_2}}$ and add $(d \| c, h_3)$ to $L_{H3}$.
A.3 Repeat the following for every entry $(h_4, s)$ in $L_{H4}$.
      (a) Compute $t = d \oplus h_4$, $u = f(t \| s)$.
      (b) If $u = u^*$, abort the game. (event: $F_{1.1}$).
A.4 Return $h_3$.
```

Since this modification does not change the view of the adversary unless AskH_3^+ happens, we have

$$\Pr[F_{1.0}] = \Pr[F_{1.1}] \,. \tag{14}$$

Game 1.2 Modify the decryption oracle so that it returns a *random message* when a decryption query is made on a ciphertext whose associated $d \,\|\, c$ was not yet asked to H_3. Modify H_3 for consistency, too.

```
Decryption Oracle $\mathcal{D}(u, c)$.
D.1 Compute $t \| s = f^{-1}(u)$.
D.2 $h_4 \leftarrow H_4(s)$.
D.3 Let $d = t \oplus h_4$. If $(d \| c, [h_3]) \notin L_{H3}$, go to the next step. Otherwise,
    return $m_1 \| m_2 \| m_e$ computed normally by using $t$, $s$, $d$, and $h_3$.
D.4 Return $m_1 \| m_2 \| m_e$ computed as follows.
      (a) Select $m_1$, $m_2$, and $w$ uniformly and compute $m_e = \mathsf{D}_w(c)$.
      (b) Add $(u, c, w, m_1, m_2)$ to $L_{\mathrm{watch}}$.
```

```
Hash Oracle $H_3(d \| c)$.
A.1 If $(d \| c, [h_3]) \in L_{H3}$, return $h_3$.
A.2 Choose $h_3 \leftarrow \{0,1\}^{k_{m_2}}$ and put $(d \| c, h_3)$ to $L_{H3}$.
A.3 Repeat the following for every entry $(h_4, s)$ in $L_{H4}$.
      (a) Compute $t = d \oplus h_4$, $u = f(t \| s)$, $v = h_3 \oplus s$.
      (b) If $u = u^*$, abort the game. (event: $F_{1.2}$).
      (c) If $(u, c, [w], [m_1], [m_2]) \in L_{\mathrm{watch}}$, do as follows.
            – Select $r \leftarrow \{0,1\}^{k_r}$ and compute $z = r \| m_1$, $h_2 = d \oplus z$, $h_1 =$
              $m_2 \oplus v$.
            – Add $(z, w)$, $(z, h_1)$, and $(v, h_2)$ to $L_G$, $L_{H1}$, and $L_{H2}$, respectively.
            – Remove entry $(u, c, w, m_1, m_2)$ from $L_{\mathrm{watch}}$.
A.4 Return $h_3$.
```

The following bound can be shown.

$$|\Pr[F_{1.1}] - \Pr[F_{1.2}]| \leq \frac{q_d^2}{2^{k_{m_1}}} + \frac{q_{h_1} + q_g}{2^{k_r}} + \frac{q_{h_2}\, q_d}{2^{k_{m_2}}}. \tag{15}$$

Game 1.3 Modify the decryption oracle so that it also returns a *random message* when a decryption query is made on a ciphertext whose associated s was not yet asked to H_4.

```
**Decryption Oracle** $\mathcal{D}(u, c)$.
D.1 Compute $t \,\|\, s = f^{-1}(u)$.
D.2 If $(s, [h_4]) \in L_{H4}$ and $(d \,\|\, c, [h_3]) \in L_{H3}$ for $d = t \oplus h_4$, then return
    $m_1 \,\|\, m_2 \,\|\, m_e$ computed normally by using $t$, $s$, $d$, and $h_3$.
D.3 Otherwise, return $m_1 \,\|\, m_2 \,\|\, m_e$ computed as follows.
    (a) Select $m_1$, $m_2$, and $w$ uniformly and compute $m_e = \mathsf{D}_w(c)$.
    (b) Add $(u, c, w, m_1, m_2)$ to $L_{\mathrm{watch}}$.
```

The following bound can be shown.

$$|\Pr[F_{1.2}] - \Pr[F_{1.3}]| \leq \frac{q_d \, q_{h_3}}{2^{k_r + k_{m_1}}}. \tag{16}$$

Game 1.4 Modify the decryption oracle so that it uses a lookup table instead of computing $t \,\|\, s = f^{-1}(u)$.

```
**Decryption Oracle** $\mathcal{D}(u, c)$.
D.1 If $(u, c, [t], [s]) \in L_X$, then continue the normal decryption procedure by
    using $t$ and $s$ and return the obtained message.
D.2 Otherwise, return random $m_1 \,\|\, m_2 \,\|\, m_e$ computed as follows.
    (a) Select $m_1$, $m_2$, and $w$ uniformly and compute $m_e = \mathsf{D}_w(c)$.
    (b) Add $(u, c, w, m_1, m_2)$ to $L_{\mathrm{watch}}$ and return $m_1 \,\|\, m_2 \,\|\, m_e$.
```

```
**Hash Oracle** $H_3(d \,\|\, c)$.
A.1 If $(d \,\|\, c, [h_3]) \in L_{H3}$, return $h_3$.
A.2 Choose $h_3 \leftarrow \{0,1\}^{k_{m_2}}$ and put $(d \,\|\, c, h_3)$ to $L_{H3}$.
A.3 Repeat the following for every entry $(h_4, s)$ in $L_{H4}$.
    (a) Compute $t = d \oplus h_4$, $u = f(t \,\|\, s)$, $v = h_3 \oplus s$.
    (b) If $u = u^*$, abort the game with status 1 (event: $F_{1.4}$).
    (c) If $(u, c, [w], [m_1], [m_2]) \in L_{\mathrm{watch}}$, do as follows
        – Select $r \leftarrow \{0,1\}^{k_r}$ and compute $z = r \,\|\, m_1$, $h_2 = d \oplus z$, $h_1 =$
          $m_2 \oplus v$.
        – Add $(z, w)$, $(z, h_1)$, and $(v, h_2)$ to $L_G$, $L_{H1}$, and $L_{H2}$, respectively.
        – Remove entry $(u, c, w, m_1, m_2)$ from $L_{\mathrm{watch}}$.
    (d) Put $(u, c, t, s)$ to $L_X$.
A.4 Return $h_3$.
```

```
**Hash Oracle** $H_4(s)$.
B.1 If $(s, [h_4]) \in L_{H4}$, return $h_4$.
B.2 Choose $h_4 \leftarrow \{0,1\}^{k_r + k_{m_1}}$ and put $(s, h_4)$ to $L_{H4}$.
B.3 Repeat the following for every entry $([d], [c], [h_3])$ in $L_{H3}$.
    (a) Let $t = d \oplus h_4$, $v = s \oplus h_3$, and $u = f(t \,\|\, s)$.
    (b) Put $(u, c, t, s)$ to $L_X$.
B.4 Return $h_4$.
```

Since the adversary's view is not influenced by this modification, we have

$$\Pr[F_{1.3}] = \Pr[F_{1.4}]. \tag{17}$$

Game 1.4 does not use f^{-1} and any *-marked internal values at all. Challenge u^* is a random element in $\{0,1\}^n$, and $s^* \parallel t^*$ such that $f(s^* \parallel t^*) = u^*$ can be extracted if $F_{1.4}$ happens. It is thus straightforward to construct adversary B that computes f^{-1} using adversary A that causes $F_{1.4}$. We thus have

$$\Pr[F_{1.4}] \le \mathbf{Adv}^{\mathrm{owp}}_{\mathsf{B},f}(k)\ . \tag{18}$$

The running time of B is bounded by that of A plus $O(q_h^2)$.

From (11), (14), (16), (17), and (18), we have

$$\begin{aligned}\mathbf{Adv}^{\mathrm{cca}}_{\mathsf{A}}(k) \le{}& \mathbf{Adv}^{\text{ind-pa}}_{\mathsf{C,SE}}(k_e) + 2\cdot \mathbf{Adv}^{\mathrm{owp}}_{\mathsf{B},\mathcal{P}}(n)\\ &+ \frac{4(q_{h_1}+q_g)}{2^{k_r}} + \frac{2q_d^2}{2^{k_{m_1}}} + \frac{2q_{h_2}(q_d+1)}{2^{k_{m_2}}} + \frac{2q_{h_3}(q_d+1)}{2^{k_r+k_{m_1}}}\ .\end{aligned}$$

Finally, using $k_{m_1} \ge 2k_r$, $k_{m_2} \ge 3k_r$ and setting $q_h = q_{h_1} + q_{h_2} + q_{h_3} + q_{h_4} + q_g$, this simplifies to the claimed form in the theorem as follows.

$$\begin{aligned}\mathbf{Adv}^{\mathrm{cca}}_{\mathsf{A}}(k) &\le \mathbf{Adv}^{\text{ind-pa}}_{\mathsf{C,SE}}(k_e) + 2\cdot \mathbf{Adv}^{\mathrm{owp}}_{\mathsf{B},\mathcal{P}}(n) + \frac{4q_h}{2^{k_r}} + \frac{2q_d^2}{2^{2k_r}} + \frac{2q_h(q_d+1)}{2^{3k_r}}\\ &\le \mathbf{Adv}^{\text{ind-pa}}_{\mathsf{C,SE}}(k_e) + 2\cdot \mathbf{Adv}^{\mathrm{owp}}_{\mathsf{B},\mathcal{P}}(n) + O(\frac{q_h+q_d}{2^{k_r}}) \end{aligned} \tag{19}$$

4.2 Proof of Theorem 2

Fix ε and t. We require $\mathbf{Adv}^{\mathrm{cca}}_{\mathsf{A}}(k) \le 1/2^{\varepsilon}$ for adversaries A running time in 2^t. Using the explicit bound (19) from the proof of Theorem 1, it is sufficient to set k_r so that

$$\mathbf{Adv}^{\text{ind-pa}}_{\mathsf{C,SE}}(k_e) + 2\cdot \mathbf{Adv}^{\mathrm{owp}}_{\mathsf{B},\mathcal{P}}(n) + \frac{4q_h}{2^{k_r}} + \frac{2q_d^2}{2^{2k_r}} + \frac{2q_h(q_d+1)}{2^{3k_r}} = \frac{1}{2^{\varepsilon}} \tag{20}$$

is fulfilled. By assuming that k_e and n are set to satisfy

$$\mathbf{Adv}^{\text{ind-pa}}_{\mathsf{C,SE}}(k_e) + 2\cdot \mathbf{Adv}^{\mathrm{owp}}_{\mathsf{B},\mathcal{P}}(n) \le 1/2^{\varepsilon+1},$$

it is sufficient to choose k_r such that

$$\frac{4q_h}{2^{k_r}} + \frac{2q_d^2}{2^{2k_r}} + \frac{2q_h(q_d+1)}{2^{3k_r}} \le \frac{1}{2^{\varepsilon+1}}. \tag{21}$$

To achieve semantic security, $q_h/2^{k_r} \leq 1$ and $q_d/2^{k_r} \leq 1$ must hold. Since 2^t upper bounds the running time, $q_h \leq 2^t$ and $q_d \leq 2^t$ must hold, too. By using these bounds, the left side of (21) simplifies to

$$\frac{1}{2^{k_r}}(4q_h + 2q_d + q_h + 1) \leq \frac{8 \cdot 2^t}{2^{k_r}}. \tag{22}$$

Thus we have

$$\frac{8 \cdot 2^t}{2^{k_r}} \leq \frac{1}{2^{\varepsilon+1}},$$

which results in $t + \varepsilon + 4 \leq k_r$. Since $\ell_{\mathsf{oh}} = k_r$ holds for all messages of size equal or larger than $n - k_r$ bits, $\ell_{\mathsf{oh}} = k_r = t + \varepsilon + 4$ is sufficient. It matches the lower bound up to the constant term.

5 Conclusion and Open Problems

We propose a variant of OAEP that attains an optimal overhead in the random oracle model and thereby proved that the lower bound of ciphertext overhead is tight even with respect to IND-CCA security. Open problems include:

- Show the bound without random oracles. In the standard model, the schemes in [7, 18] have the shortest known ciphertext overhead consisting of two group elements that results in $\ell_{\mathsf{oh}} \geq 4t + 2\varepsilon$ bits. It remains as a very interesting open question whether or not the optimality can be achieved without random oracles.
- Optimal ciphertext overhead for shorter messages. We refer to [2] whose (DH-based) schemes offer $\ell_{\mathsf{oh}} \geq 2t + \varepsilon$ for short messages.
- Show that 4-round is *necessary* (or not) in our construction.

References

[1] M. Abe, R. Gennaro, and K. Kurosawa. Tag-KEM/DEM: A new framework for hybrid encryption. *Journal of Cryptology*, 21(1):97–130, 2008.

[2] M. Abe, E. Kiltz, and T. Okamoto. Compact CCA-secure encryption for arbitrary messages. Unpublished Manuscript. Available from the authors. 2007.

[3] M. Abe, E. Kiltz, and T. Okamoto. Chosen ciphertext security with optimal overhead. IACR ePrint Archive 2008/374, September 2, 2008.

[4] M. Bellare and P. Rogaway. Optimal asymmetric encryption. *EUROCRYPT '94*, *LNCS* 950, pages 92–111. Springer-Verlag, 1995.

[5] M. Bellare and P. Rogaway. Code-based game-playing proofs and the security of triple encryption. *Eurocrypt '06*, *LNCS* 4004, pages 409–426. Springer-Verlag, 2006. Full version available from IACR ePrint Archive 2004/331.

[6] B. Bjørstad, A. Dent, and N. Smart. Efficient KEMs with partial message recovery. *Cryptography and Coding 2007*, *LNCS* 4887, pages 233–256. Springer-Verlag, 2007.

[7] X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In *ACM Conference on Computer and Communications Security*, pages 320–329. ACM, 2005. Also available at IACR e-print 2005/288.

[8] J. Coron, H. Handschuh, M. Joye, P. Paillier, D. Pointcheval, and C. Tymen. GEM: A generic chosen-ciphertext secure encryption method. In *CT-RSA 2001*, *LNCS* 2271, pages 263–276. Springer-Verlag, 2002.

[9] J. Coron, H. Handschuh, M. Joye, P. Paillier, D. Pointcheval, and C. Tymen. Optimal chosen-ciphertext secure encryption of arbitrary-length messages. In *PKC 2002*,*LNCS* 2274, pages 17–33. Springer-Verlag, 2002.

[10] J. S. Coron, M. Joye, D. Naccache, and P. Paillier. Universal padding schemes for RSA. In *CRYPTO '02*, *LNCS* 2422, pages 226–241. Springer-Verlag, 2002.

[11] J. S. Coron, J. Patarin, and Y. Seurin. The random oracle model and the ideal cipher model are equivalent. In *CRYPTO '08*, *LNCS* 5157, pages 1–20. Springer-Verlag, 2008.

[12] 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.

[13] Y. Cui, K. Kobara, and H. Imai. A generic conversion with optimal redundancy. In *CT-RSA 2005*, *LNCS* 3376, pages 104–117. Springer-Verlag, 2005.

[14] Y. Dodis, M. Freedman, S. Jarecki, and S. Walfish. Versatile padding schemes for joint signature and encryption. In *ACM CCS'04*. ACM, 2004.

[15] E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. RSA-OAEP is secure under the RSA assumption. In *CRYPTO 2001*, *LNCS* 2139, pages 260–274. Springer-Verlag, 2001.

[16] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.

[17] J. Jonsson. An OAEP variant with a tight security proof. IACR e-print Archive 2002/034, 2002.

[18] E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC'06*, *LNCS* 3876, pages 581–600. Springer-Verlag, 2006.

[19] K. Kobara and H. Imai. OAEP++: A very simple way to apply OAEP to deterministic OW-CPA primitives. IACR ePrint archive, 2002/130, 2002.

[20] Y. Komano and K. Ohta. Efficient universal padding schemes for multiplicative trapdoor one-way permutation. In *CRYPTO '03*, of *LNCS* 2729, pages 366–382. Springer-Verlag, 2003.

[21] T. Okamoto and D. Pointcheval. REACT: Rapid enhanced-security asymmetric cryptosystem transform. In *CT-RSA '2001*, LNCS 2020, pages 159–174. Springer-Verlag, 2001.

[22] D. H. Phan and D. Pointcheval. Chosen-ciphertext security without redundancy. In *Asiacrypt '03*, *LNCS* 2894, pages 1–18. Springer-Verlag, 2003.

[23] D. H. Phan and D. Pointcheval. OAEP 3-round: A generic and secure asymmetric encryption padding. In *Asiacrypt '04*, *LNCS* 3329, pages 63–78. Springer-Verlag, 2004.

[24] Z. Ramzan and L. Reyzin. On the round security of symmetric-key cryptographic primitives. In *CRYPTO 2000*, *LNCS* 1880, pages 376–393. Springer-Verlag, 2000.

[25] V. Shoup. OAEP reconsidered. In *CRYPTO 2001*, *LNCS* 2139, pages 239–259. Springer-Verlag, 2001.