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" [15] 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 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,

Scheme	Ciphertext Overhead	Assumption on TDP	#Feistel rounds
OAEP [3, 14]	$\ell_{\mathsf{oh}} \leq 3t + 2\varepsilon$	SPD-OW	2
OAEP+ [24]	$\ell_{\mathsf{oh}} \leq 3t + 2\varepsilon$	OW	2
PSS-E [9]	$\ell_{\mathsf{oh}} \leq 2t + 2\varepsilon$	SPD-OW	2
PSP2 S-Pad [13]	$\ell_{\mathsf{oh}} \leq 2t + 2\varepsilon$	OW	4
OAEP-3R [22]	$\ell_{\mathsf{oh}} \leq 2t + \varepsilon$	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.

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 [3], 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.

IND-CCA Security via Validity Checking. As in OAEP, a common approach [24, 18, 20, 9, 19, 13] 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 [24, 18] 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 [9, 13] 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 [21, 22]. 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 [12] 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, [3, 16] 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., [11, 7, 8, 1, 5]), 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 [21] but instead use a 4-round Feistel network. (See Figure 1 in Section 4 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. (See the analysis of Case 1 in Section 5.2 for more details.) The fourth round of the Feistel network is then needed to cover m_1. (See Section 4.3 for more detailed arguments about the number of rounds.)

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 [5] 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.* (We do not pursue this line in this paper due to the space limitation.)

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? [23] 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 4.3 for further details.

1.4 Related Work

In Other Models. [21] 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 [10] but the reduction is very costly and a *tight reduction* needed to retain the optimal overhead is highly unlikely. Note that [21] could only present a non-optimal scheme in the random oracle model, which shows the difficulty of achieving the optimality.

In the standard model, the IND-CCA secure schemes in [6, 17] have the shortest known ciphertext overhead consisting of two group elements which require an overhead of $\ell_{\mathsf{oh}} \geq 4t + 2\varepsilon$ bits. It remains as a very interesting open question whether or not the optimality can be achieved without random oracles.

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

2.1 Formal Argument

Let $\mathsf{PKE} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. Public-key pk is associated with the message space $\mathcal{M}$ and the randomness space $\mathcal{R}$ used for encryption. For $(pk, sk) \leftarrow \mathcal{G}(1^k)$, let $C(M)$ for $M \in \mathcal{M}$ denote the set of ciphertexts that recover message M. To obtain a simple form of the lower bound, we restrict ourselves to PKE whose ciphertext overhead is independent of the public-key, message, and the randomness. Namely we assume that

$$\ell^k_{\mathsf{oh}} = |\mathcal{E}_{pk}(M; r)| - |M| \tag{1}$$

is a fixed positive constant for any $pk \in \mathcal{G}(1^k)$, $M \in \mathcal{M}$ and $r \in \mathcal{R}$. We define such ℓ^k_{oh} as the ciphertext overhead with respect to k.

Let A be an adversary attacking the semantic security of PKE. We characterize the adversary by parameters t and ε in such a way that A runs in step 2^t and breaks the semantic security of PKE with advantage at most 2^ε. 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. Later, we will discuss about this abstruction in more detail.

Now consider the following generic attack launched by A.

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 the adversary A with respect to pk is

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

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 $c \in C(M_0)$,

$$\begin{aligned}
\mathbf{Adv}_{\mathsf{A},pk} &\geq \sum_{c \in C(M_0)} p(c)(1-(1-\frac{1}{2^\eta})^{2^t}) \\
&\geq 1-(1-\frac{1}{2^\eta})^{2^t} \\
&\geq \frac{2^t}{2^\eta} - \frac{2^t-1}{2^{2\eta}}. \qquad (3)
\end{aligned}$$

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

$$\begin{aligned}
\mathbf{Adv}_{\mathsf{A}}(k) &= \sum_{pk \in \mathcal{G}(k)} p'(pk) \cdot \mathbf{Adv}_{\mathsf{A},pk} \\
&\geq \sum_{pk \in \mathcal{G}(k)} p'(pk) \cdot \left(\frac{2^t}{2^\eta} - \frac{2^t-1}{2^{2\eta}}\right) \\
&\geq \sum_{pk \in \mathcal{G}(k)} p'(pk) \cdot \left(\frac{2^t}{2^{\ell^k_{\mathsf{oh}}}} - \frac{2^t-1}{2^{2\ell^k_{\mathsf{oh}}}}\right) \\
&\geq \frac{2^t}{2^{\ell^k_{\mathsf{oh}}}} - \frac{2^t-1}{2^{2\ell^k_{\mathsf{oh}}}} \\
&\geq \frac{1}{2} \cdot \frac{2^t}{2^{\ell^k_{\mathsf{oh}}}}. \qquad (4)
\end{aligned}$$

Since we require $\mathbf{Adv}_{\mathsf{A}}(k) \leq 2^{-\varepsilon}$, it holds that

$$2^{-\varepsilon} \geq \frac{1}{2} \cdot \frac{2^t}{2^{\ell^k_{\mathsf{oh}}}} \qquad (5)$$

for $t, \varepsilon \geq 1$. Thus we have the lower bound:

$$\ell^k_{\mathsf{oh}}(t,\varepsilon) \geq t + \varepsilon - 1\ . \qquad (6)$$

The constant -1 is actually close to zero for reasonably large t and ε. 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 $\ell^k_{\mathsf{oh}}(t,\varepsilon) \geq t + \varepsilon$.

The above argument counts encryption as one step. We use the same abstraction in our framework in more general form. That is, we count a fundamental cryptographic operation such as hashing, group operation, or encryption and decryption we are focusing on as one step of computation for the adversary. Hence 2^t is understood as the number of times that the adversary can perform the cryptographic operation in question. Precise assessment is always possible by incorporating scaling factors that represent exact number of steps decided by the target computation model for the cryptographic operations in question. The scaling factors may depend on k. It is important to note that, in this framework, we only consider the meaningful cases where all the scaling factors are independent from t and considerably small for the sake of efficiency of the encryption scheme. We thus wrap these constant factors as

$$\ell^k_{\mathsf{oh}}(t,\varepsilon) \geq t + \varepsilon + O(1)$$

in our arguments. This abstraction of computation provides the same ground in arguing the optimality of the overhead, and the resulting simple form of the lower bound is useful in showing the relation between the power of the adversary and the ciphertext overhead. If needed, one can always do more detailed assessment taking precise scaling factors into account.

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 [14, Th. 1], the advantage of an adversary A making up to q decryption and hash queries is upper bounded by

$$\mathbf{Adv}^{\mathrm{cca}}_{\mathsf{A}}(k) \leq \epsilon_{\mathrm{spd}}(n) + \frac{c\,q^2}{2^{k_r}} + \frac{c'q}{2^{k_v}}\ , \tag{7}$$

where $\epsilon_{\mathrm{spd}}(n)$ is the probability of breaking set partial one-way property of the underlying 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 consider $\epsilon_{\mathrm{spd}}(n)$ 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 (7) must be smaller than $2^{-\varepsilon} - \epsilon_{\mathrm{spd}}(n) \geq (1 - c'')\,2^{-\varepsilon}$. Namely,

$$\frac{c\,2^{2t}}{2^{k_r}} \leq (1 - c'')\,2^{-\varepsilon} \quad \text{and} \quad \frac{c'2^t}{2^{k_v}} \leq (1 - c'')\,2^{-\varepsilon} \tag{8}$$

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

$$k_r = 2t + \varepsilon \quad \text{and} \quad k_v = t + \varepsilon \tag{9}$$

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

$$k_r + k_v = 3t + 2\varepsilon + O(1). \tag{10}$$

Regarding the basic version of OAEP whose security is lowered to CPA by omitting zero-paddings, the ciphertext overhead $k_r = t + \varepsilon + O(1)$ is obtained in the same way based on its concrete security analysis shown as Th.13 of [4].

3 Definitions

3.1 Public Key Encryption

We follow the standard definition of chosen ciphertext security in the random oracle model. Let $\mathsf{PKE} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme where $\mathcal{G}, \mathcal{E}, \mathcal{D}$ are a key generation algorithm, an encryption algorithm, and a decryption algorithm, respectively. Let $\mathcal{H}$ denote the random oracle(s). Let A be an oracle machine that plays the following game.

1. $(pk, sk) \leftarrow \mathcal{G}(1^k)$
2. $(M_0, M_1, \rho) \leftarrow \mathsf{A}^{\mathcal{D}_{sk}, \mathcal{H}}$
3. $b \leftarrow \{0, 1\}$, $c^* \leftarrow \mathcal{E}_{pk}(M_b)$
4. $\tilde{b} \leftarrow \mathsf{A}^{\mathcal{D}_{sk}, \mathcal{H}}(\rho, c^*)$

Here, k is a security parameter and ρ is an internal state of A. Messages M_0 and M_1 must be in the same size and c^* must not be asked to $\mathcal{D}_{sk}$ in the final step. Let $\mathbf{Adv}^{\mathrm{cca}}_{\mathsf{A},\mathsf{PKE}}(k)$ denote the advantage function of A defined by $\mathbf{Adv}^{\mathrm{cca}}_{\mathsf{A},\mathsf{PKE}}(k) = 2 \cdot |\Pr[\hat{b} = b] - 1/2|$. The probability is taken over all coin flips during the game including the selection of $\mathcal{H}$. We say that PKE is CCA-secure if there exists a negligible function ϵ in k such that $\mathbf{Adv}^{\mathrm{cca}}_{\mathsf{A},\mathsf{PKE}}(k) \leq \epsilon$ holds for all A running in polynomial-time in k.

3.2 Symmetric-key Encryption

Let $\mathsf{SE}_{k_e} = (\mathsf{E}, \mathsf{D})$ be a symmetric-key encryption scheme where E is an encryption algorithm and D is a decryption algorithm and its key-space is $\{0,1\}^{k_e}$. Let C be a machine that plays the following game.

1. $w \leftarrow \{0,1\}^{k_e}$
2. $(M_0, M_1, \rho) \leftarrow \mathsf{C}(1^{k_e})$
3. $b \leftarrow \{0,1\}$, $c^* = \mathsf{E}_w(M_b)$
4. $\tilde{b} \leftarrow \mathsf{C}(\rho, c^*)$

Here, ρ is an internal state of C. Messages M_0 and M_1 must be in the same size. Let $\mathbf{Adv}^{\text{ind-pa}}_{\mathsf{C},\mathsf{SE}}(k_e)$ denote the advantage of C defined by $\mathbf{Adv}^{\text{ind-pa}}_{\mathsf{C},\mathsf{SE}}(k_e) = 2 \cdot |\Pr[\tilde{b} = b] - 1/2|$. We say that SE is passively secure if there exists a negligible function ϵ in k_e such that $\mathbf{Adv}^{\text{ind-pa}}_{\mathsf{C},\mathsf{SE}}(k_e) \leq \epsilon$ holds for all C running in polynomial-time in k_e.

For our construction, we require SE_{k_e} to be length-preserving where the length of a message and its ciphertext are the same. A simple one-time pad, whose key is generated through a pseudorandom number generator in practice, fulfills this requirement.

3.3 Trapdoor One-Way Permutations

Let $\mathcal{P}_n$ be a family of trapdoor permutations over $\{0,1\}^n$. By $(f, f^{-1}) \leftarrow \mathcal{P}_n$, we mean that a permutation f and its inverse function f^{-1} over $\{0,1\}^n$ are efficiently and uniformly chosen. Both f and f^{-1} must be efficiently computable. The inverse function f^{-1} is called a trapdoor.

Let B be a machine that plays the following game.

1. $(f, f^{-1}) \leftarrow \mathcal{P}_n$, $X \leftarrow \{0,1\}^n$, $Y = f(X)$.
2. $\tilde{X} \leftarrow \mathsf{B}(f, Y)$

By $\mathbf{Adv}^{\mathrm{owp}}_{\mathsf{B},\mathcal{P}}(n)$, we denote the probability of $\tilde{X} = X$, which is taken over all the coin flips during the game. We say that $\mathcal{P}_n$ is one-way if there exists a negligible function ϵ in n such that $\mathbf{Adv}^{\mathrm{owp}}_{\mathsf{B},\mathcal{P}}(n) \leq \epsilon$ holds for any adversary running in time τ.

4 Proposed Scheme

Our scheme requires symmetric-key encryption schemes and trapdoor permutation families as building blocks. The symmetric encryption schemes must be length-preserving and passively secure (indistinguishable against passive attacks), and the trapdoor permutation family must be one-way.

4.1 Description

We assume that a trapdoor one-way permutation family $\mathcal{P}_n$ and a length-preserving symmetric-key encryption scheme $\mathsf{SE}_{k_e} = (\mathsf{E}, \mathsf{D})$ are available, for $n, k_e \in \mathbb{N}$. Let $k_r \in \mathbb{N}$ be a parameter that represents the size of randomness used for encryption. (These parameters are treated independently so that their individual impact to the ciphertext overhead can be explicitly stated. It is possible to treat them as functions of a single security parameter as usual.)

The proposed scheme $\mathsf{PKE} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is as follows. See Figure 1 for graphical presentation. Also see Section 4.3 for a discussion about variants.

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

$$k_{m_1} \geq 2k_r, \quad k_{m_2} \geq 3k_r, \quad n = k_r + k_{m_1} + k_{m_2} \tag{11}$$

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

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

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

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

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

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

and $u = f(t \,\|\, s)$. 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 $y = t \,\|\, s \in \{0,1\}^{k_r+k_{m_1}} \times \{0,1\}^{k_{m_2}}$. Then compute the following values:

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

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

4.2 Security and Optimality

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*

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

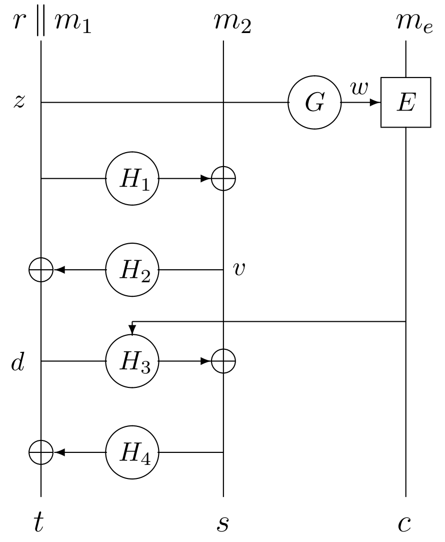

Fig. 1. The diagram of (a part of) encryption. Input message is $m = m_1 \,\|\, m_2 \,\|\, m_e \in \{0,1\}^{k_{m_1}} \times \{0,1\}^{k_{m_2}} \times \{0,1\}^*$ and the randomness is $r \in \{0,1\}^{k_r}$. The actual ciphertext is (u, c) where $u = f(t \,\|\, s)$.

The theorem is proven in Section 5.1. Note that the number of hash queries includes the ones made through 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 $\mathcal{P}$ 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 [18, 13] helps achieving a linear running time if desired.

Theorem 2 (Optimality in Ciphertext Overhead). *If* $\mathbf{Adv}^{\text{ind-pa}}_{\mathsf{C},\mathsf{SE}}(k_e) + 2\mathbf{Adv}^{\text{owp}}_{\mathsf{B},\mathcal{P}}(n) \le 2^{-(\varepsilon+1)}$ *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.*

A simple proof is shown in Section 5.3. Note that parameters k_e and n are independent of the overhead and can be set arbitrary to fulfill the condition.

4.3 A Note on Variants

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. Observe that, if both $h_2 = H_2(v)$ and $h_3 = H_3(d)$ are asked, one can compute $t = d$, $s = h_3 \oplus v$, $u = f(t \,\|\, s)$. Thus we can simulate the decryption oracle for query u only by looking at these hash queries. On the other hand, if one of these hash queries are not asked, the decryption oracle has to return random $m_1 \,\|\, m_2$.

Suppose that the adversary creates two ciphertexts, u and u' as follows. Randomly choose t, s, t' and compute $s' = H_3(t) \oplus s \oplus H_3(t')$, $u = f(t \,\|\, s)$, $u' = f(t' \,\|\, s')$. Clearly, they yield the same v as $v = H_3(t) \oplus s = H_3(t') \oplus s'$. Apparently, the relation between u and u' cannot be detected since $H_2(v)$ is not asked. Hence the decryption oracle returns random $m_1 \,\|\, m_2$ and ${m_1}' \,\|\, {m_2}'$ to the decryption queries for u and u', respectively. Now, the adversary asks $H_2(v)$ and obtains h_2. For consistency, it must hold that $h_2 = (r \,\|\, m_1) \oplus t = (r' \,\|\, {m_1}') \oplus t'$. However, since m_1 and ${m_1}'$ are randomly chosen before the simulator sees t and t', such a relation is fulfilled only with negligible probability. The adversary can notice the inconsistency by checking the relation. Thus the simulation fails.

This shows the difficulty of 3-round design but does not show the impossibility. It is an interesting open problem to show whether it is indeed impossible or not.

Including c into a hash other than H_3. We here discusses some variants obtained by including c into a hash function rather than H_3. In summary, H_1 and H_4 are not the right place to input c.

Variant 1: $H_2(v \,\|\, c)$**.** The proof of Theorem 1 shows that this variant is also secure.

Variant 2: $H_1(z \,\|\, c)$**.** This is clearly a wrong choice since (u^*, c^*) and (u^*, c) yields the same m_1.

Variant 3: $H_4(s \,\|\, c)$**.** 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 asking to H_3. Suppose that the adversary finds (t^*, s^*). It obtains $h_4^* = H_4(s^* \,\|\, c^*)$ and $d^* = h_4^* \oplus t^*$. It then selects arbitrary c and ask $h_4 = H_4(s^* \,\|\, c)$. Note that c must be different from c^*. It further computes $t = d^* \oplus h_4$ and $u = f(t \,\|\, s^*)$. Observe that (u, c) recovers d^* and v^* since $d = t \oplus H_4(s^* \,\|\, c) = d^* \oplus h_4 \oplus H_4(s^* \,\|\, c) = d^* \oplus h_4 \oplus h_4 = d^*$ 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 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. Note that this attack is prevented by giving the feedback (also) to H_3 since c must be different from c^* to have (u, c) different from (u^*, c^*).

5 Proofs

5.1 Proof of Theorem 1

We follow the game based proof method [11] starting from the original CCA game as defined in Section 3.1. In the following, let X_i denote the event that A outputs $\tilde{b} = b$ in Game i.

Game 0. Let A be an adversary in the original CCA game. By definition, we have

$$\Pr[X_0] = \frac{1}{2} \cdot \mathbf{Adv}^{\mathrm{cca}}_{\mathsf{A}}(k) + \frac{1}{2}. \tag{12}$$

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 $(u^*, c^*)$.
```

Observe that, for every $h_4^* \in \{0,1\}^{k_r+k_{m_1}}$ and $h_3^* \in \{0,1\}^{k_{m_2}}$, there exist h_2^* and h_1^* that make $({m_1}^*, {m_2}^*, t^*, s^*, h_3^*, h_4^*)$ consistent for some r^*, i.e., $h_2^* = d^* \oplus (r^* \,\|\, {m_1}^*)$ and $h_1^* = v^* \oplus {m_2}^*$ hold for $d^* = h_4^* \oplus t^*$ and $v^* = s^* \oplus h_3^*$. In Game 0, however, $H_2(v^*)$ and $H_1(r^* \,\|\, {m_1}^*)$ will be defined with random fresh values and it is unlikely that the consistent value h_2^* and h_1^* are assigned to their output. The same is true for w^* and $G(r^* \,\|\, {m_1}^*)$. Accordingly, the view of the adversary in Game 0 and Game 1 is the same unless v^* is asked to H_2 or $z^*(= r^* \,\|\, {m_1}^*)$ is asked to H_1 or G.

Let AskH_2^* denote the event that v^* (defined for some fixed t^*, s^*, h_3^*, h_4^*) is asked to H_2. Let AskH_2^- (and AskH_2^+) denote the special case of event AskH_2^* such that v^* is asked to H_2 *before* (and *after*, resp.) $d^* \,\|\, c^*$ is asked to H_3. Define AskH_1^*, AskH_1^-, AskH_1^+, AskG^*, AskG^-, and AskG^+ in the same manner with respect to event that z^* is asked to H_1 and G. Finally, define AskH_3^*, AskH_3^-, and AskH_3^+ accordingly with respect to event that $d^* \,\|\, c^*$ is asked to H_3 before or after s^* is asked to H_4. From the above observation, we have

$$|\Pr[X_0] - \Pr[X_1]| \le \Pr[\mathsf{AskG}^* \vee \mathsf{AskH}_1^* \vee \mathsf{AskH}_2^*] \;. \tag{13}$$

Observe that the "$^-$" events can only happen coincidentally due to the randomness of the hash outputs. Also observe that the "$^+$" events happen, by definition, only if the previous event happens. Thus we obtain

$$\begin{aligned}\Pr[\mathsf{AskG}^* \vee \mathsf{AskH}_1^* \vee \mathsf{AskH}_2^*] &\le \Pr[\mathsf{AskG}^- \vee \mathsf{AskH}_1^- \vee \mathsf{AskH}_2^-] + \Pr[\mathsf{AskG}^+ \vee \mathsf{AskH}_1^+ \vee \mathsf{AskH}_2^+] \\ &\le \Pr[\mathsf{AskG}^- \vee \mathsf{AskH}_1^- \vee \mathsf{AskH}_2^-] + \Pr[\mathsf{AskH}_3^*] \\ &\le \frac{q_g}{2^{k_r}} + \frac{q_{h_1}}{2^{k_r}} + \frac{q_{h_2}}{2^{k_{m_2}}} + \frac{q_{h_3}}{2^{k_r+k_{m_1}}} + \Pr[\mathsf{AskH}_3^+] \;. \end{aligned} \tag{14}$$

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:

$$\Pr[X_1] \le \frac{1}{2} + \frac{1}{2} \cdot \mathbf{Adv}_{\mathsf{C},\mathsf{SE}}^{\text{ind-pa}}(k_e) \;, \tag{15}$$

for some suitable adversary C that has about the same running time as A.

From (12), (13), (14), and (15), we obtain

$$\mathbf{Adv}_{\mathsf{A}}^{\text{cca}}(k) \le \mathbf{Adv}_{\mathsf{C},\mathsf{SE}}^{\text{ind-pa}}(k_e) + 2\left(\frac{q_g + q_{h_1}}{2^{k_r}} + \frac{q_{h_2}}{2^{k_{m_2}}} + \frac{q_{h_3}}{2^{k_r+k_{m_1}}} + \Pr[\mathsf{AskH}_3^+]\right) . \tag{16}$$

It remains to estimate $\Pr[\mathsf{AskH}_3^+]$ which will be related to the advantage of breaking the one-way property of f. We initiate a new series of sub-games starting from Game 1, whose goal is a game where a reduction from causing AskH_3^+ to breaking one-way property of f is trivial.

In the following games, each random oracle is simulated with an independent list. Let L_X denote the list for hash oracle X, which is initially empty. By $b \leftarrow X(a)$ we mean that a is asked to oracle X and b is returned as output. When X is first asked on fresh input a, output b is uniformly selected and (a, b) is stored to L_X. If a has been asked before, corresponding b is read from L_X and returned. By $(a, [b]) \in L_X$, we mean that table L_X includes an entry (α, β) where $\alpha = a$. It also means that, if such an entry exists, β is referred as b. List L_X is *consistent* for oracle X if every input a is unique in L_X.

Game 1.0 This game is the same as Game 1. By $F_{1.0}$, we denote the event that AskH_3^+ has happened, i.e., (recalling the above definition), $F_{1.0} =$ "$d^* \,\|\, c^*$ is asked to H_3 after s^* is asked to H_4". Since this is just a change of notation, we have

$$\Pr[\mathsf{AskH}_3^+] = \Pr[F_{1.0}] \tag{17}$$

By $F_{1.i}$ for $i = 1, ...,$ we denote the same event in the following sub-games Game 1.i.

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

$$\Pr[F_{1.0}] = \Pr[F_{1.1}] \ . \tag{18}$$

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. (If $d \,\|\, c$ was already asked, it decrypts normally as in the last game, i. e., follow the Feistel network in the reverse order by using the corresponding h_4 and h_3.) It further records the returned random messages into a list, say L_{watch}, and assign consistent hash values when they are asked later. We also modify H_3 oracle to avoid inconsistency. Details are as follows. Let L_{watch} be initially empty.

```
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 $L_{\text{watch}}$.

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 $(u, c, [w], [m_1], [m_2]) \in L_{\text{watch}}$, 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 $L_{\text{watch}}$.
A.4 Return $h_3$.
```

The only difference to the last game is the way that the decryption queries are handled on a ciphertext (u, c) whose associated value $d \,\|\, c$ was not queried to H_3. However, the view of the adversary does not change unless it queries H_3 on $d \,\|\, c$. In that case Step A.3c of the above implementation of H_3 will find an entry of the form $(u, c, [w], [m_1], [m_2])$ in L_{watch} and can determine h_3 to make the (already fixed) output of the decryption query consistent. Let $\mathsf{HashErrA}$ denote the event that the hash assignments in Step A.3c yield any inconsistent hash table. Unless $\mathsf{HashErrA}$ happens, the distribution of the adversary's view is unchanged. Thus we have

$$|\Pr[F_{1.1}] - \Pr[F_{1.2}]| \le \Pr[\mathsf{HashErrA}]. \tag{19}$$

The following is our main technical lemma. Its proof will be given in Section 5.2.

Lemma 3.

$$\Pr[\mathsf{HashErrA}] \le \frac{q_d^2}{2^{k_{m_1}}} + \frac{q_{h_1} + q_g}{2^{k_r}} + \frac{q_{h_2}\, q_d}{2^{k_{m_2}}}. \tag{20}$$

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 $L_{\text{watch}}$.
```

Observe that, in the previous game, a decryption query (u, c) with a fresh h_4 causes a normal decryption process if $d \,\|\, c$ is already in L_{H3}. In the current game, however, such a query is answered with a random message. Unless such a query is made, nothing is changed. Since h_4 has been assigned randomly, $d = t \oplus h_4$ distributes uniformly over $\{0,1\}^{k_r + k_{m_1}}$ and therefore the event that $d \,\|\, c \in L_{H3}$ happens only by chance. Thus we have

$$|\Pr[F_{1.2}] - \Pr[F_{1.3}]| \le \frac{q_d\, q_{h_3}}{2^{k_r + k_{m_1}}}. \tag{21}$$

Note that after this modification, the regular decryption procedure is invoked only when both $h_4 = H_4(s)$ and $h_3 = H_3(d \,\|\, c)$ have been asked before the corresponding decryption query is made.

Game 1.4 Modify the decryption oracle so that it does not compute $t \,\|\, s = f^{-1}(u)$ anymore. Instead, it uses a lookup table, say L_X, to obtain t and s. The table is maintained by oracles H_3 and H_4 so that it contains the t, s values of all possible decryption queries whose H_3 and H_4 queries are already made. Details as follows.

This modification is just conceptual and does not change the adversary's view. Hence we have

$$\Pr[F_{1.3}] = \Pr[F_{1.4}]. \tag{22}$$

```
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 $L_{\text{watch}}$ 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 $(u, c, [w], [m_1], [m_2]) \in L_{\text{watch}}$, 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 $L_{\text{watch}}$.
    (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 $h_4$.
```

Now observe that Game 1.4 does not use f^{-1} at all. It does not use any *-marked internal values either. The challenge u^* is a random element in $\{0,1\}^n$, and s^* and t^* with $f(s^* \,\|\, t^*) = u^*$ become available exactly when event $F_{1.4}$ happens. It is thus straightforward to show that the adversary that causes event $F_{1.4}$ with some probability can be used to compute f^{-1} with the same probability. We thus have

$$\Pr[F_{1.4}] \le \mathbf{Adv}^{\text{owp}}_{\mathsf{B},f}(k) \ , \tag{23}$$

for a suitable adversary B whose running time is bounded by the running time of A plus $O(q_h^2)$.

To complete the proof we collect the probabilities relating the different games. From (18), (20), (21), (22) and (23), we have

$$\Pr[\mathsf{AskH}_3^+] \le \frac{q_d^2}{2^{k_{m_1}}} + \frac{q_{h_1} + q_g}{2^{k_r}} + \frac{q_{h_2}\, q_d}{2^{k_{m_2}}} + \frac{q_d\, q_{h_3}}{2^{k_r + k_{m_1}}} + \mathbf{Adv}^{\text{owp}}_{\mathsf{B},\mathcal{P}}(n). \tag{24}$$

From (16) and (24),

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

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

$$\begin{aligned} \mathbf{Adv}^{\text{cca}}_{\mathsf{A}}(k) &\le \mathbf{Adv}^{\text{ind-pa}}_{\mathsf{C},\mathsf{SE}}(k_e) + 2 \cdot \mathbf{Adv}^{\text{owp}}_{\mathsf{B},\mathcal{P}}(n) + \frac{4q_h}{2^{k_r}} + \frac{2q_d^2}{2^{2k_r}} + \frac{2q_h(q_d + 1)}{2^{3k_r}} \\ &\le \mathbf{Adv}^{\text{ind-pa}}_{\mathsf{C},\mathsf{SE}}(k_e) + 2 \cdot \mathbf{Adv}^{\text{owp}}_{\mathsf{B},\mathcal{P}}(n) + O\Big(\frac{q_h + q_d}{2^{k_r}}\Big) \end{aligned} \tag{25}$$

as claimed.

5.2 Proof of Lemma 3

First note that the assignments to H_1, H_2, and G take place at most q_d times. We evaluate the probability of the failure in the assignments throughout the game.

Case 1: Assignment of $h_1 = H_1(z)$ fails (event: $\mathsf{FailH1}$). We first note a simple way to bound $\mathsf{FailH1}$. Recall that $z = r \,\|\, m_1$. Since r is chosen randomly, it appears among the q_{h_1} queries to H_1 with probability $\frac{q_{h_1}}{2^{k_r}}$. Since there are at most q_d assignments, the probability of $\mathsf{FailH1}$ is at most $\frac{q_d\, q_{h_1}}{2^{k_r}}$. This loose bound, however, would result in setting $k_r \geq 2k$, which is not optimal.

Our idea to obtain a tighter bound is to analyze the probability of a conflict on $r \,\|\, m_1$ for each m_1 separately. To this end we first classify all H_1 queries into classes indexed by some m_1. For every string str in $\{0,1\}^{k_{m_1}}$, let Q_{str} denote the number of H_1 queries of the form $r \,\|\, \mathsf{str}$ for some string $r \in \{0,1\}^{k_r}$. Note that $\sum_{\mathsf{str}\in\{0,1\}^{k_{m_1}}} Q_{\mathsf{str}} \leq q_{h_1}$. Since r is uniformly chosen in Step A.3c, the probability that a string $r \,\|\, \mathsf{str}$ for a fixed str has been queried is at most $Q_{\mathsf{str}}/2^{k_r}$. Let MColl denote the event that the same m_1 is selected twice (or more) in Step D.4 (during the whole execution of the experiment). Since m_1 is uniformly and independently chosen in every execution of Step A.3c, we have

$$\Pr[\mathsf{MColl}] \leq \frac{q_d^2}{2^{k_{m_1}}}. \tag{26}$$

Assume that MColl does not happen and therefore each m_1 selected in Step D.4 is unique. Then, by the union bound, the total probability of the failure throughout the game is

$$\Pr[\mathsf{FailH1} \mid \neg\mathsf{MColl}] \leq \sum_{\mathsf{str}\in\{0,1\}^{k_{m_1}}} \frac{Q_{\mathsf{str}}}{2^{k_r}} \leq \frac{q_{h_1}}{2^{k_r}}. \tag{27}$$

Case 2: Assignment of $w = G(r \,\|\, m_1)$ fails (event: FailG). By the same argument as above, conditioned on $\neg\mathsf{MColl}$, the total probability of failure during the simulation is

$$\Pr[\mathsf{FailG} \mid \neg\mathsf{MColl}] \leq \frac{q_g}{2^{k_r}}. \tag{28}$$

Case 3: Assignment of $h_2 = H_2(v)$ fails (event: $\mathsf{FailH2}$). Observe that for every two distinct queries (u, c) and (u', c') in L_{watch} appearing in Step A.3c, it must be the case that $c = c'$ and $s \neq s'$ to fulfill $(u, c) \neq (u', c')$. Accordingly, we have $v = s \oplus h_3 \neq s' \oplus h_3 = v'$. Thus, all v appearing within a single execution of Step A.3c are distinct.[1] [2]

Consider one execution of the H_3 oracle on some fixed input $d \,\|\, c \in L_{H3}$ at Step A.2, i. e., at the point where value h_3 is chosen randomly. Let S_{H_2} denote the set of all values v stored in L_{H2} at that moment. Let $S_v(d \,\|\, c)$ denote the set of all v used to compute h_1 in Step A.3c. Let $S_s(d \,\|\, c)$ be the set of all values s from Step A.3a used to compute $v \in S_v(d \,\|\, c)$ in Step A.3c. (Note that $S_s(d \,\|\, c)$ may be a proper subset of the set of all s appearing in L_{H4}.) Now event $\mathsf{FailH2}$ occurs if and only if there exists $s \in S_s(d \,\|\, c)$ and $v \in S_{H_2}$ such that $v = s \oplus h_3$.

Since h_3 is chosen uniformly in Step A.2, all elements in $S_s(d \,\|\, c)$ and S_{H_2} are independent of h_3. We can therefore analyze the probability that event $\mathsf{FailH2}$ happens solely based on the uniform

[1] This is where we use the structure that c is given to H_3. If c is given only to H_4, for instance, we could no longer conclude $v \neq v'$.

[2] The same argument is possible for a variant such that c is given to H_2 instead, i.e., $H_2(v \,\|\, c)$. In that case, we argue that $v = v'$ happens only if $u = u'$. Then $c \neq c'$ must hold and the inputs to H_2 are different, i.e., $v \,\|\, c \neq v' \,\|\, c'$.

choice of $h_3 \in \{0,1\}^{k_{m_2}}$. For a single execution of Step A.3c, the error probability is thus upper bound by $|S_{H_2}| \cdot |S_s(d \,\|\, c)|/2^{k_{m_2}}$. By summing up $|S_s(d \,\|\, c)|$ for every query $d \,\|\, c$ appearing during the game, we have

$$\sum_{d \,\|\, c \in L_{H3}} |S_s(d \,\|\, c)| \leq |L_{\text{watch}}| \leq q_d \tag{29}$$

since Step A.3c is done at most once for every (u, c) in L_{watch}. Also observe that $|S_{H_2}| \leq q_{h_2}$ holds for every execution of Step A.3c. Therefore, throughout the game, we have

$$\Pr[\mathsf{FailH2}] \leq \sum_{d \,\|\, c \in L_{H3}} \frac{|S_{H_2}| \cdot |S_s(d \,\|\, c)|}{2^{k_{m_2}}} \leq \frac{|S_{H_2}|}{2^{k_{m_2}}} \cdot \sum_{d \,\|\, c \in L_{H3}} |S_s(d \,\|\, c)| \leq \frac{q_{h_2}\, q_d}{2^{k_{m_2}}}. \tag{30}$$

Combining (26), (27), (28) and (30), we can upper bound $\Pr[\mathsf{HashErrA}]$ as

$$\begin{aligned}
\Pr[\mathsf{HashErrA}] &\leq \Pr[\mathsf{FailH1} \vee \mathsf{FailG} \vee \mathsf{FailH2}] \\
&\leq \Pr[\mathsf{MColl}] + \Pr[\mathsf{FailH1} \vee \mathsf{FailG} \mid \neg\mathsf{MColl}] + \Pr[\mathsf{FailH2}] \\
&\leq \Pr[\mathsf{MColl}] + \Pr[\mathsf{FailH1} \mid \neg\mathsf{MColl}] + \Pr[\mathsf{FailG} \mid \neg\mathsf{MColl}] + \Pr[\mathsf{FailH2}] \\
&\leq \frac{q_d^2}{2^{k_{m_1}}} + \frac{q_{h_1} + q_g}{2^{k_r}} + \frac{q_{h_2}\, q_d}{2^{k_{m_2}}}. \qquad (31)
\end{aligned}$$

This concludes the proof of Lemma 3.

5.3 Proof of Theorem 2

First note that k_r is the only parameter that controls the ciphertext overhead in our scheme, i.e., $\ell_{\mathsf{oh}} = k_r$, for all messages of size equal or larger than $n - k_r$ bits. (For shorter messages the ciphertext overhead is no longer optimal.)

Fix ε and t. We compute a bound on k_r such that $\mathbf{Adv}_{\mathsf{A}}^{\text{cca}}(k) \leq 1/2^{\varepsilon}$ for adversaries A running time in 2^t. Using the explicit bound (25) from the proof of Theorem 1, it is sufficient to set k_r so that

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

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

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

it is sufficient to choose k_r such that

$$\frac{4q_h}{2^{k_r}} + \frac{2q_d^2}{2^{2k_r}} + \frac{2q_h(q_d+1)}{2^{3k_r}} \leq \frac{1}{2^{\varepsilon+1}}. \tag{32}$$

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 (32) simplifies to

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

Thus we have

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

which results in

$$t + \varepsilon + 4 \leq k_r.$$

Accordingly, $\ell_{\mathsf{oh}} = k_r = t + \varepsilon + 4$ is sufficient and it matches the lower bound up to the constant term as stated in the theorem.

6 Conclusion and Open Problems

We present a variant of OAEP that attains the optimal overhead in the random oracle model and thereby proved that the lower bound of the ciphertext overhead is tight even with respect to CCA security.

Open problems:

- Does the same bound hold without random oracles? Among some CCA-secure schemes without random oracles, the schemes [6, 17] have the shortest known overhead consisting of two group elements which require an overhead of $\ell_{\mathsf{oh}} \geq 4t + 2\varepsilon$.
- Construct a scheme that encrypts very short messages with optimal overhead. Schemes based on general one-way permutations can never offer the optimal overhead for short messages. For the state of art in this issue, we refer to [2] whose scheme offers $\ell_{\mathsf{oh}} \geq 2t + \varepsilon$ for messages of arbitrary (small) length.
- Show 4-round is *necessary* (or not) in our construction. See Section 4.3 for discussion.

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, T. Okamoto, and E. Kiltz. Compact cca-secure encryption for arbitrary messages. Unpublished Manuscript. Available from the authors., 2007.

[3] M. Bellare and P. Rogaway. Optimal asymmetric encryption. In A. D. Santis, editor, *Advances in Cryptology — EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111. Springer-Verlag, 1995.

[4] M. Bellare and P. Rogaway. Code-based game-playing proofs and the security of triple encryption. In *Advances in Cryptology — Eurocrypt '06*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer-Verlag, 2006. Full version available from IACR ePrint Archive 2004/331.

[5] B. Bjørstad, A. Dent, and N. Smart. Efficient KEMs with partial message recovery. In *Cryptography and Coding 2007*, volume 4887 of *Lecture Notes in Computer Science*, pages 233–256. Springer-Verlag, 2007.

[6] 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.

[7] 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*, volume 2271 of *Lecture Notes in Computer Science*, pages 263–276. Springer-Verlag, 2002.

[8] 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*, volume 2274 of *Lecture Notes in Computer Science*, pages 17–33. Springer-Verlag, 2002.

[9] J. S. Coron, M. Joye, D. Naccache, and P. Paillier. Universal padding schemes for RSA. In M. Yung, editor, *Advances in Cryptology — CRYPTO '02*, volume 2422 of *Lecture Notes in Computer Science*, pages 226–241. Springer-Verlag, 2002.

[10] J. S. Coron, J. Patarin, and Y. Seurin. The random oracle model and the ideal cipher model are equivalent. In *Advances in Cryptology — CRYPTO '08*, volume 5157 of *Lecture Notes in Computer Science*, pages 1–20. Springer-Verlag, 2008.

[11] 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.

[12] Y. Cui, K. Kobara, and H. Imai. A generic conversion with optimal redundancy. In A. Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 104–117. Springer-Verlag, 2005.

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

[14] E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. RSA-OAEP is secure under the RSA assumption. In J. Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 260–274. Springer-Verlag, 2001.

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

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

[17] E. Kiltz. Chosen-ciphertext security from tag-based encryption. In S. Halevi and T. Rabin, editors, *Theory of Cryptography Conference – TCC'06*, volume 3876 of *Lecture Notes in Computer Science*, pages 581–600. Springer-Verlag, 2006.

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

[19] Y. Komano and K. Ohta. Efficient universal padding schemes for multiplicative trapdoor one-way permutation. In *Advances in Cryptology — CRYPTO '03*, volume 2729 of *Lecture Notes in Computer Science*, pages 366–382. Springer-Verlag, 2003. Full version available from IACR ePrint Archive 2004/002.

[20] T. Okamoto and D. Pointcheval. REACT: Rapid enhanced-security asymmetric cryptosystem transform. In *CT-RSA '2001*, Lecture Notes in Computer Science. Springer-Verlag, 2001.

[21] D. H. Phan and D. Pointcheval. Chosen-ciphertext security without redundancy. In *Advances in Cryptology — Asiacrypt '03*, volume 2894 of *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag, 2003.

[22] D. H. Phan and D. Pointcheval. OAEP 3-round: A generic and secure asymmetric encryption padding. In P. J. Lee, editor, *Advances in Cryptology — Asiacrypt '04*, volume 3329 of *Lecture Notes in Computer Science*, pages 63–78. Springer-Verlag, 2004.

[23] Z. Ramzan and L. Reyzin. On the round security of symmetric-key cryptographic primitives. In M. Bellare, editor, *Advances in Cryptology — CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 376–393. Springer-Verlag, 2000.

[24] V. Shoup. OAEP reconsidered. In *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 239–259. Springer-Verlag, 2001.