論文 / 著書情報
Article / Book Information

Title	General Conversion for Obtaining Strongly Existentially Unforgeable Signatures
Authors	Isamu Teranishi, Takuro Oyama, Wakaha Ogata
Citation	IEICE Transaction, Vol. E91-A, No. 1, pp. 94-106
Pub. date	2008, 1
URL	http://search.ieice.org/
Copyright	(c) 2008 Institute of Electronics, Information and Communication Engineers

IEICE TRANS. FUNDAMENTALS, VOL.E91–A, NO.1 JANUARY 2008

PAPER Special Section on Cryptography and Information Security

General Conversion for Obtaining Strongly Existentially Unforgeable Signatures*

Isamu TERANISHI[†,††a)], **Takuro OYAMA**[††b)], *Nonmembers*, *and* **Wakaha OGATA**[††c)], *Member*

SUMMARY We say that a signature scheme is strongly existentially unforgeable (SEU) if no adversary, given message/signature pairs adaptively, can generate a signature on a new message or a new signature on a previously signed message. We propose a general and efficient conversion in the standard model that transforms a secure signature scheme to SEU signature scheme. In order to construct that conversion, we use a chameleon commitment scheme. Here a chameleon commitment scheme is a variant of commitment scheme such that one can change the committed value after publishing the commitment if one knows the secret key. We define the chosen message security notion for the chameleon commitment scheme, and show that the signature scheme transformed by our proposed conversion satisfies the SEU property if the chameleon commitment scheme is chosen message secure. By modifying the proposed conversion, we also give a general and efficient conversion in the random oracle model, that transforms a secure signature scheme into a SEU signature scheme. This second conversion also uses a chameleon commitment scheme but only requires the key only attack security for it.

key words: *signature scheme, strong unforgeability, standard model, chameleon commitment, chosen message security*

1. Introduction

Strong Existential Unforgeable Signature Scheme: Strong existential unforgeability (SEU) is a stronger variant of the usual security notion, existential unforgeability, for a signature scheme. Ordinary existential unforgeability prohibits an adversary from forging a valid signature on a message which a signer has not signed. However, this does not prohibit an adversary from forging a new valid signature on a message which a signer has already signed. That is, the adversary, by giving a message/signature pair (M, σ), may be able to forge a new valid signature $\sigma' \neq \sigma$ on M. SEU is a security notion that ensures not only existential unforgeability but also that no adversary can execute the type of forgery mentioned above.

SEU is useful in constructing many applications, such as IND-CCA2 secure public-key encryption schemes [6], [10] and a group signature scheme [4]. We review how SEU signatures are used in such applications. In the encryption schemes [6], [10], an SEU signature σ is used as one part of a ciphertext. It is a signature on the other part, C, of the ciphertext. The SEU property ensures the IND-CCA2 security of these schemes. Indeed, if the signature scheme is not SEU, an adversary may be able to obtain a new ciphertext (C, σ') by modifying the signature of another ciphertext (C, σ). This means that the encryption is malleable [10], and hence is not IND-CCA2 secure.

In group signature schemes [4], an authority issues a signature σ on a user's secret key x in advance. The signature will be used as an ID of the user. Hence, if the user succeeds in forging a signature, he also succeeds in forging his ID. Therefore, no signature should be able to be forged, especially, no new signature $\sigma' \neq \sigma$ on the user's secret key x should be. This is the reason we require not only the usual existential unforgeability but SEU property.

Chameleon Commitment Scheme: *Chameleon commitment scheme* [13], [14] (which is also called *randomized trapdoor hashing scheme*) is a variant of commitment scheme such that one can change the committed value after publishing the commitment, if one knows the secret key.

It is known that one can construct such scheme based on the discrete logarithm assumption [13], [14]. The commitment of this scheme is equal to that of Pedersen commitment [16]. That is, the commitment of a message M is $C = g^M h^R$, where (g, h) is a public key and R is a random value. If one knows the discrete logarithm x of h based on g, one can change the committed value of C to an arbitrary message M', by finding R' which satisfies $C = g^{M'} h^{R'}$.

The security notion for the chameleon commitment scheme is similar to that for the ordinary commitment scheme, but its binding property is a bit stronger. That is, it has to satisfy the following *strong binding property*: no one can find two different pairs, (M_1, R_1) and (M_2, R_2), such that the commitment $\mathsf{Com}(M_1, R_1)$ of M_1 generated by using the random string R_1 is the same as the commitment $\mathsf{Com}(M_2, R_2)$ of M_2 generated by using the random string R_2.

Krawczyk and Rabin [13], [14] also showed that their commitment scheme is secure under the discrete logarithm assumption.

1.1 Our Contributions

In this paper, we do the following things:

- We define a new and stronger security notion for

Manuscript received March 22, 2007.
Manuscript revised July 13, 2007.

[†]The author is with the NEC Corporation, Kawasaki-shi, 211-8666 Japan.
[††]The authors are with Tokyo Institute of Technology, Tokyo, 152-8550 Japan.

*The proceedings version of this paper was presented at INDOCRYPT 2006 [24].

a) E-mail: teranisi@ah.jp.nec.com
b) E-mail: taku-zy@crypt.ss.titech.ac.jp
c) E-mail: wakaha@mot.titech.ac.jp

DOI: 10.1093/ietfec/e91–a.1.94

a chameleon commitment scheme and propose new chameleon commitments satisfying this notion.
- By using the above study, we propose two general and efficient conversions, both of which transform a secure signature scheme to a SEU signature scheme.

Results for Chameleon Commitment: The known definition of the strong binding property is weak in the sence that an adversary is not allowed to access any oracle. In this sence, we can say that the known definition is of the key only attack scenario. Therefore, we can consider a stronger security notion, strong binding property *against chosen message attack*, where the adversary can access an oracle knowing a secret key, and make the oracle commit values, and change the commmited values.

In this paper, we give the formal definition of this security notion, and propose two schemes satisfying it. The first scheme is based on the discrete logarithm assumption. The second scheme is constructed from a chameleon commitment scheme satisfying only the security against key only attack.

Chosen message secure chameleon commitment can be useful when it is used as a component of protocols. We will show that it is useful when we construct a SEU signature scheme. It is also useful in the Zhang's work [26]. Zhang proposed a conversion which transforms a tag based encryption to a CCA2 secure public-key encryption, by using some kind of a chameleon commitment. One can show that our chosen message secure chameleon commitments can be used for their purpose.

Results for SEU signature: By using a chameleon commitment scheme, we propose two general and efficient conversions, both of which transform a secure signature scheme into a SEU signature scheme. The conversions ensure the tight security reduction to the underlying security assumptions. That is, if there exists an adversary who succeeds in breaking the SEU property of the converted scheme with probability ε' within t' steps, there exists an adversary who can break at least one of the underlying assumptions with probability $\varepsilon \simeq \varepsilon'$ within $t \simeq t'$ steps.

Moreover, the schemes transformed by our conversions satisfy the on-line/off-line property [21], if we use suitable chameleon commitment schemes. That is, signers can precompute almost all operations on the signing before they are given a message. Therefore, the signer can generate signatures quite efficiently.

There is a trade-off between the securities of these two conversions. The security of a signature scheme transformed by the first conversion can be proved in the standard model, but requires chosen message security for the chameleon commitment scheme. In contrast, the security of a signature scheme transformed by the second conversion can be proved only in the random oracle model, but only requires key only attack security for the chameleon commitment scheme.

This trade-off between securities effects the efficiency of the converted schemes. The second conversion only requires weaker security property for the chameleon commitment scheme and therefore can be implemented by the most efficient chameleon hashing scheme $C = g^M h^R$. It means that the second conversion generates an efficient converted scheme. In contrast, the first conversion has to be implemented by a more complicated chameleon hashing scheme because it requires chosen message security for the chameleon hashing scheme. Therefore, the first conversion only generates somewhat less efficient converted scheme than the second conversion does, although only the first conversion ensures the security to the converted scheme under the standard model.

1.2 Theoretical Interests

From the above results, we can conclude that a chosen message attack secure chameleon commitment scheme and a SEU conversion exist if a claw-free permutation pair exists. This is because we constructed the commitment scheme and the SEU conversion based on a key only attack secure chameleon commitment and because Krawczyk and Rabin [13] constructed a key only attack secure chameleon commitment from a claw-free permutation pair.

1.3 Related Works

Comparing With Trivial Construction of SEU conversion: Rompel [19] showed that oneway function is constructable from an existentially unforgeable signature scheme. Moreover, a SEU secure signature scheme is constructable from oneway-based signature scheme (see 6.5.2. of [12], for instance). Combining of these facts means that we can trivially construct a SEU conversion.

Theoretically, this trivial conversion is superior to our conversion. It requires no additional assumption when we prove the SEU security of the converted scheme, although our conversion requires the existence assumption of a claw-free permutation pair additionally.

However, our conversion is quite more efficient than this trivial conversion. Moreover, our conversion ensures the tight reducibility. Therefore, we can say that our contribution is in proposing a conversion which is efficient and ensures the tight reducibility.

The Result of Boneh, Shen, and Waters [5]: In PKC 2006, Boneh, Shen, and Waters [5] proposed a SEU signature scheme by modifying the Waters signature scheme [25]. They also showed that their modification is applicable to not only the Waters scheme but also any existentially unforgeable signature schemes satisfying the *partitioned* property [5]. However, there are signature schemes, which seems to be non-partitioned, such as DSS, the Camenisch-Lysyanskaya scheme [7], and Okamoto scheme [20]. Moreover, the modified scheme does not satisfy the on-line/off-line property. Our conversions are the first ones that can tightly convert *any* signature scheme, and are also the first

ones that ensure the on-line/off-line property.

The Result of Steinfeld, Pieprzyk, and Wang [23]: Steinfeld, Pieprzyk, and Wang [23] propose a similar conversion to our conversion of the standard model. Their work was done independently and concurrently when we published the extended abstract version [24] of this paper, and the key idea behind the construction of their conversion is the same as that of ours.

We can say that the results of this full paper is a generalization of both our result of the extended abstract [24] and their result [23]. Our proposed conversion implemented with our proposed first or second chameleon commitment scheme is coincident with our conversion of the extended abstract [24] or Steinfeld et al.'s conversion respectively.

The Result of Zhang [26]: Zhang [26] independently defined a weaker variant of our chosen message security of a chameleon commitment, named the *oracle collision resistance* (OCR), and gave an OCR secure scheme under a non-standard assumption, the one-more discrete log assumption [17]. His OCR is equivalent to a variant of our chosen message security where an adversary is not allowed to select committed value and to access the committing oracle twice or more times. We remark this was considered under a different scenario, namely, achieving the CCA2 security of a public-key encryption scheme.

2. Preliminary

2.1 Chameleon Commitment

Chameleon commitment scheme [13], [14] (which is also called *randomized trapdoor hashing scheme*) is a variant of commitment scheme such that one can change the committed value after publishing the commitment, if one knows the secret key:

Definition 2.1 (**Chameleon Commitment Scheme).** Chameleon commitment scheme is a tuple $\Omega = (\mathsf{Gen}, \{\mathcal{R}_{\mathsf{cpk}}\}, \mathsf{Com}, \mathsf{Cham})$ as follows.

- Gen is called a *key generation algorithm*. On inputting 1^κ, Gen outputs a pair of *public key* cpk and a *secret key* csk.
- $\{\mathcal{R}_{\mathsf{cpk}}\}$ is a family of sets $\mathcal{R}_{\mathsf{cpk}}$. Each $\mathcal{R}_{\mathsf{cpk}}$ is a set of random numbers associated with a public key cpk. One can take an element $R \in \mathcal{R}_{\mathsf{cpk}}$ uniformly and randomly in polynomial time.
- Com is called *committing algorithm*. On inputting cpk, a message M and an element $R \in \mathcal{R}_{\mathsf{cpk}}$, Com outputs a *commitment* C of M with respect to the *witness* R. The algorithm Com has to be deterministic.
- Cham is called *chameleon algorithm*. On inputting csk, a message M, a witness R and another message M', Cham outputs a witness R'. For any messages M and M', and for any $R \in \mathcal{R}_{\mathsf{cpk}}$, if we set $R' = \mathsf{Cham}_{\mathsf{csk}}(M, R, M')$, then $\mathsf{Com}_{\mathsf{cpk}}(M, R) = \mathsf{Com}_{\mathsf{cpk}}(M', R')$ holds.

—$\mathsf{Gen}(1^\kappa)$—
$g \leftarrow \mathcal{G}$, $x \leftarrow \mathbb{Z}_q$, $h \leftarrow g^x$,
$\mathsf{cpk} \leftarrow (g, h)$, $\mathsf{csk} \leftarrow x$. Outputs $(\mathsf{cpk}, \mathsf{csk})$.
$\mathcal{R}_{\mathsf{cpk}} = \mathbb{Z}_q$
—$\mathsf{Com}_{\mathsf{cpk}}(M, R)$—
$C \leftarrow g^{H(M)} h^R$. Outputs C.
—$\mathsf{Cham}_{\mathsf{csk}}(M, R, M')$—
Randomly select $R' \in \mathbb{Z}_q$
satisfying $H(M) + Rx = H(M') + R'x \bmod q$.
Outputs R'.

Fig. 1 Chameleon commitment of [13], [14].

Let κ be a security parameter. For non-negative valued functions $t = t(\kappa)$ and $\varepsilon = \varepsilon(\kappa)$, we say that $\Omega = (\mathsf{Gen}, \{\mathcal{R}_{\mathsf{cpk}}\}, \mathsf{Com}, \mathsf{Cham})$ is (t, ε)-*secure* if it satisfies both of the following two security requirements, *uniformity* and (t, ε)-*strong binding property*:

Definition 2.2 (**Uniformity).** We say that Ω satisfies *uniformity* if it satisfies the following property: Let $(\mathsf{cpk}, \mathsf{csk})$ be a public key/secret key pair. Let M and M' be arbitrary messages. For a uniformly randomly selected witness $R \in \mathcal{R}_{\mathsf{cpk}}$, we set $R' = \mathsf{Cham}_{\mathsf{csk}}(M, R, M')$. Then R' distributes uniformly on $\mathcal{R}_{\mathsf{cpk}}$.

Definition 2.3 ((t, ε)**-Strong Binding Property (against the key only attack)).** Let $\mathcal{A}$ be an adversary. We consider the following game: $(\mathsf{cpk}, \mathsf{csk}) \leftarrow \mathsf{Gen}(1^\kappa)$, $((M, R), (\hat{M}, \hat{R})) \leftarrow \mathcal{A}(\mathsf{cpk})$, and output win if and only if both $(M, R) \neq (\hat{M}, \hat{R})$ and $\mathsf{Com}_{\mathsf{cpk}}(M, R) = \mathsf{Com}_{\mathsf{cpk}}(\hat{M}, \hat{R})$ hold. We say that Ω satisfies (t, ε)-*strong binding property* if, for any adversary $\mathcal{A}$ which terminates within t steps, the probability $\mathcal{A}$ wins is less than ε.

We should note the following two points about the above definition. First, we use the term "strong" binding property, because an adversary is required to satisfy not the condition $M \neq \hat{M}$ but the stronger condition $(M, R) \neq (\hat{M}, \hat{R})$. Krawczyk and Rabin [13], [14] also define (t, ε)-*binding property* where an adversary is required to satisfy only $M \neq \hat{M}$, but we do not use this property in this paper.

Second, one can easily show that a "perfect hiding property" of the commitment scheme follows from its uniformity. That is, if Ω satisfies the uniformity, then the distribution of $C_{\mathsf{cpk}}(M, R)$ and $C_{\mathsf{cpk}}(\hat{M}, \hat{R})$ are perfectly indistinguishable, where M and $\hat{M}$ are arbitrary messages and R and $\hat{R}$ are randomly selected witnesses.

Krawczyk and Rabin [13], [14] proposed the chameleon commitment scheme described in Fig. 1, and they show that their scheme is secure under the discrete logarithm assumption.

Definition 2.4 (**Discrete Logarithm Assumption).** Let κ be a security parameter, and $\{\mathcal{G}_\kappa\}$ be a family of cyclic groups $\mathcal{G} = \mathcal{G}_\kappa$ with the prime order $q = q_\kappa$. Let $t = t(\kappa)$ and $\varepsilon = \varepsilon(\kappa)$ be non-negative valued functions.

We say that the (t, ε)-*discrete logarithm assumption* holds in $\{\mathcal{G}_\kappa\}$ if, for any adversary $\mathcal{A}$ which terminates within t steps, $\Pr(h \leftarrow \mathcal{G}, z \leftarrow \mathbb{Z}_q, g \leftarrow h^z, u \leftarrow \mathcal{A}(g, h) : z = u) < \varepsilon$. holds.

Theorem 2.5. *[13], [14] Let $\mathcal{G}$ be a group on which (t, ε)-discrete logarithm assumption holds. Then the chameleon commitment scheme described in Fig. 1 is (t, ε)-secure.*

2.2 Other Notions

Definition 2.6. (**Existential Unforgeability** [11], **Strong Existential Unforgeability (SEU)** [1]) Let κ be a security parameter, $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ be a signature scheme, and $\mathcal{A}$ be an adversary. Let $O^{\mathsf{sig}}_{\mathsf{sk}}$ be an oracle named *signing oracle* which, on inputting a message M, outputs a signature σ on M. We consider the following game:

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa)$,
$(M_0, \sigma_0) \leftarrow \mathcal{A}^{O^{\mathsf{sig}}_{\mathsf{sk}}}(\mathsf{pk})$
If $\mathsf{Ver}_{\mathsf{pk}}(M_0, \sigma_0) = \mathsf{reject}$, return 0
Return 1.

We set (M_i, σ_i) to the pair of the i-th signing query of $\mathcal{A}$ and the corresponding answer. We say that $\mathcal{A}$ *wins* if the output of the above game is 1 and $M_0 \neq M_i$ holds for any i. We also say that $\mathcal{A}$ *wins strongly* if the output of the above game is 1 and $(M_0, \sigma_0) \neq (M_i, \sigma_i)$ holds for any i.

Let $t = t(\kappa)$, $q_S = q_S(\kappa)$, and $\varepsilon = \varepsilon(\kappa)$ be non-negative valued functions. We say that $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ is (t, q_S, ε)-*existentially unforgeable* (resp. (t, q_S, ε)-*strongly existentially unforgeable (SEU)*) if for any adversary $\mathcal{A}$ such that it terminates within t steps and has made at most q_S queries to the signing oracle, the probability that $\mathcal{A}$ will win (resp. strongly win) is less than ε.

If $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ is a signature scheme in the random oracle model [2], we say that Σ is $(t, q_S, q_H, \varepsilon)$-*existentially unforgeable* (resp. $(t, q_S, q_H, \varepsilon)$-*strongly existentially unforgeable (SEU)*) if for any adversary $\mathcal{A}$ such that it terminates within t steps and has made at most q_S queries to the signing oracle and at most q_H queries to the random oracle, the probability that $\mathcal{A}$ will win (resp. strongly win) the above game is less than ε.

Definition 2.7 (**Collision Resistant Hash Function).** Let κ be a security parameter and Let $\{H_\kappa\}$ be a family of functions $H = H_\kappa : \{0,1\}^* \to \{0,1\}^\kappa$ named *hash functions*. Let $t = t(\kappa)$ and $\varepsilon = \varepsilon(\kappa)$ be non-negative valued functions. We say that $\{H_\kappa\}$ is (t, ε)-*collision resistant* if any adversary $\mathcal{A}$, who terminates within t steps, satisfies $\Pr((m_0, m_1) \leftarrow \mathcal{A}(1^\kappa) : H(m_0) = H(m_1)) < \varepsilon$.

3. Chosen Message Security of Chameleon Commitment

The known definition of strong binding property, given in Sect. 2.1, is weak in the sence that an adversary is not allowed to access any oracle. In this sence, we can say that the known definition is of the key only attack scenario. Therefore, we can consider a stronger security notion, strong binding property *against chosen message attack*, where the adversary can access an oracle knowing a secret key, and make the oracle commit values, and change the commmited values.

In this section, we formalize this security notion, and construct two chameleon commitment schemes satisfying it.

—$O(\mathsf{csk}, \cdot)$—
(At the beginning of an experiment,
initialize List to ε and i to 0.).
/* Here i is the number of COM-query.*/
If a query (M, COM) is input,
  $i \leftarrow i + 1$, $R \leftarrow \mathcal{R}_{\mathsf{cpk}}$, $C \leftarrow \mathsf{Com}_{\mathsf{cpk}}(M, R)$,
  List $\leftarrow$ List $\cup \{(i, C, M, R)\}$, outputs C.
If a query (i, M', CHAM) is input,
  If there is some (C, M, R) satisfying $(i, C, M, R) \in$ List,
    $R' \leftarrow \mathsf{Cham}_{\mathsf{csk}}(M, R, M')$.
    List $\leftarrow$ List $\setminus \{(i, C, M, R)\}$, output R'.
  Otherwise
    Output $\perp$.

Fig. 2 The description of oracle O.

3.1 Definition of Chosen Message Security

In the chosen message attack scenario, an adversary is allowed to access an honest commiter. Formally, the honest commiter is realized as an oracle $O(\mathsf{csk}, \cdot)$. The description of the oracle $O(\mathsf{csk}, \cdot)$ is as follows. If an adversary sends a message M_i with a symbol COM, O selects $R_i \in \mathcal{R}_{\mathsf{cpk}}$ uniformly randomly, computes $C_i = \mathsf{Com}_{\mathsf{cpk}}(M_i, R_i)$, stores (C_i, M_i, R_i), and outputs C_i. If an adversary sends a number i, message M'_i, and a symbol CHAM, it computes $R'_i = \mathsf{Cham}_{\mathsf{csk}}(M_i, R_i, M'_i)$ and outputs R'_i. For each i, an adversary is allowed to make query (i, M'_i, CHAM) at most once. Moreover, the adversary is not allowed to make query (i, M'_i, CHAM) such that i is more than the number of COM-query. If the adversary makes such queries, O outputs $\perp$. The formal definition of O is depicted in Fig. 2.

Definition 3.1 (**(t, ε, q)-Strong Binding Property against the Chosen Message Attack).** Let $\Omega = (\mathsf{Gen}, \{R_{\mathsf{cpk}}\}, \mathsf{Com}, \mathsf{Cham})$ be a chameleon commitment scheme. Let $\mathcal{A}$ be an adversary. We consider the following game: $(\mathsf{cpk}, \mathsf{csk}) \leftarrow \mathsf{Gen}(1^\kappa)$, $((M, R), (\hat{M}, \hat{R})) \leftarrow \mathcal{A}^{O(\mathsf{csk}, \cdot)}(\mathsf{cpk})$, and output win if and only if both $(M, R) \neq (\hat{M}, \hat{R})$ and $\mathsf{Com}_{\mathsf{cpk}}(M, R) = \mathsf{Com}_{\mathsf{cpk}}(\hat{M}, \hat{R})$ hold.

We say that the chameleon commitment scheme Ω satisfies (t, ε, q)-*strong binding property against the chosen message attack* if, for any adversary $\mathcal{A}$ which terminates within t steps, and makes COM queries at most q times, (and therefore makes CHAM queries at most q times), the probability $\mathcal{A}$ wins is less than ε.

We say that Ω is (t, ε, q)-*secure against the chosen message attack* if it satisfies the uniformity property and (t, ε, q)-strong binding property against the chosen message attack.

3.2 Idea behind Constructions

We will give two chameleon commitment schemes which are secure against the chosen message attack in Sects. 3.3

—$\mathsf{Gen}(1^\kappa)$— $g \leftarrow \mathcal{G}, x, y \leftarrow \mathbb{Z}_q, (h_1, h_2) \leftarrow (g^x, g^y)$, $\mathsf{cpk} \leftarrow (g, h_1, h_2)$, $\mathsf{csk} \leftarrow (x, y)$. Outputs $(\mathsf{cpk}, \mathsf{csk})$.
$\mathcal{R}_{\mathsf{cpk}} = {\mathbb{Z}_q}^2$
—$\mathsf{Com}_{\mathsf{cpk}}(M, R)$— Parse R as (r, s), $C \leftarrow g^{H(M)} {h_1}^r {h_2}^s$. Outputs C.
—$\mathsf{Cham}_{\mathsf{csk}}(M, R, M')$— Parse R as (r, s). Randomly select $R' = (r', s') \in {\mathbb{Z}_q}^2$ satisfying $H(M) + rx + sy = H(M') + r'x + s'y \bmod q$. Outputs R'.

Fig. 3 First chameleon commitment.

and 3.4. The basic idea behind the constructions of these schemes are the same. That is, we construct chameleon commitment schemes which have two secret keys.

We show why we can ensure the chosen message security of such schemes. In both schemes, the key only attack securities of the schemes are equivalent to the difficulty of finding one of the secret keys. The other secret key is used to simulate the oracle for the chosen message security. That is, a simulator, who is given one of secret key, can run an adversary against the chosen message security, simulate the oracle by using this secret key, obtain an output of the adversary, and find the other unknown secret key.

3.3 The First Proposed Chameleon Commitment

We construct a chosen message secure chameleon commitment scheme based on the discrete logarithm assumption. Let κ be a security parameter, $\mathcal{G}$ be a cyclic group with order q. Our proposed scheme is described in Fig. 3.

Theorem 3.2. *Let E be the exponentiation cost on a group $\mathcal{G}$. Suppose that the (t, ε)-discrete logarithm assumption on $\mathcal{G}$ holds and the (t, ε)-collision resistance of H holds. Then the chameleon commitment scheme $\Omega = (\mathsf{Gen}, \{\mathcal{R}_{\mathsf{cpk}}\}, \mathsf{Com}, \mathsf{Cham})$ described in Fig. 3 is (t', ε', q')-secure against the chosen message attack. Here*

$$t = t' + 2q'E + (\text{lower terms}),$$
$$\varepsilon = \frac{\varepsilon'}{3} - (\text{lower terms}).$$

Proof. Ω clearly satisfies the uniformity property. We show that Ω satisfies the (t', ε', q')-strong binding property against the chosen message attack. Let us make a contradictory supposition that Ω does not satisfy the (t', ε', q')-strong binding property against the chosen message attack. That is, we assume that there exists an adversary $\mathcal{A}$ that can break the strong binding property against the chosen message attack with probability ε' within t' step and within q' signing queries. Then with probability ε', $\mathcal{A}$ finally outputs a collision pair $((M, R), (\hat{M}, \hat{R}))$ satisfying $(M, R) \neq (\hat{M}, \hat{R})$ and $\mathsf{Com}_{\mathsf{cpk}}(M, R) = \mathsf{Com}_{\mathsf{cpk}}(\hat{M}, \hat{R})$ holds. We parse R and $\hat{R}$ as (r, s) and $(\hat{r}, \hat{s})$, respectively. Let $\varepsilon_1, \varepsilon_2$, and ε_3 be the probability that $\mathcal{A}$ outputs a collision pair and the following (1), (2), and (3) hold, respectively.

(1) $M \neq \hat{M}$ and $(r, s) = (\hat{r}, \hat{s})$,
(2) $r \neq \hat{r}$,
(3) $s \neq \hat{s}$

By using $\mathcal{A}$ as a subroutine, we will construct three machines $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$, and will show the following facts:

- If $\varepsilon_1 \geq \varepsilon'/3$, $\mathcal{B}_1$ succeeds in breaking (t, ε)-collision resistance of H.
- If $\varepsilon_2 \geq \varepsilon'/3$ or $\varepsilon_3 \geq \varepsilon'/3$, $\mathcal{B}_2$ or $\mathcal{B}_3$ succeeds in breaking (t, ε)-discrete logarithm assumption, respectively.

Clearly $\varepsilon_1 + \varepsilon_2 + \varepsilon_3 \geq \varepsilon'$ holds. Therefore, at least one of ε_i is not less than $\varepsilon'/3$. This means that the theorem holds.

The case where $\varepsilon_1 \geq \varepsilon'/3$ holds: By using $\mathcal{A}$ as a subroutine, we construct an adversary $\mathcal{B}_1$ which can break the (t, ε_1)-collision resistance of H. The adversary $\mathcal{B}_1$ runs $\mathcal{A}$ as follows:

Setup: $\mathcal{B}_1$ executes $\mathsf{Gen}(1^\kappa)$ and obtains a public key $\mathsf{cpk} = (g, h_1, h_2)$ and $\mathsf{csk} = (x, y)$ as outputs. Then $\mathcal{B}_1$ initializes List to the empty list, and executes $\mathcal{A}(\mathsf{cpk})$.

Oracle Simulation: Since $\mathcal{B}_1$ knows the secret key csk, $\mathcal{B}_1$ can simulate $O(\mathsf{csk}, \cdot)$-oracle.

Extraction: $\mathcal{A}$ finally outputs a pair $((M, R), (\hat{M}, \hat{R}))$. With the probability at least ε', the properties $(M, R) \neq (\hat{M}, \hat{R})$ and $\mathsf{Com}_{\mathsf{cpk}}(M, R) = \mathsf{Com}_{\mathsf{cpk}}(\hat{M}, \hat{R})$ hold.

We consider the case where these properties hold. We parse R and $\hat{R}$ as (r, s) and $(\hat{r}, \hat{s})$, respectively. Since $\mathcal{A}$ is type 1 adversary, $(r, s) = (\hat{r}, \hat{s})$ holds. Since the equality $\mathsf{Com}_{\mathsf{cpk}}(M, R) = \mathsf{Com}_{\mathsf{cpk}}(\hat{M}, \hat{R})$ holds, it follows that $g^{H(M)} {h_1}^r {h_2}^s = g^{H(\hat{M})} {h_1}^{\hat{r}} {h_2}^{\hat{s}} = g^{H(\hat{M})} {h_1}^r {h_2}^s$. Hence, $H(M) = H(\hat{M})$ holds. Since $M \neq \hat{M}$ holds, this means that $(M, \hat{M})$ is a collision pair of H. That is, $\mathcal{B}_1$ succeeds in obtaining the collision pair $(M, \hat{M})$.

One can easily show that $\mathcal{B}_1$ terminates within t steps and has the success probability ε_1. If $\varepsilon_1 \geq \varepsilon'/3$, $\mathcal{B}_1$ breaks the (t, ε)-collision resistance of H.

The case where $\varepsilon_2 \geq \varepsilon'/3$ holds: By using $\mathcal{A}$ as a subroutine, we construct an adversary $\mathcal{B}_2$ that can solve the discrete logarithm problem. Let $(g_*, h_*) \in \mathcal{G}^2$ be an instance of the discrete logarithm problem in $\mathcal{G}$. The aim of $\mathcal{B}_2$ is to obtain $z_* \in \mathbb{Z}_q$ satisfying $h_* = {g_*}^{z_*}$. The adversary $\mathcal{B}_2$ runs $\mathcal{A}$ as follows:

Setup: $\mathcal{B}_2$ selects $y \in \mathbb{Z}_q$ randomly and sets $\mathsf{cpk} = (g, h_1, h_2)$ to $(g_*, h_*, {g_*}^y)$. $\mathcal{B}_2$ initializes List to the empty list. Then $\mathcal{B}_2$ provides cpk to $\mathcal{A}$ and runs $\mathcal{A}$.

Oracle Simulation: Let i be a positive integer. If $\mathcal{A}$ makes i-th COM-query (M_i, COM), $\mathcal{B}$ selects $r'_i, s_i \in \mathbb{Z}_q$ randomly, computes $C_i = g^{H(M_i)} {h_1}^{r'_i} {h_2}^{s_i}$ $(i, C_i, M_i, (r'_i, s_i))$ to List, and sends C_i back to $\mathcal{A}$. If $\mathcal{A}$ makes CHAM-query (i, M'_i, CHAM), $\mathcal{B}_2$ finds $(i, C_i, M_i, (r'_i, s_i)) \in \mathsf{List}$. If there is no such $(i, C_i, M_i, (r'_i, s_i))$, $\mathcal{B}_2$ sends $\perp$ back to $\mathcal{A}$. Otherwise, $\mathcal{B}_2$ selects $s'_i \in \mathbb{Z}_q$ satisfying $H(M_i) + s_i y = H(M'_i) + s'_i y \bmod q$, remove $(i, C_i, M_i, (r'_i, s_i))$ from List, and outputs $R'_i = (r'_i, s'_i)$.

Extraction: Suppose that $\mathcal{A}$ succeeds in outputting

—$\mathsf{Gen}(1^\kappa)$— $(\mathsf{cpk}_1, \mathsf{csk}_1) \leftarrow \mathsf{Gen}^{(1)}(1^\kappa)$, $(\mathsf{cpk}_2, \mathsf{csk}_2) \leftarrow \mathsf{Gen}^{(2)}(1^\kappa)$, $\mathsf{cpk} \leftarrow (\mathsf{cpk}_1, \mathsf{cpk}_2)$, $\mathsf{csk} \leftarrow (\mathsf{csk}_1, \mathsf{csk}_2)$. Outputs $(\mathsf{cpk}, \mathsf{csk})$.
$\mathcal{R}_{\mathsf{cpk}} = \mathcal{R}^{(1)}_{\mathsf{cpk}_1} \times \mathcal{R}^{(2)}_{\mathsf{cpk}_2}$
—$\mathsf{Com}_{\mathsf{cpk}}(M, R)$— Parse R as (R_1, R_2). $C_1 \leftarrow \mathsf{Com}^{(1)}_{\mathsf{cpk}_1}(M, R_1)$, $C_2 \leftarrow \mathsf{Com}^{(2)}_{\mathsf{cpk}_2}(C_1, R_2)$. Outputs C_2.
—$\mathsf{Cham}_{\mathsf{csk}}(M, R, M')$— Parse R as (R_1, R_2). $R'_1 \leftarrow \mathsf{Cham}^{(1)}_{\mathsf{csk}_1}(M, R_1, M')$, $R'_2 \leftarrow R_2$. Outputs $R' = (R'_1, R'_2)$.

Fig. 4 Second chameleon commitment.

$((M, R), (\hat{M}, \hat{R}))$ satisfying both $(M, R) \neq (\hat{M}, \hat{R})$ and $\mathsf{Com}_{\mathsf{cpk}}(M, R) = \mathsf{Com}_{\mathsf{cpk}}(\hat{M}, \hat{R})$. We parse R and $\hat{R}$ as (r, s) and $(\hat{r}, \hat{s})$ respectively. Then, from the definition of Com, $\mathsf{Com}_{\mathsf{cpk}}(M, R) = \mathsf{Com}_{\mathsf{cpk}}(\hat{M}, \hat{R})$ means that $g^{H(M)} {h_1}^r {h_2}^s = g^{H(\hat{M})} {h_1}^{\hat{r}} {h_2}^{\hat{s}}$ holds. Therefore, it follows ${g_*}^{H(M)+ys} {h_*}^r = g^{H(M)} {h_1}^r {h_2}^s = g^{H(\hat{M})} {h_1}^{\hat{r}} {h_2}^{\hat{s}} = {g_*}^{H(\hat{M})+y\hat{s}} {h_*}^{\hat{r}}$. Recall that $r \neq \hat{r}$ holds. Therefore, $z_* = (H(M)+sy-H(\hat{M})-\hat{s}y)/(\hat{r}-r) \bmod q$ is the discrete logarithm of g_* based on h_*. Therefore, $\mathcal{B}_2$ outputs z_* and stops.

One can easily show that $\mathcal{B}_2$ terminates within t steps and has the success probability ε_2. If $\varepsilon_2 \geq \varepsilon'/3$, $\mathcal{B}_2$ can break the (t, ε)-discrete logarithm problem in $\mathcal{G}$.

The case where $\varepsilon_3 \geq \varepsilon'/3$ holds: The construction of $\mathcal{B}_3$ is quite similar to the construction of $\mathcal{B}_2$, although $\mathcal{B}_3$ embeds (g_*, h_*) not to (g, h_1) but to (g, h_2). Therefore, we omit the details. □

3.4 The Second Proposed Chameleon Commitment

We next construct a chosen message secure chameleon commitment scheme by using key only attack secure chameleon commitment schemes. Let κ be a security parameter, $\Omega^{(1)} = (\mathsf{Gen}^{(1)}, \{R^{(1)}_{\mathsf{cpk}}\}, \mathsf{Com}^{(1)}, \mathsf{Cham}^{(1)})$ and $\Omega^{(2)} = (\mathsf{Gen}^{(2)}, \{\mathcal{R}^{(2)}_{\mathsf{cpk}}\}, \mathsf{Com}^{(2)}, \mathsf{Cham}^{(2)})$ be two chameleon commitment schemes. Our proposed scheme is described in Fig. 4.

Theorem 3.3. *Let U be the maximum of the computational cost of $\mathsf{Com}^{(1)}$ and that of $\mathsf{Com}^{(2)}$. Let T be the maximum of the computational cost of $\mathsf{Cham}^{(1)}$ and that of $\mathsf{Cham}^{(2)}$. Suppose that $\Omega^{(1)}$ and $\Omega^{(2)}$ satisfy (t, ε)-security against the key only attack. Then the chameleon commitment scheme $\Omega = (\mathsf{Gen}, \{\mathcal{R}_{\mathsf{cpk}}\}, \mathsf{Com}, \mathsf{Cham})$ described in Fig. 4 satisfies (t', ε', q')-security against the chosen message attack. Here*

$$t = t' + 2q(U + T) + (\text{lower terms})$$
$$\varepsilon = \frac{\varepsilon'}{2} - (\text{lower terms})$$

Since we can construct a key only attack secure chameleon commitment from a claw-free permutation pair [13], we can obtain the following corollary:

Corollary 3.4. *If there exists a claw-free permutation pair, then there exists a chameleon commitment scheme which is secure against chosen message attack.* □

Proof of Theorem 3.3. One can easily show that Ω satisfies the uniformity property. Therefore, we only show that Ω satisfies the (t', ε')-strong binding property against the chosen message attack. Suppose that there exists an adversary $\mathcal{A}$ that can break the (t', ε')-strong binding property against the chosen message attack. Then with probability ε', $\mathcal{A}$ finally outputs $((M, R), (\hat{M}, \hat{R}))$ satisfying $(M, R) \neq (\hat{M}, \hat{R})$ and $\mathsf{Com}_{\mathsf{cpk}}(M, R) = \mathsf{Com}_{\mathsf{cpk}}(\hat{M}, \hat{R})$ holds. We parse R and $\hat{R}$ as (R_1, R_2) and $(\hat{R}_1, \hat{R}_2)$ respectively, and set $C_1 = \mathsf{Com}^{(1)}_{\mathsf{cpk}_1}(M, R_1)$ and $\hat{C}_1 = \mathsf{Com}^{(1)}_{\mathsf{cpk}_1}(\hat{M}, \hat{R}_1)$.

One can easily show that $(M, R) \neq (\hat{M}, \hat{R})$ holds if and only if at least one of the following two properties holds:

(1) $(M, R_1) \neq (\hat{M}, \hat{R}_1)$ and $C_1 = \hat{C}_1$
(2) Either $R_2 \neq \hat{R}_2$ or $C_1 \neq \hat{C}_1$ holds.

Let ε_1 (resp. ε_2) be the probability that $\mathcal{A}$ succeeds in computing a collision pair with property (1) (resp. (2)).

By using $\mathcal{A}$ as a subroutine, we will construct two machines $\mathcal{B}_1$ and $\mathcal{B}_2$, and will show the following facts:

- If $\varepsilon_1 \geq \varepsilon'/2$, $\mathcal{B}_1$ succeeds in breaking (t, ε)-strong binding property of $\Omega^{(1)}$ against the key only attack.
- If $\varepsilon_2 \geq \varepsilon'/2$, $\mathcal{B}_2$ succeeds in breaking (t, ε)-strong binding property of $\Omega^{(2)}$ against the key only attack.

Clearly, at least one of $i \in \{1, 2\}$ satisfies $\varepsilon_i \geq \varepsilon'/2$. This means that the theorem holds.

The case where $\varepsilon_1 \geq \varepsilon'/2$ holds: By using $\mathcal{A}$ as a subroutine, we construct an adversary $\mathcal{B}_1$ which can break the (t, ε_1)-strong binding property of $\Omega^{(1)}$ against the key only attack. Let cpk^*_1 be an input of $\mathcal{B}_1$. $\mathcal{B}_1$ runs $\mathcal{A}$ as follows:

Setup: $\mathcal{B}_1$ executes $\mathsf{Gen}^{(2)}(1^\kappa)$ and obtains $(\mathsf{cpk}_2, \mathsf{csk}_2)$ as an output. Then, $\mathcal{B}_1$ initializes List to the empty list, sets $\mathsf{cpk}_1 = \mathsf{cpk}^*_1$ and $\mathsf{cpk} = (\mathsf{cpk}_1, \mathsf{cpk}_2)$ and executes $\mathcal{A}(\mathsf{cpk})$.

Oracle Simulation: Let i be a positive integer. If $\mathcal{A}$ makes i-th COM-query (M_i, COM), $\mathcal{B}_1$ selects $R_{1,i}$ and $R_{2,i}$ randomly, computes $C_{1,i} = \mathsf{Com}^{(1)}_{\mathsf{cpk}_1}(M_i, R_{1,i})$ and $C_{2,i} = \mathsf{Com}^{(2)}_{\mathsf{cpk}_2}(C_{1,i}, R_{2,i})$, adds $(i, C_{2,i}, M_i, (R_{1,i}, R_{2,i}))$, and sends $C_{2,i}$ back to $\mathcal{A}$.

If $\mathcal{A}$ makes CHAM-query (i, M'_i, CHAM), $\mathcal{B}_1$ finds $(i, C_{2,i}, M_i, (R_{1,i}, R_{2,i})) \in \mathsf{List}$. If there is no such $(i, C_{2,i}, M_i, (R_{1,i}, R_{2,i}))$, $\mathcal{B}_1$ sends $\perp$ back to $\mathcal{A}$. Otherwise, $\mathcal{B}_1$ selects $R'_{1,i}$, computes $C'_{1,i} = \mathsf{Com}^{(1)}_{\mathsf{cpk}_1}(M'_i, R'_{1,i})$ and $R'_{2,i} = \mathsf{Cham}^{(2)}_{\mathsf{csk}_2}(C_{1,i}, R_{2,i}, C'_{1,i}, \mathsf{CHAM})$, removes $(i, C_{2,i}, M_i, (R_{1,i}, R_{2,i}))$ from List, and outputs $R'_i = (R'_{1,i}, R'_{2,i})$.

One can easily show that $C_{2,i} = \mathsf{Com}_{\mathsf{cpk}}(M'_i, R'_i)$ holds. By using the uniformity of $\Omega^{(1)}$ and $\Omega^{(2)}$, one can also easily show that the distribution of the output of $\mathcal{B}$ is the same as that of the output of the oracle O.

Extraction: $\mathcal{A}$ finally outputs a pair $((M, R), (\hat{M}, \hat{R}))$. We parse R and $\hat{R}$ as (R_1, R_2) and $(\hat{R}_1, \hat{R}_2)$ respectively. Let $C_1 = \mathsf{Com}^{(1)}_{\mathsf{cpk}_1}(M, R_1)$ and $\hat{C}_1 = \mathsf{Com}^{(1)}_{\mathsf{cpk}_1}(\hat{M}, \hat{R}_1)$. If $(M, R_1) \neq (\hat{M}, \hat{R}_1)$ and $C_1 = \hat{C}_1$ hold, $\mathcal{B}_1$ outputs $((M, R_1), (M', R'_1))$.

One can easily show that $\mathcal{B}_1$ terminates within t steps and has the success probability ε_2. If $\varepsilon_2 \geq \varepsilon'/2$, $\mathcal{B}_1$ breaks (t, ε)-strong binding property of $\Omega^{(1)}$ against the key only attack.

The case where $\varepsilon_2 \geq \varepsilon'/2$ holds: By using $\mathcal{A}$ as a subroutine, we construct an adversary $\mathcal{B}_2$ that breaks the strong binding property of $\Omega^{(2)}$. Let cpk_2^* be an input of $\mathcal{B}_2$. The adversary $\mathcal{B}_2$ runs $\mathcal{A}$ as follows:

Setup: $\mathcal{B}_2$ executes $\mathsf{Gen}^{(1)}(1^\kappa)$ and obtains $(\mathsf{cpk}_1, \mathsf{csk}_1)$ as an output of it. Then $\mathcal{B}_2$ initializes List to the empty list, sets $\mathsf{cpk}_2 = \mathsf{cpk}_2^*$ and $\mathsf{cpk} = (\mathsf{cpk}_1, \mathsf{cpk}_2)$ and executes $\mathcal{A}(\mathsf{cpk})$.

Oracle Simulation: If $\mathcal{A}$ makes i-th COM-query (M_i, COM), $\mathcal{B}$ selects $R_{1,i}$ and $R'_{2,i}$ randomly, computes $C_{1,i} = \mathsf{Com}^{(1)}_{\mathsf{cpk}_1}(M_i, R_{1,i})$ and $C_{2,i} = \mathsf{Com}^{(2)}_{\mathsf{cpk}_2}(C_{1,i}, R'_{2,i})$, add $(i, C_{2,i}, M_i, (R_{1,i}, R'_{2,i}))$ to List, and sends $C_{2,i}$ back to $\mathcal{A}$.

If $\mathcal{A}$ makes CHAM-query $(C_{2,i}, M'_i, \mathsf{CHAM})$, $\mathcal{B}_2$ finds $(i, C_{2,i}, M_i, (R_{1,i}, R'_{2,i})) \in \mathsf{List}$. If there is no such $(i, C_{2,i}, M_i, (R_{1,i}, R'_{2,i}))$, $\mathcal{B}_2$ sends $\perp$ back to $\mathcal{A}$. Otherwise, $\mathcal{B}_2$ computes $R'_{1,i} = \mathsf{Cham}^{(1)}_{\mathsf{csk}_1}(M_i, R_{1,i}, M'_i)$, removes $(i, C_{2,i}, M_i, (R_{1,i}, R'_{2,i}))$ from List, and outputs $R'_i = (R'_{1,i}, R'_{2,i})$.

One can easily show that $C_{2,i} = \mathsf{Com}_{\mathsf{cpk}}(M'_i, R'_i)$ holds. By using the uniformity of $\Omega^{(1)}$ and $\Omega^{(2)}$, one can also easily show that the distribution of the output of $\mathcal{B}$ is the same as that of the output of the oracle O.

Extraction: $\mathcal{A}$ finally outputs a pair $((M, R), (\hat{M}, \hat{R}))$. We parse R and $\hat{R}$ as (R_1, R_2) and $(\hat{R}_1, \hat{R}_2)$ respectively. $\mathcal{B}_2$ sets $C_1 = \mathsf{Com}^{(1)}_{\mathsf{cpk}_1}(M, R_1)$ and $\hat{C}_1 = \mathsf{Com}^{(1)}_{\mathsf{cpk}_1}(\hat{M}, \hat{R}_1)$. Then $\mathcal{B}_2$ outputs $((C_1, R_2), (\hat{C}_1, \hat{R}_2))$.

With probability ε_2, $\mathcal{A}$ outputs a collision pair $((M, R), (\hat{M}, \hat{R}))$ satisfying $(M, R) \neq (\hat{M}, \hat{R})$, $\mathsf{Com}_{\mathsf{cpk}}(M, R) = \mathsf{Com}_{\mathsf{cpk}}(\hat{M}, \hat{R})$, and $(C_1, R_2) \neq (\hat{C}_1, \hat{R}_2)$. In this case, $\mathsf{Com}^{(2)}_{\mathsf{cpk}_2}(C_1, R_2) = \mathsf{Com}_{\mathsf{cpk}}(M, R) = \mathsf{Com}_{\mathsf{cpk}}(\hat{M}, \hat{R}) = \mathsf{Com}^{(2)}_{\mathsf{cpk}_2}(\hat{C}_1, \hat{R}_2)$. This means that $\mathcal{B}_2$ succeeds in computing a collision of $\mathsf{Com}^{(2)}$ with probability ε_2. One can easily show that $\mathcal{B}_2$ terminates within t steps. Therefore, if $\varepsilon_2 \geq \varepsilon'/2$, $\mathcal{B}_2$ breaks (t, ε)-strong binding property of $\Omega^{(2)}$. □

4. Proposed Conversion of Signature Schemes in the Standard Model

In this section, we construct a general and efficient conversion in the standard model, such that the conversion transforms a secure signature scheme to an SEU signature scheme.

4.1 Idea behind Construction

The most basic idea behind our proposed conversion is the same as that of the previous conversion in [5]. Therefore, we first review the idea in [5]. A signature on the converted schemes is a pair (σ, R') satisfying the tricky property $\sigma = \mathsf{Sig}_{\mathsf{sk}}(C(\sigma \| M; R'))$, where M is a message, σ is a signature on the original scheme and $C(\sigma \| M; R')$ is a chameleon hash value of $\sigma \| M$ generated by using the random R'. A signer can compute such signature σ as follows: compute commitment $C(0; R)$ of a message 0, and a signature $\sigma = \mathsf{Sig}_{\mathsf{sk}}(C(0; R))$, and change the committed value of C from 0 to $\sigma \| M$ by using secret key.

Since $\sigma = \mathsf{Sig}_{\mathsf{sk}}(C(\sigma \| M; R'))$ holds, we can recognize σ as "the signature on (the commitment of) the signature itself." Therefore, in order to forge a new signature $(\hat{\sigma}, \hat{R})$ of the converted scheme on a message M, an adversary has to forge a signature (that is, $\hat{\sigma} = \mathsf{Sig}_{\mathsf{sk}}(C(\hat{\sigma} \| M; \hat{R}))$) of the original scheme on a *new* message $C(\sigma' \| M; R')$. However, this is impossible because the original scheme is existentially unforgeable. Therefore, the converted scheme is SEU secure.

However, the idea mentioned above does not work generally. Recall that, when we prove the security of the converted scheme, we have to construct a simulator which can simulate a signing oracle without using the secret key. Moreover, recall that one can change the committed value only if he knows the secret key. Hence the simulator cannot change the committed value and therefore cannot simulate the signing oracle. (Therefore, the authors of the previous paper [5] only consider a signature scheme satisfying some special property.)

In order to enable a simulator to simulate the signing oracle, we assume that the chameleon commitment scheme is secure against the chosen message attack. That is, we assume that the chameleon commitment scheme is secure even if one can access an oracle which knows a secret key, commits a value, and changes the commmited value. By making this oracle change the commmited value, the simulator can simulate the signing oracle.

—$\mathsf{Gen}'(1^\kappa)$—
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa)$, $(\mathsf{cpk}, \mathsf{csk}) \leftarrow \mathsf{CGen}(1^\kappa)$.
$\mathsf{pk}' \leftarrow (\mathsf{pk}, \mathsf{cpk})$, $\mathsf{sk}' \leftarrow (\mathsf{sk}, \mathsf{csk})$. Output $(\mathsf{pk}', \mathsf{sk}')$.
—$\mathsf{Sig}'_{\mathsf{sk}'}(M)$—
Select $R \in \mathcal{R}_{\mathsf{cpk}}$ randomly.
$C \leftarrow \mathsf{Com}_{\mathsf{cpk}}(0, R)$, $\sigma \leftarrow \mathsf{Sig}_{\mathsf{sk}}(C)$,
$R' \leftarrow \mathsf{Cham}_{\mathsf{csk}}(0, R, M \| \sigma)$.
$\sigma' \leftarrow (\sigma, R')$. Output σ'.
—$\mathsf{Ver}'_{\mathsf{pk}'}(M, \sigma')$—
Parse σ' as (σ, R'). $C \leftarrow \mathsf{Com}_{\mathsf{cpk}}(M \| \sigma, R')$.
If $\mathsf{Ver}_{\mathsf{pk}}(C, \sigma) = \mathsf{accept}$ then return accept.
Otherwise return reject.

Fig. 5 Proposed conversion in the standard model.

4.2 Proposed Scheme

Let κ be a security parameter, $\mathcal{G}$ be a cyclic group with order q, and $\Omega = (\mathsf{CGen}, \{\mathcal{R}_{\mathsf{cpk}}\}, \mathsf{Com}, \mathsf{Cham})$ be a chameleon commitment scheme. Let $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ be a signature scheme. Our conversion transforms the scheme Σ into the signature scheme $\Sigma' = (\mathsf{Gen}', \mathsf{Sig}', \mathsf{Ver}')$ described in Fig. 5.

We next study the on-line/off-line property of the converted scheme Σ'. Before receiving a message, the signer

can precompute the following parts of signing computations: $C = \mathsf{Com}_{\mathsf{cpk}}(0, R)$ and $\sigma = \mathsf{Sig}_{\mathsf{sk}}(C)$. Moreover, the computation of the last part, $R' = \mathsf{Cham}_{\mathsf{csk}}(0, R, M\|\sigma)$, is quite efficient, if we set Ω to the first proposed chameleon commitment scheme of Sect. 3. This means that our converted scheme Σ' satisfies on-line/off-line property if we set Ω to such a chameleon commitment scheme.

Theorem 4.1. *Let S be the signing cost of Σ. Let U and T be the computational cost of* Com *and* Cham.

Suppose that there exists an adversary that can break the (t', q_S, ε')-SEU property of the signature scheme $\Sigma' = (\mathsf{Gen}', \mathsf{Sig}', \mathsf{Ver}')$. Then, there exists an adversary that can break either the (t, q_S, ε)-existential unforgeability of the underlying signature scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$, or the (t, ε, q_S)-strong binding property of Ω against the chosen message attack. Here

$$\begin{cases} t = t' + (S + U + T)q_S + (\text{lower terms}), \\ \varepsilon = \varepsilon'/2 - (\text{lower terms}). \end{cases}$$

Proof. Let $\mathcal{A}$ be an adversary that breaks the (t', q_S, ε')-SEU property of Σ'. The adversary $\mathcal{A}$ is first given a public key $\mathsf{pk}' = (\mathsf{pk}, \mathsf{cpk})$. $\mathcal{A}$ makes queries $M_1, \ldots, M_{q_S}$ to the signing oracle $O^{\mathsf{Sig}'}_{\mathsf{sk}'}$ adaptively, and receives the signatures $\sigma'_1 = (\sigma_1, R'_1), \ldots, \sigma'_{q_S} = (\sigma_{q_S}, R'_{q_S})$ on these messages. $\mathcal{A}$ finally outputs a message M and a signature $\sigma' = (\sigma, R')$. We let C_i and C be $\mathsf{Com}_{\mathsf{cpk}}(M_i, R'_i)$ and $\mathsf{Com}_{\mathsf{cpk}}(M, R')$ respectively.

Let ε_1 and ε_2 be the probability that $\mathcal{A}$'s output is a valid message-signature pair with the following property (1) and (2), respectively.

(1) $C \neq C_i$ for any i
(2) $C = C_i$ for some i

By using $\mathcal{A}$ as a subroutine, we will construct two machines $\mathcal{B}_1$ and $\mathcal{B}_2$, and will show the following facts:

- If $\varepsilon_1 \geq \varepsilon'/2$, then $\mathcal{B}_1$ succeeds in breaking the (t, q_S, ε)-existential unforgeability of Σ.
- If $\varepsilon_2 \geq \varepsilon'/2$, then $\mathcal{B}_2$ succeeds in breaking the (t, ε, q_S)-strong binding property of Ω against the chosen message attack.

This means that the theorem holds.

The case where $\varepsilon_1 \geq \varepsilon'/2$ holds: By using $\mathcal{A}$ as a subroutine, we construct an adversary $\mathcal{B}_1$ that can break the (t, q_S, ε_1)-existential unforgeability of Σ. Let pk_* be the input of $\mathcal{B}_1$. The adversary $\mathcal{B}_1$ runs $\mathcal{A}$ as follows:

Setup: $\mathcal{B}_1$ computes $(\mathsf{cpk}, \mathsf{csk}) = \mathsf{CGen}(1^\kappa)$ and sets $\mathsf{pk} = \mathsf{pk}_*$ and $\mathsf{pk}' = (\mathsf{pk}, \mathsf{cpk})$. Then $\mathcal{B}_1$ provides pk' to $\mathcal{A}$.

Signing Oracle Simulation: Let M_i be the i-th queried message of $\mathcal{A}$. $\mathcal{B}_1$ executes the same algorithm as $\mathsf{Sig}'_{\mathsf{sk}'}(M_i)$ except that $\mathcal{B}_1$ does not execute $\mathsf{Sig}_{\mathsf{sk}}$ but makes a query to the signing oracle. More precisely, $\mathcal{B}_1$ executes the following procedures. $\mathcal{B}_1$ selects $R \in \mathcal{R}_{\mathsf{cpk}}$ randomly, computes $C = \mathsf{Com}_{\mathsf{cpk}}(0, R)$, makes query C to the signing oracle, receives $\sigma = \mathsf{Sig}_{\mathsf{sk}}(C)$ from the oracle as an answer, computes $R' = \mathsf{Cham}_{\mathsf{csk}}(0, R, M\|\sigma)$, sets $\sigma' = (\sigma, R')$, and sends σ' back to $\mathcal{A}$.

Extraction: Eventually, $\mathcal{A}$ outputs a message M and a signature $\sigma' = (\sigma, R')$. $\mathcal{B}_1$ outputs (C, σ), where $C = \mathsf{Com}_{\mathsf{cpk}}(M, R')$.

Clearly, $\mathcal{B}_1$ stops within t steps. We next estimate the probability that $\mathcal{B}_1$ succeeds in forging a new valid message-signature pair. With probability ε_1, (a) $\sigma' = (\sigma, R')$ is a valid signature on M, (b) $(M, \sigma') \neq (M_i, \sigma'_i)$ holds for any i, and (c) $C \neq C_i$ holds for any i. If (a) holds, σ is a valid signature on C. If (c) holds, $\mathcal{B}_1$ has never queried C to the signing oracle. (Recall that the queries that $\mathcal{B}_1$ has made to the signing oracle are $C_1, \ldots, C_{q_S}$.) Consequently, the probability that $\mathcal{B}_1$ succeeds in breaking existential unforgeability is at least ε_1.

The case where $\varepsilon_2 \geq \varepsilon'/2$ holds: By using $\mathcal{A}$ as a subroutine, we construct a machine $\mathcal{B}_2$ that can break the (t, ε, q_S)-strong binding property of Ω against the chosen message attack. Let cpk_* be an input of $\mathcal{B}_2$. $\mathcal{B}_2$ runs $\mathcal{A}$ as follows:

Setup: $\mathcal{B}_2$ computes $(\mathsf{pk}, \mathsf{sk}) = \mathsf{Gen}(1^\kappa)$ and sets $\mathsf{cpk} = \mathsf{cpk}_*$ and $\mathsf{pk}' = (\mathsf{pk}, \mathsf{cpk})$. Then $\mathcal{B}_2$ provides pk' to $\mathcal{A}$.

Signing Oracle Simulation: Let M_i be the i-th queried message of $\mathcal{A}$. Recall that $\mathcal{B}_2$ is allowed to access the oracle O. Recall also that $\mathcal{B}_2$ knows the secret key sk. $\mathcal{B}_2$ makes COM-query $(0, \mathsf{COM})$ to O, receives a commitment C_i of the message 0 as an answer from it, computes $\sigma_i = \mathsf{Sig}_{\mathsf{sk}}(C_i)$ by using the secret key sk, makes CHAM-query $(i, M_i\|\sigma_i, \mathsf{CHAM})$ to O, and receives R'_i as an answer from it, sets $\sigma'_i = (\sigma_i, R'_i)$, and sends σ'_i back to $\mathcal{A}$.

Extraction: Suppose that $\mathcal{A}$ outputs a message M and a valid signature $\sigma' = (\sigma, R')$ on M such that $(M, \sigma') \neq (M_j, \sigma'_j)$ for any j. We also suppose that $C = C_i$ holds for some i. From the definition of C and C_i, $\mathsf{Com}_{\mathsf{cpk}}(M\|\sigma, R') = C = C_i = \mathsf{Com}_{\mathsf{cpk}}(M_i\|\sigma_i, R'_i)$ holds.

Since $(M, \sigma') \neq (M_i, \sigma'_i)$, $\sigma' = (\sigma, R')$, and $\sigma'_i = (\sigma_i, R'_i)$ hold, $(M\|\sigma, R') \neq (M_i\|\sigma_i, R'_i)$ holds. This means that $((M\|\sigma, R'), (M_i\|\sigma_i, R'_i))$ is a collision pair of $\mathsf{Com}_{\mathsf{cpk}}$. Therefore, $\mathcal{B}_2$ outputs $((M\|\sigma, R'), (M_i\|\sigma_i, R'_i))$, as an answer to the strong binding property game.

One can easily show that $\mathcal{B}_2$ succeeds in breaking (t, ε_2, q_S)-strong collision resistance of Ω against the chosen message attack. □

5. Proposed Conversion in the Random Oracle Model

In this section, we construct a general and efficient conversion in the random oracle model such that the conversion transforms a secure signature scheme to an SEU signature scheme. Let κ be a security parameter and $\Omega = (\mathsf{CGen}, \{\mathcal{R}_{\mathsf{cpk}}\}, \mathsf{Com}, \mathsf{Cham})$ be a chameleon commitment scheme. Let $\mathcal{H} : \{0,1\}^* \to \{0,1\}^p$ be a hash function, which we will replace with the random oracle when we prove the security of the converted scheme. Let $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ be a signature scheme. Our conversion transforms the scheme Σ to the signature scheme $\Sigma' = (\mathsf{Gen}', \mathsf{Sig}', \mathsf{Ver}')$

—$\mathsf{Gen}'(1^\kappa)$—
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa)$, $(\mathsf{cpk},\mathsf{csk}) \leftarrow \mathsf{CGen}(1^\kappa)$.
$\mathsf{pk}' \leftarrow (\mathsf{pk},\mathsf{cpk})$, $\mathsf{sk}' \leftarrow (\mathsf{sk},\mathsf{csk})$.
Output $(\mathsf{pk}',\mathsf{sk}')$.
—$\mathsf{Sig}'_{\mathsf{sk}'}(M)$—
Select R randomly.
$C \leftarrow \mathsf{Com}_{\mathsf{cpk}}(0,R)$, $\sigma \leftarrow \mathsf{Sig}_{\mathsf{sk}}(C)$,
$R' \leftarrow \mathsf{Cham}_{\mathsf{csk}}(0,R,\mathcal{H}(M\|\sigma))$.
$\sigma' \leftarrow (\sigma,R')$. Output σ'.
—$\mathsf{Ver}'_{\mathsf{pk}'}(M,\sigma')$—
Parse σ' as (σ,R'). $C \leftarrow \mathsf{Com}_{\mathsf{cpk}}(\mathcal{H}(M\|\sigma),R')$.
If $\mathsf{Ver}_{\mathsf{pk}}(C,\sigma) = \mathsf{accept}$ then return accept.
Otherwise return reject.

Fig. 6 Proposed conversion in the random oracle model.

described in Fig. 6.

As in the case of the conversion in the previous section, the signer can precompute, before receiving a message, the following parts of signing computations: $C = \mathsf{Com}_{\mathsf{cpk}}(0,R)$ and $\sigma = \mathsf{Sig}_{\mathsf{sk}}(C)$. Moreover, the computation of the last part, $R' = \mathsf{Cham}_{\mathsf{csk}}(0,R,\mathcal{H}(M\|\sigma))$, is quite efficient if we set Ω to the chameleon commitment of Sect. 2.1. This means that our converted scheme Σ' satisfies on-line/off-line property if we set Ω to such chameleon commitment schemes.

Theorem 5.1. *Let S and V be the signing and verification costs of Σ, U and T be the computational cost of* Com *and* Cham *and p be the bit length of hash values of $\mathcal{H}$. We let $q = q(\kappa)$ be $(\max_{C:\text{bit string}}\{\Pr[C = \mathsf{Com}_{\mathsf{cpk}}(0,R)]\})^{-1}$. (That is, q^{-1} is the min-entropy of the random variable $\mathsf{Com}_{\mathsf{cpk}}(0,R)$.) Here the probability is in the case of where* cpk *is taken by* $\mathsf{CGen}(1^\kappa)$ *and R are taken uniformly randomly from $\mathcal{R}_\kappa$.*

Suppose that there exists an adversary which can break $(t',q_S,q_H,\varepsilon')$-SEU property of the signature scheme $\Sigma' = (\mathsf{Gen}',\mathsf{Sig}',\mathsf{Ver}')$ in the random oracle model.

Then there exists either an adversary which can break either the (t,q_S,ε)-existentially unforgeable of $\Sigma = (\mathsf{Gen},\mathsf{Sig},\mathsf{Ver})$ or (t,ε)-security (against the key only attack) of Ω.

Here

$$\begin{cases} t = t' + (S+V+U+T)q_S + (\text{lower terms}), \\ \varepsilon = (\varepsilon'/3) - (q_H+q_S)q_S/p - (q_S/q) - (\text{lower terms}). \end{cases}$$

We note that, if we set Ω to the chameleon commitment of Sect. 2.1, both the values q and p are the same as the order of the group $\mathcal{G}$.

Proof of Theorem 5.1. Let $\mathcal{A}$ be an adversary which breaks the $(t',q_S,q_H,\varepsilon')$-SEU property of Σ'. The adversary $\mathcal{A}$ is first given a public key $\mathsf{pk}' = (\mathsf{pk},\mathsf{cpk})$. $\mathcal{A}$ makes queries $M_1,\ldots,M_{q_S}$ to the signing oracle $O^{\mathsf{Sig}'}_{\mathsf{sk}'}$ adaptively, and receives the signatures $\sigma'_1 = (\sigma_1,R'_1),\ldots,\sigma'_{q_S} = (\sigma_{q_S},R'_{q_S})$ on these messages as the answers from $O^{\mathsf{Sig}'}_{\mathsf{sk}'}$. $\mathcal{A}$ finally outputs a message M and a signature $\sigma' = (\sigma,R')$. We let m_i, m, C_i, and C be $\mathcal{H}(M_i\|\sigma_i)$, $\mathcal{H}(M\|\sigma)$, $\mathsf{Com}_{\mathsf{cpk}}(m_i,R'_i)$, $\mathsf{Com}_{\mathsf{cpk}}(m,R')$.

Let ε_1, ε_2, and ε_3 be the probability that $\mathcal{A}$ will break the SEU property and the following (1), (2), and (3) will hold:

(1) $C \neq C_i$ holds for any i.
(2) $C = C_i$ holds for some i. Moreover, there is k such that, when the signing oracle computes $\sigma_k = \mathsf{Sig}_{\mathsf{sk}}(C_k)$, $M_k\|\sigma_k$ has already been queried to the random oracle by the signing oracle or the adversary.
(3) $C = C_i$ holds for some i. Moreover, there exists no such k as described in (2).

Note that the latter condition of (2) means that the equality $M_k\|\sigma_k = M_j\|\sigma_j$ holds for some $j < k$, or $\mathcal{A}$ succeeds in predicting σ_k and making query $M_k\|\sigma_k$ to the random oracle before the signing oracle computes σ_k.

Clearly, at least one of ε_1, ε_2, or ε_3 is not less than $\varepsilon'/3$. By using $\mathcal{A}$ as a subroutine, we will construct three machines $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ and will show the following facts:

- If $\varepsilon_i \geq \varepsilon'/3$ holds for at least one of $i = 1,2$, then $\mathcal{B}_i$ succeeds in breaking the (t,q_S,ε)-existential unforgeability of Σ.
- If $\varepsilon_3 \geq \varepsilon'/3$ holds, then $\mathcal{B}_3$ succeeds in breaking the (t,ε,q_S)-strong binding property against the key only attack.

This means that the theorem holds.

The case where $\varepsilon_1 \geq \varepsilon'/3$ holds: Let pk_* be a randomly selected public key of the signature scheme $\Sigma = (\mathsf{Gen},\mathsf{Sig},\mathsf{Ver})$. By using $\mathcal{A}$ as a subroutine, we construct an adversary $\mathcal{B}_1$ that can break the (t,q_S,ε)-existential unforgeability of Σ. The adversary $\mathcal{B}_1$ runs $\mathcal{A}$ as follows:

Setup: $\mathcal{B}_1$ computes $(\mathsf{cpk},\mathsf{csk}) = \mathsf{CGen}(1^\kappa)$ sets $\mathsf{pk} = \mathsf{pk}_*$ and $\mathsf{pk}' = (\mathsf{pk},\mathsf{cpk})$, provides pk' to $\mathcal{A}$, and runs $\mathcal{A}$.

Random Oracle Simulation: Let X be a query of $\mathcal{A}$. If $\mathcal{H}(X)$ has already been determined, $\mathcal{B}_1$ sends $\mathcal{H}(X)$ back to $\mathcal{A}$. Otherwise, $\mathcal{B}_1$ selects $m \in \mathbb{Z}_q$ randomly, sets $\mathcal{H}(X)$ to m, and sends $m = \mathcal{H}(X)$ back to $\mathcal{A}$.

Signing Oracle Simulation: Let M_i be the i-th queried message of $\mathcal{A}$. The adversary $\mathcal{B}_1$ selects R_i randomly, computes $C_i = \mathsf{Com}_{\mathsf{cpk}}(0,R_i)$, makes the query C_i to its signing oracle, and receives a signature σ_i on C_i as the answer.

$\mathcal{B}_1$ determines $m_i = \mathcal{H}(M_i\|\sigma_i)$ as in the case of the random oracle simulation. That is, if $\mathcal{H}(M_i\|\sigma_i)$ has not been determined yet, $\mathcal{B}_1$ takes m_i randomly and sets $m_i = \mathcal{H}(M_i\|\sigma_i)$. Otherwise, $\mathcal{B}_1$ obtains the hash value m_i from the hash table.

Then $\mathcal{B}_1$ computes $R'_i = \mathsf{Cham}_{\mathsf{csk}}(0,R_i,\mathcal{H}(M_i\|\sigma_i))$. sets $\sigma' = (\sigma,R'_i)$. One can easily show that σ'_i is a valid signature on M_i. Then $\mathcal{B}_1$ finally sends σ'_i to $\mathcal{A}$.

Extraction: Suppose that $\mathcal{A}$ succeeds in forging a message/signature pair (M,σ'). That is, suppose that $\mathcal{A}$ outputs a message M and a valid signature $\sigma' = (\sigma,R')$ on M, such that $(M,\sigma') \neq (M_i,\sigma'_i)$ holds for any i.

$\mathcal{B}_1$ determines $m = \mathcal{H}(M\|\sigma)$ as in the case of the random oracle simulation. Then $\mathcal{B}_1$ computes $C = \mathsf{Com}_{\mathsf{cpk}}(m,R')$. Since $\sigma' = (\sigma,R')$ is a valid signature on M, σ is a valid signature on C. If $C \neq C_i$ holds for any i, $\mathcal{B}_1$

outputs (C, σ). Otherwise, the simulation fails.

The number of steps until $\mathcal{B}_1$ terminates is clearly not more than t. We estimate the success probability of $\mathcal{B}_1$. $\mathcal{B}_1$ succeeds in forging a signature if $\mathcal{A}$ outputs a valid pair and $C \neq C_i$ holds for any i. Therefore, from the definition of ε_1, the probability that $\mathcal{B}_1$ will succeed in forging a signature is at least ε_1. If $\varepsilon_1 \geq \varepsilon'/3$, $\mathcal{B}_1$ breaks the (t, q_S, ε)-existensial unforgeability of Σ.

The case where $\varepsilon_2 \geq \varepsilon'/3$ holds: By using $\mathcal{A}$ as a subroutine, we construct a machine $\mathcal{B}_2$ that can break the (t, q_S, ε)-existensial unforgeability of Σ.

$\mathcal{B}_2$ computes $(\mathsf{cpk}, \mathsf{csk}) = \mathsf{CGen}(1^\kappa)$ sets $\mathsf{pk} = \mathsf{pk}_*$ and $\mathsf{pk}' = (\mathsf{pk}, \mathsf{cpk})$, provides pk' to $\mathcal{A}$ and runs $\mathcal{A}$. $\mathcal{B}_2$ simulates the random oracle similarly to $\mathcal{B}_1$, maintaining a hash table. (The hash table contains pairs (X, m) where X is a hash query and m is the answer of the query, that is, $\mathcal{H}(X) = m$.)

For each k, when $\mathcal{A}$ makes k-th signing query M_k, $\mathcal{B}_2$ simulates the initial steps of the singing oracle for Σ'. That is, $\mathcal{B}_2$ selects R_k randomly and computes $C_k = \mathsf{Com}_{\mathsf{cpk}}(0, R_k)$. Then $\mathcal{B}_2$ stops simulation of the singing oracle for Σ' for a while. (We stress that $\mathcal{B}_2$ does not make query C_k to the signing oracle for Σ at this point.)

Then $\mathcal{B}_2$ finds $M\|\sigma$ satisfying $\mathsf{Ver}_{\mathsf{pk}}(C_k, \sigma) = \mathsf{accept}$ from the hash table. If there is such σ (in this case, we call the current k an *expected* k), $\mathcal{B}_2$ checks that $C_k = C_i$ holds for some $i = 1, \ldots, k-1$, where C_i is the i-th signing query $\mathcal{B}_2$ made. If $C_k = C_i$ holds for some i, the simulation fails and $\mathcal{B}_2$ outputs a symbol fail_1 and terminates. Otherwise, $\mathcal{B}_2$ outputs (C_k, σ) as a forged pair.

In the case that k is not an expected one, that is, there is no σ in the hash table satisfying the above condition, $\mathcal{B}_2$ determines (σ_k, R_k) similarly to $\mathcal{B}_1$, and continues the simulation. (Remember that $\mathcal{B}_2$ makes a signing query C_k.)

If $\mathcal{A}$ outputs a forged pair before $\mathcal{B}_2$ finds expected k, $\mathcal{B}_2$ outputs a symbol fail_2 and terminates.

The number of steps until $\mathcal{B}_2$ terminates is clearly not more than t. We estimate the probability that $\mathcal{B}_2$ succeeds in forging a signature. From the definition of ε_2, the probaility that $\mathcal{B}_2$ does not output fail_2 is at least ε_2. Moreover, from the definition of q, the equality $C_k = C_i$ holds for the expected k and for some i with probability at most $1/q$. Therefore, $\mathcal{B}_2$ succeeds in forging a new message/signature pair (C_k, σ) with probability $\varepsilon_2 - (q_S/q)$. Therefore, if $\varepsilon_2 \geq \varepsilon'/3$, $\mathcal{B}_2$ breaks the (t, q_S, ε)-existential unforgeability of Σ.

The case where $\varepsilon_3 \geq \varepsilon'/3$ holds: By using $\mathcal{A}$ as a subroutine, we construct a machine $\mathcal{B}_3$ that can break the (t, ε)-strong binding property of Ω against the key only attack. Let cpk_* be an instance of the game for the strong binding property. The aim of $\mathcal{B}_3$ is to obtain $((M, R), (\hat{M}, \hat{R}))$ satisfying $\mathsf{Com}_{\mathsf{cpk}}(M, R) = \mathsf{Com}_{\mathsf{cpk}}(\hat{M}, \hat{R})$ and $(M, R) \neq (\hat{M}, \hat{R})$. $\mathcal{B}_3$ runs $\mathcal{A}$ as follows:

Setup: $\mathcal{B}_3$ computes $(\mathsf{pk}, \mathsf{sk}) = \mathsf{Gen}(1^\kappa)$, sets $\mathsf{cpk} = \mathsf{cpk}_*$ and $\mathsf{pk}' = (\mathsf{pk}, \mathsf{cpk})$, provides pk' to $\mathcal{A}$, and runs $\mathcal{A}$.

Random Oracle Simulation: Let X be a query of $\mathcal{A}$. If $\mathcal{H}(X)$ is already determined, $\mathcal{B}_3$ sends $\mathcal{H}(X)$ back to $\mathcal{A}$. Otherwise, $\mathcal{B}_3$ selects $m \in \mathbb{Z}_q$ randomly, sets $\mathcal{H}(X)$ to m, and sends $m = \mathcal{H}(X)$ back to $\mathcal{A}$.

Signing Oracle Simulation: Let M_i be the i-th queried message of $\mathcal{A}$. $\mathcal{B}_3$ selects $m_i, R'_i \in \mathbb{Z}_q$ randomly, and sets $C_i = \mathsf{Com}_{\mathsf{cpk}}(m_i, R'_i)$. By using the secret key sk, $\mathcal{B}_3$ computes $\sigma_i = \mathsf{Sig}_{\mathsf{sk}}(C_i)$. If the hash value corresponding to $M_i\|\sigma_i$ has already been determined, then the simulation fails and $\mathcal{B}_3$ outputs a symbol fail_1 and terminates. Otherwise, $\mathcal{B}_3$ sets the hash value $\mathcal{H}(M_i\|\sigma_i)$ to m_i. One can easily show that $\sigma'_i = (\sigma_i, R'_i)$ is a valid signature on M_i. Since Ω satisfies the uniformity property, $\sigma'_i = (\sigma_i, R'_i)$ generated by $\mathcal{B}_3$ has the same distribution as that generated by the signing oracle. $\mathcal{B}_3$ finally sends σ'_i to $\mathcal{A}$.

Extraction: Suppose that $\mathcal{A}$ outputs a message M and a valid signature $\sigma' = (\sigma, R')$ on M, such that $(M, \sigma') \neq (M_i, \sigma'_i)$ holds for any i. If there is no i satisfying $C = C_i$, the simulation fails and $\mathcal{B}_3$ outputs a symbol fail_2 and terminates.

We consider the case where $C = C_i$ holds for some i. In this case, $\mathsf{Com}_{\mathsf{cpk}}(m, R') = C = C_i = \mathsf{Com}_{\mathsf{cpk}}(m_i, R'_i)$ holds, where we let m and m_i be $\mathcal{H}(M\|\sigma)$ and $\mathcal{H}(M_i\|\sigma_i)$. If $(m, R') \neq (m_i, R'_i)$ holds, this means that $\mathcal{B}_3$ succeeds in computing a collision pair $((m, R'), (m_i, R'_i))$ for Com. Otherwise, the simulation fails and $\mathcal{B}_3$ outputs a symbol fail_3 and terminates.

The number of steps until $\mathcal{B}_3$ terminates is clearly not more than t.

We next estimate the probability that $\mathcal{B}_3$ will succeed in attacking the strong binding property. We suppose the following three events occur: (a) $\mathcal{A}$ succeeds in forging a signature, (b) $C = C_i$ holds for some i, and (c) there is no k such that, when the signing oracle computes $\sigma_k = \mathsf{Sig}_{\mathsf{sk}}(C_k)$, $M_k\|\sigma_k$ has already been written in the hash table. Then, $\mathcal{B}_3$ does not output fail_1 nor fail_2. Further, from the definition of ε_3, these three events occur with probability at least $\varepsilon_3 \geq \varepsilon'/3$.

We estimate the probability that $\mathcal{B}_3$ will output fail_3. $\mathcal{B}_3$ outputs fail_3 only if there exists i such that $\mathcal{H}(M\|\sigma) = \mathcal{H}(M_i\|\sigma_i)$ and $R' = R'_i$ hold. On the other hand, if $\mathcal{A}$ succeeds in computing a valid new message-signature pair, $(M, (\sigma, R')) = (M, \sigma') \neq (M_i, \sigma'_i) = (M_i, (\sigma_i, R'_i))$ for all i. Therefore, $\mathcal{B}_3$ outputs fail_3 only if there exists i such that $M\|\sigma \neq M'_i\|\sigma'_i$ and $\mathcal{H}(M\|\sigma') = \mathcal{H}(M_i\|\sigma'_i)$ hold.

Here we can estimate the probability that

$$^{\exists}\ell,\ ^{\exists}X \in \text{(hash table)} :\\ X \neq M_\ell\|\sigma_\ell \wedge \mathcal{H}(X) = \mathcal{H}(M_\ell\|\sigma_\ell)$$

will hold is at most $(q_H + q_S)q_S/p$. (Recall that $\mathcal{A}$ and the signing oracle $O^{\mathsf{Sig}'}_{\mathsf{sk}'}$ makes at most q_H and q_S hash queries respectively. Recall also that hash values are randomly selected p bit strings.) Then, the probability that $\mathcal{B}_3$ outputs fail_3 is clearly at most $(q_H + q_S)q_S/p$.

From the above discussion, the probability that $\mathcal{B}_3$ will succeed in attacking the strong binding property is at least ε. □

6. Concrete Conversions

We can implement our proposed conversions from Sects. 4 and 5, by using the chameleon commitment scheme from Sects. 3.3 and 2.1 respectively. Then, we can obtain the concrete conversions based on the discrete logarithm assumptions. However, we can subtly simplify such conversions in order to obtain more efficient scheme. Figure 7 and Fig. 8 shows these simplified conversions. In these figures, the signing algorithm is simplified. That is, the signer computes not the commitment $C = g^{\mathcal{H}(0)}h^r$ of the message 0 but $C = g^w$. One can easily show that the simplified schemes are also secure.

From the proofs of Theorems 3.2, 4.1, and 5.1, the following two facts hold:

Theorem 6.1. *Suppose that there exists a $(t', q_S, q_H, \varepsilon')$-adversary against the SEU property of the signature scheme $\Sigma' = (\mathsf{Gen}', \mathsf{Sig}', \mathsf{Ver}')$ of Fig. 7. Let S' be the signing cost of Σ'.*

Then there exists an adversary that can break either the (t, q_S, ε)-existential unforgeability of the underlying signature scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$, the (t, ε)-discrete logarithm problem in $\mathcal{G}$, or (t, ε)-collision resistant of H. Here

$$t = t' + q_S S' + (\text{lower terms}),$$
$$\varepsilon = \tfrac{\varepsilon'}{4} - (\text{lower terms}).$$

Theorem 6.2. *Suppose that there exists an adversary that can break $(t', q_S, q_H, \varepsilon')$-SEU property of the signature scheme $\Sigma' = (\mathsf{Gen}', \mathsf{Sig}', \mathsf{Ver}')$ of Fig. 8 in the random oracle model. Let S' be the signing cost of Σ' and E be the exponentiation cost on $\mathcal{G}$.*

Then there exists an adversary that can break either the (t, q_S, ε)-existential unforgeability of the underlying signature scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$, or the (t, ε)-discrete logarithm problem in $\mathcal{G}$. Here

$$\begin{cases} t = t' + q_S(S' + E) + (\text{lower terms}), \\ \varepsilon = \frac{\varepsilon'}{9} - \frac{(q_H+q_S)q_S}{3q} - (\text{lower terms}). \end{cases}$$

—$\mathsf{Gen}'(1^\kappa)$—
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{SGen}(1^\kappa)$, $(x, y) \leftarrow {\mathbb{Z}_q}^2$, $g \leftarrow \mathcal{G}$, $(h_1, h_2) \leftarrow (g^x, g^y)$.
$\mathsf{pk}' \leftarrow (\mathsf{pk}, g, h_1, h_2)$, $\mathsf{sk}' \leftarrow (\mathsf{sk}, x, y)$. Output $(\mathsf{pk}', \mathsf{sk}')$.
—$\mathsf{Sig}'_{\mathsf{sk}'}(M)$—
$w \in \mathbb{Z}_q$, $C \leftarrow g^w$, $\sigma \leftarrow \mathsf{Sig}_{\mathsf{sk}}(C)$,
Randomly choose $r', s' \in \mathbb{Z}_q$ satisfying $w = H(M) + r'x + s'y$.
$\sigma' \leftarrow (\sigma, r', s')$. Output σ'.
—$\mathsf{Ver}'_{\mathsf{pk}'}(M, \sigma')$—
Parse σ' as (σ, r', s'). $C \leftarrow g^{H(M)}{h_1}^{r'}h_2^{s'}$.
If $\mathsf{Ver}_{\mathsf{pk}}(C, \sigma) = \mathsf{accept}$ then return accept.
Otherwise return reject.

Fig. 7 Concrete conversion in the standard model.

—$\mathsf{Gen}'(1^\kappa)$—
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{SGen}(1^\kappa)$, $x \leftarrow \mathbb{Z}_q$, $g \leftarrow \mathcal{G}$, $h \leftarrow g^x$.
$\mathsf{pk}' \leftarrow (\mathsf{pk}, g, h)$, $\mathsf{sk}' \leftarrow (\mathsf{sk}, x)$. Output $(\mathsf{pk}', \mathsf{sk}')$.
—$\mathsf{Sig}'_{\mathsf{sk}'}(M)$—
$w \in \mathbb{Z}_q$, $C \leftarrow g^w$, $\sigma \leftarrow \mathsf{Sig}_{\mathsf{sk}}(C)$,
Randomly choose $r' \in \mathbb{Z}_q$ satisfying $w = \mathcal{H}(M) + rx$.
$\sigma' \leftarrow (\sigma, r')$. Output σ'.
—$\mathsf{Ver}'_{\mathsf{pk}'}(M, \sigma')$—
Parse σ' as (σ, r'). $C \leftarrow g^{\mathcal{H}(M)}h^{r'}$.
If $\mathsf{Ver}_{\mathsf{pk}}(C, \sigma) = \mathsf{accept}$ then return accept.
Otherwise return reject.

Fig. 8 Concrete conversion in the random oracle model.

6.1 Estimations of Securities

We estimate the security of the converted schemes more intuitively. In order to do it, we introduce the notions, which we call *difficulty*. For an adversary $\mathcal{X}$ against (t, ε)-discrete logarithm problem in $\mathcal{G}$, we let $\mathcal{X}^*$ be an adversary which executes $\mathcal{X}$ until $\mathcal{X}$ succeeds in solving the problem. Then $\mathcal{X}^*$ solves the discrete logarithm problem with t/ε steps on average and with success probability 1. This means that we can use the value t/ε in order to estimate how difficult one solves the discrete logarithm problem. Therefore, we say that the discrete logarithm problem has *difficulty* T if there is no (t, ε)-adversary $\mathcal{X}$ satisfying $t/\varepsilon < T$. We also define the difficulties of the existential unforgeability and the SEU property similarly.

Then we can obtain the following two facts:

Proposition 6.3. *Let S be the signing cost of a signature scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$, and E be the exponentiation cost on a group $\mathcal{G}$. Suppose that the existential unforgeability of Σ, the discrete logarithm problem in $\mathcal{G}$, and the collision resistance problem of H have difficulty T_1, T_2, and T_3, respectively. Let C_1 and C_2 be $2 + 2(E/S)$ and $20(1 + E/S)$. Let T'_{sm} and T'_{ro} be the difficulties of the SEU property of the signature scheme described in Fig. 7 and 8 respectively.*

Then the following inequalities hold:

$$T'_{\mathsf{sm}} \geq \min\{T_1, T_2, T_3\}/C_1 + (\text{lower terms}). \quad (1)$$
$$T'_{\mathsf{ro}} \geq \min\{T_1, T_2\}/C_2 + (\text{lower terms}). \quad (2)$$

Proof. Equation (1) clearly follows from Theorem 6.1. Therefore, we here only show equation (2).

We will show that there exists an adversary $\mathcal{A}_0$ which can break the SEU property of Σ' within $2E2^{\kappa/2} +$ (lower terms) step and with the probability 1. Therefore, by substituting $\mathcal{A}$ to $\mathcal{A}_0$ (if we need), we can assume that $t'/\varepsilon' \leq 2^{\kappa/2} +$ (lower terms) holds.

From the definition of q_S and q_H, the inequalities $Sq_S \leq t'$ and $q_H \leq t'$ hold. Since $S' = S + E +$ (lower terms) holds, it follows that

$$\begin{aligned} t &\leq t' + q_S(S' + E) \leq (1 + (S' + E)/S)t' \\ &= (1 + (S + 2E)/S)t' = (2 + 2E/S)t', \\ (q_H + q_S)q_S/(3q\varepsilon') &\leq (t' + t'/S)(t'/S)/(3q\varepsilon') \\ &\simeq t'^2/(3qS\varepsilon') \leq (t'/\varepsilon')^2/(3qS) \leq (2^{\kappa/2})^2/(32^{\kappa-1}S) \\ &= 2/3S, \text{ and} \\ \varepsilon &= \varepsilon'/9 - (q_H + q_S)q_S/(3q) \\ &= \varepsilon' \cdot (1/9 - (q_H+q_S)q_S/(3q\varepsilon')) \geq \varepsilon' \cdot (1/9 - 2/3S) \\ &\geq \varepsilon'/10, \end{aligned}$$

	[5]	Conv. of Fig. 7	Conv. of Fig. 8
Condition on Σ	Partitioned	Nothing	Nothing
Model	Standard	Random Oracle	Standard
Reduction	Tight	Tight	Tight
Precomputation before Signing	0	$S + E$	$S + E$
Signing using Precomp. data	$S + E$	0	0
Total Signing Cost	$S + E$	$S + E$	$S + E$
Verification Cost	$V + E$	$V + E$	$V + E$
Signature Length	$\|\sigma\| + \|q\|$	$\|\sigma\| + \|q\|$	$\|\sigma\| + 2\|q\|$

Fig. 9 Comparison.

(because $S \gg 0$ holds if $\kappa \gg 0$). Hence, it follows that

$$\begin{aligned}\min\{T_1, T_2\} &\leq t/\varepsilon \leq (2 + 2E/S)t'/(\varepsilon'/10) \\ &= 20(1 + E/S) \cdot (t'/\varepsilon') = C_0 \cdot (t'/\varepsilon').\end{aligned}$$

Hence, it follows that $\min\{T_1, T_2\} \leq C_0 \cdot T'$. Therefore, $T' \leq \min\{T_1, T_2\}/C_0$ holds.

We finally construct $\mathcal{A}_0$. $\mathcal{A}_0(\mathsf{pk}')$ computes the discrete logarithm x of (g, h) by using the Baby Step and Giant Step (BSGS) algorithm [3], sends an arbitrarily message M to the signing oracle as a query, receives the answer $\sigma' = (\sigma, r)$, and computes $m = \mathcal{H}(M||\sigma)$ and $C = g^m h^r$. Then $\mathsf{Ver}_{\mathsf{pk}}(C, \sigma) = \mathsf{accept}$ holds. $\mathcal{A}_0$ then selects an arbitrarily message $M_0 \neq M$, computes $m_0 = \mathcal{H}(M_0||\sigma)$, selects $r_0 \in \mathbb{Z}_q$ satisfying $m_0 + r_0 x = m + rx \bmod q$, sets $\sigma_0 = (\sigma, r_0)$, and outputs (M_0, σ_0). One can easily show that σ_0 is a valid signature on $M_0 \neq M$. Since BSGS algorithm requires $2^{\kappa/2}$ + (lower terms) steps, the number of steps of $\mathcal{A}_0$ is also $2^{\kappa/2}$ + (lower terms). □

6.2 Comparison of Efficiencies

We finally compare the efficiency of the converted signature schemes of Fig. 7 and Fig. 8 with the signature scheme transformed by the previous conversion [5]. See Fig. 9. In this figure, S and V represent the computational cost of the signing and verifying algorithms of the original signature scheme Σ, respectively. The value E represents the exponentiation cost on $\mathcal{G}$, $|\sigma|$ represents the bit length of a signature of Σ, and $|q|$ represents the bit length of q.

We assume that one computes $g^m h^r$ by using the simultaneous exponentiation technique [15]. That is, we assume that the computational cost to compute $g^m h^r$ is equal to E. We also assume that the computational cost of a multiplication on $\mathcal{G}$ and a hashing are very small.

The conversion reported in [5] is applicable for a signature scheme that satisfies the partitioned property [5]. However, there are signature schemes, which seems to be non-partitioned, such as DSS, the Camenisch-Lysyanskaya scheme [7], and Okamoto scheme [20]. In contrast, our two conversions are applicable to any signature scheme.

The signing costs of all of three schemes are equal. However, in the case of schemes transformed by our conversions, signers can precompute almost all signing operations before they are given messages.

The signature length of the signature scheme of Fig. 7 is longer than that of Fig. 8. But the security of the former scheme is proved without assuming the random oracle, although that of the latter scheme is proved only when one assumes the existence of the random oracle.

7. Conclusion

We defined chosen message security for a chameleon commitment scheme, and constructed two chameleon commitment schemes which are secure against the chosen message attack. The first proposed scheme was constructed by using group operations, and is secure against the chosen message attack if the group satisfies the discrete logarithm assumption. The second proposed scheme was generally constructed by using two chameleon commitment schemes, and is secure against the chosen message attack if the original two chameleon commitments are secure against the key only attack.

By using chameleon commitments, we then constructed two conversions, which were the first that could transform any secure signature scheme to a SEU signature scheme. There were trade-off between the two conversions. The first conversion did not use the random oracle but required the chosen message attack security for the chameleon commitment scheme. In contrast, the second conversion used the random oracle but only required the key only attack security for the chameleon commitment scheme.

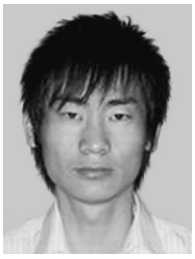
The proposed two conversions ensured the tight security reduction to the underlying security assumptions. Moreover, signers of the converted schemes could precompute almost all operations on the signing before they were given a message, (if we chose appropriate chameleon commitment scheme). Therefore, the signer could generate signatures quite efficiently.

References

[1] J.H. An, Y. Dodis, and T. Rabin, "On the security of joint signature and encryption," Eurocrypt 2002, pp.83–107, 2002.
[2] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," ACM Conference on Computer and Communications Security 1993, pp.62–73, 1993.
[3] I.F. Blake, G. Seroussi, and N.P. Smart, Elliptic Curve in Cryptography, Cambridge University Press, 1999.
[4] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," Crypto 2004, pp.41–55, 2004.
[5] D. Boneh, E. Shen, and B. Waters, "Strongly unforgeable signatures based on computational Diffie-Hellman," PKC 2006, pp.229–240, 2006.

[6] R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," Eurocrypt 2004, pp.229–235, 2004.
[7] J. Camenisch and A. Lysyanskaya, "A signature scheme with efficient protocols," SCN 2002, pp.268–289, 2002.
[8] S. Contini, A.K. Lenstra, and R. Steinfeld, "VSH, an efficient and provable collision-resistant hash function," Eurocrypt 2006, pp.165–182, 2006.
[9] I. Damgård, "Towards practical public key systems secure against chosen ciphertext attacks," CRYPTO'91, pp.445–456, 1991.
[10] D. Dolev, C. Dwork, and M. Naor, "Non-malleable cryptography," SIAM J. Comput., vol.30, no.2, pp.391–437, 2000.
[11] S. Goldwasser, S. Micali, and R.L. Rivest, "A digital signature scheme secure against adaptivechosen-message attacks," SIAM J. Comput., vol.17, no.2, pp.281–308, 1988.
[12] O. Goldreich, Foundations of Cryptography, Vol.I Basic Tools, Cambridge University Press, 2001.
[13] H. Krawczyk and T. Rabin, "Chameleon hashing and signatures," 1997. http://ibm.com/security/chameleon.ps, http://eprint.iacr.org/1998/010
[14] H. Krawczyk and T. Rabin, "Chameleon signatures," NDSS 2000, pp.143–154, 2000.
[15] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.
[16] T.P. Pedersen, "A threshold cryptosystem without a trusted party (extended abstract)," EUROCRYPT'91, pp.522–526, Springer-Verlag, 1991.
[17] P. Paillier and D. Vergnaud, "Discrete-log-based signatures may not be equivalent to discrete log," Asiacrypt 2005, pp.1–20, 2005.
[18] C. Rackoff and D.R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," CRYPTO'91, pp.433–444, 1991.
[19] J. Rompel, "One-way functions are necessary and sufficient for secure signatures," STOC 1990, pp.387–394, 1990.
[20] T. Okamoto, "Efficient blind and partially blind signatures without random oracles," TCC 2006, pp.80–99, 2006.
[21] A. Shamir and Y. Tauman, "Improved online/offline signature schemes," Crypto 2001, pp.355–367, 2001.
[22] V. Shoup, "Lower bound for discrete logarithms and related problems," Eurocrypt'97, pp.256–266, 1997.
[23] R. Steinfeld, J. Pieprzyk, and H. Wang, "How to strengthen any weakly unforgeable signature into a strongly unforgeable signature," CT-RSA, 2007.
[24] Extended Abstract Version of this paper, Indocrypt 2006, pp.191–205, 2006.
[25] B. Waters, "Efficient identity-based encryption without random oracles," Eurocrypt 2005, pp.114–127, 2005.
[26] R. Zhang, "Tweaking TBE/IBE to PKE transforms with chameleon hash functions," ACNS'07, Full version is available at http://staff.aist.go.jp/r-zhang/research/papers.html#CCATBE

Isamu Teranishi received B.S. and M.E. degrees in mathematics in 2000 and 2002 respectively, from Tokyo Institute of Technology. He joined the NEC as a researcher in 2002.



Takuro Oyama received B.E. in Computer Science and M.E. in Communications and Integrated Systems from Tokyo Institute of Technology in 2005 and 2007 respectively. He has joined Fujitsu Limited since 2007 and engaged in Software Development for prevention against information leak.

Wakaha Ogata received B.S., M.E. and D.E. degrees in electrical and electronic engineering in 1989, 1991 and 1994, respectively, from Tokyo Institute of Technology. From 1995 to 2000, she was an Assistant Professor at Himeji Institute of Technology. Since 2000, she has been an Associate Professor at Tokyo Institute of Technology. Her current interests are information security and cryptography.