Constructing Practical Signcryption KEM from Standard Assumptions without Random Oracles

Xiangxue Li[1,2], Haifeng Qian[1,⋆], Yu Yu[3], Yuan Zhou[4], and Jian Weng[5]

[1] Department of Computer Science and Technology, East China Normal University
hfqian@cs.ecnu.edu.cn
[2] State Key Laboratory of Integrated Services Networks, Xidian University
[3] Institute for Interdisciplinary Information Sciences, Tsinghua University
[4] Network Emergency Response Technical Team/Coordination Center, China
[5] Department of Computer Science, Jinan University

Abstract. We present a direct construction for signcryption Key Encapsulation Mechanism (KEM) without random oracles under standard complexity assumptions. Chosen-ciphertext security is proven in the standard model under the DBDH assumption, and unforgeability is proven in the standard model under the CDH assumption. The proof technique allows us to achieve strong unforgeability from the weakly unforgeable Waters signature. The validity of the ciphertext of our signcryption KEM can be verified publicly, without knowledge of the decryption key.

Keywords: Signcryption, KEM, Standard Model, Standard Assumption.

1 Introduction

Signcryption [25] provides confidentiality and non-repudiation simultaneously for the messages sent over an insecure channel, at lower costs of computation and communication than those required in both signature-then-encryption ($\mathcal{S}t\mathcal{E}$) and encryption-then-signature ($\mathcal{E}t\mathcal{S}$) approaches. Thus, protocols based on signcryption are considerably more efficient than those traditional approaches that combine both encryption and signature. One may apply signcryption to obtain a performance-enhanced protocol which contributes to the practical and engineering side of real-world applications [20,21,12,22,11].

For long messages, it is quite inefficient in the real-life applications to apply signcryption directly. Inspired by traditional hybrid encryption techniques, Dent [9] generalized the KEM paradigm to the signcryption setting by proposing new security criteria and a construction for the signcryption KEM (SC-KEM) to provide in KEM the authentication service. Such a construction combines the convenience of a signcryption with the efficiency of a symmetric-key system [8]. By using such a construction, a random session key is first encapsulated by a signcryption KEM, then the data (plaintext) is encrypted by the session key, and finally two ciphertexts are both sent over an insecure channel.

⋆ Corresponding author.

1.1 The State of the Art

Dent [9,10] introduced the concept of signcryption KEM which includes an authentication in KEM by constructing two signcryption KEM schemes with insider security and outsider security, respectively. A signcryption scheme is outsider secure if it is secure against attacks made by any third party, i.e., attacks made by an entity who is neither the sender nor the receiver. This is a weaker notion of security than has been traditionally dealt with by signcryption schemes, a notion known as insider security. Actually, the work [10] improved the model in [9] (which only covers outsider security) by providing a signcryption KEM with insider security such that the resultant scheme is secure against attacks against the confidentiality of the message made by any third party and from forgery attacks made by any person except the sender.

However, insider security proposed by Dent [10] is only considered for authenticity. In other words, the model in [10] allows an attacker to recover the symmetric key generated by signcryption KEM during the attacks. Comparatively, the stronger notion named full insider security [2,23] protects the sender's authenticity even against the receiver, and the receiver's privacy even against the sender, at the same time.

Recently, Tan proposed in [23] a signcryption KEM with full insider security. Much different from those schemes [9,10], Tan's SC-KEM is proven secure in the standard model whose security does not rely on random oracles. Another signcryption KEM in the standard model was presented in [18] which is shown more efficient than Tan's scheme in terms of computational cost and communication overhead.

1.2 Motivation

Concrete constructions for signcryption KEM are evaluated according to the following perspectives: (1) the complexity assumptions on which security of the construction is based; (2) the expansion of a single ciphertext; (3) the operational assumption of setting up the construction practically; and other interesting features (e.g., public verifiability of the ciphertext).

All existing constructions of SC-KEM need the recipient's private keys to verify the validity of the ciphertexts. Hence these schemes can not be used in applications where a ciphertext need to be validated by any third party that knows the public key of the sender as in usual signature scheme. As a technically higher standard, public verifiability of ciphertexts enables any member of the public to independently fully verify the accuracy of a ciphertext [1,14]. Additionally, the constructions in [9] and [10] are proven secure in the random oracle model that serves as a heuristic. Although those in [18] and [23] are without random oracles, yet they utilize standard signatures as building blocks, thus we can't reduce the computation and the size of a single ciphertext fewer than the underlying signature, and the readers may refer to section 4 for more details on the sizes of standard model based signatures; on the other hand, the construction in [18] is based on non-standard GHDH assumption [16]. For all, the question of constructing a signcryption KEM that is secure under the standard assumptions without random oracles (and achieving public verifiability) remains open.

1.3 Our Contributions

In this paper we provide an elegant construction for SC-KEM to give a positive answer to the question. Our signcryption KEM achieves the following desirable features simultaneously, compared with the previous constructions.

1. *Full Insider Security* (FIS): Our SC-KEM is proven secure in the standard model with respect to insider adversaries.
2. *Standard Complexity Assumptions*: Security of our SC-KEM relies on the well-established DBDH and the CDH assumptions. Prior to our work, the SC-KEM scheme [18] requires non-standard assumption (i.e., the Gap Hashed Diffie-Hellman assumption [16]) to prove security in the standard model.
3. *Small Ciphertext Expansion*: The ciphertext of our SC-KEM consists of three elements of $\mathbb{G}$. It outperforms all known standard model-based constructions (that use strongly unforgeable signatures as building block) because according to the state of the art [19] a strongly unforgeable signature contains at least 3 group elements since each Waters signature, the only known signature secure under CDH assumption without random oracles, has two group elements. Our construction is also comparable to, though not quite as efficient as, the Dent signcryption KEM schemes [9,10] in the random oracle model.
4. *Additional Interesting Features*: On one hand, our construction enjoys *simple setup operation* since it only needs one key generation algorithm to generate the keys of both the sender and the receiver. Whereas, two different key generation algorithms are required in [18] and [23] respectively to generate the key pairs of the sender and the receiver. Thus, the setup process of these SC-KEM schemes is more complicate than that of ours. On the other hand, in the standard model based SC-KEM schemes in [18,23] only the receiver has the capability of verifying the correctness of a ciphertext as the private key of the receiver is required in verifying operation; whereas in our SC-KEM scheme, a given ciphertext can get checked for validity solely based on the knowledge of the public keys of the parties.

2 Preliminaries

2.1 Bilinear Group

Consider the following setting: Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p; the group action on $\mathbb{G}$, $\mathbb{G}_T$ can be computed efficiently; g is a generator of $\mathbb{G}$; $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is an efficiently computable map with the following properties [3,4,24]: Bilinear: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$; Efficiently computable: $e(u, v)$ is efficiently computable for any input pair $(u, v) \in \mathbb{G} \times \mathbb{G}$; Non-degenerate: $e(g, g) \neq 1$. We say that $\mathbb{G}$ is a bilinear group if it satisfies these requirements.

2.2 Complexity Assumptions

Definition 1 (DBDH). *Let a, b, c and z be random from $\mathbb{Z}_p$, g the generator of $\mathbb{G}$ of prime order p. The (t, ε)-DBDH assumption says that there is no algorithm $\mathcal{A}$ that can distinguish the tuple $(g^a, g^b, g^c, e(g,g)^{abc})$ from the tuple $(g^a, g^b, g^c, e(g,g)^z)$ in time t with advantage ε, where the advantage of $\mathcal{A}$ is defined as the probability*

$$\mathrm{Adv}_{\mathcal{A}}^{\mathsf{DBDH}} = \left|\Pr[\mathcal{A}(g^a, g^b, g^c, e(g,g)^{abc}) = 1] - \Pr[\mathcal{A}(g^a, g^b, g^c, e(g,g)^z) = 1]\right|.$$

Definition 2 (CDH). *In a bilinear group $\mathbb{G}$, the computational Diffie-Hellman problem is: given $(g, g^a, g^b) \in \mathbb{G}^3$ for some (randomly chosen) $a, b \in_R \mathbb{Z}_p$, to find $g^{ab} \in \mathbb{G}$. The success probability of an algorithm $\mathcal{A}$ in solving the CDH problem on $\mathbb{G}$ is defined as*

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{CDH}} \overset{\mathrm{def}}{=} \Pr\left[\mathcal{A}(g, g^a, g^b) = g^{ab} : a, b \xleftarrow{R} \mathbb{Z}_p\right].$$

The probability is over the random choice of g from $\mathbb{G}$, of a, b from $\mathbb{Z}_p$, and the coin tosses of $\mathcal{A}$. $\mathcal{A}$ (t, ε)-breaks the CDH problem on $\mathbb{G}$ if $\mathcal{A}$ runs in time at most t, and $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{cdh}}$ is at least ε.

2.3 Collision Resistant Hash Function

Definition 3 (CRHF). *Let $\mathcal{H} = \{H_k\}$ be a hash family of functions $H_k : \{0,1\}^* \to \{0,1\}^n$ indexed by k. We say that algorithm $\mathcal{A}$ (t, ε_{cr})-breaks the collision-resistance of $\mathcal{H}$ if*

$$\Pr[\mathcal{A}(k) = (x, x') : H_k(x) = H_k(x'), x \neq x'] \geq \varepsilon_{cr},$$

where the probability is over the random choice of k and the random bits of $\mathcal{A}$. $\mathcal{H}$ is (t, ε_{cr})-collision-resistant if no t-time adversary has advantage at least ε_{cr} in breaking the collision-resistance of $\mathcal{H}$.

2.4 Definition of Signcryption KEM

Definition 4 (SC-KEM). *A signcryption KEM consists of three algorithms:*

$\mathsf{KeyGen}(1^\lambda)$: *key generation algorithm, on input a security parameter λ, outputs the sender's public/private key pair (pk_s, sk_s) and the receiver's public/private key pair (pk_r, sk_r). We write $(pk, sk) = \mathsf{KeyGen}(1^\lambda)$.*

$\mathsf{KeyEnc}(sk_s, pk_r)$: *key encapsulation algorithm, on input the sender's private key sk_s and the receiver's public key pk_r, outputs a symmetric key K which may be used in the subsequent data encapsulation mechanism, and a ciphertext C which is an encapsulation of the key K. We write $(K, C) = \mathsf{KeyEnc}(sk_s, pk_r)$.*

$\mathsf{KeyDec}(pk_s, sk_r, C)$: *key decapsulation algorithm, on input the sender's public key pk_s, the receiver's private key sk_r and the encapsulation C of some symmetric key K, outputs either the symmetric key K or the error symbol $\perp$ in case the ciphertext is not valid. We write $K = \mathsf{KeyDec}(pk_s, sk_r, C)$.*

Correctness requires that for all public/private key pair $(pk_s, sk_s), (pk_r, sk_r)$ it follows $K = \mathsf{KeyDec}(pk_s, sk_r, C)$ for all $(K, C) = \mathsf{KeyEnc}(sk_s, pk_r)$.

2.5 Security Model of SC-KEM

We borrow the following models which are commonly used in the literature [9,10,18,23]. The paper focuses on the direct construction for SC-KEM without random oracles under standard complexity assumptions, in the security models.

Confidentiality. The attack model [18,23] of confidentiality for a signcryption KEM is defined in the following game, termed the $\mathsf{IND\text{-}CCA2}$ game, played between a hypothetical challenger $\mathcal{C}$ and a two-phase attacker $\mathcal{A}$.

- **Setup**: On input a given security parameter λ, the challenger $\mathcal{C}$ runs the key generation algorithm $\mathsf{KeyGen}(1^\lambda)$ to produce the sender's key pair $(pk_s^\star, sk_s^\star)$ and the receiver's key pair $(pk_r^\star, sk_r^\star)$, and sends $(pk_s^\star, sk_s^\star)$ and $pk_r^\star$ to the attacker $\mathcal{A}$, while keeping $sk_r^\star$ secret.
- **Phase 1**: During this phase, $\mathcal{A}$ may make the polynomially bounded queries of key decapsulation. In a key decapsulation query, $\mathcal{A}$ submits to the challenger $\mathcal{C}$ a ciphertext C associated with the sender's public key pk_s for key decapsulation. Herein, the public key pk_s may be generated by $\mathcal{A}$ as it wishes. The challenger $\mathcal{C}$ performs key decapsulation operation for $\mathcal{A}$ in the algorithm KeyDec by using the private key $sk_r^\star$ and then sends the result $K = \mathsf{KeyDec}(pk_s, sk_r^\star, C)$ or $\perp$ (if C is not valid) to $\mathcal{A}$.
- **Challenge**: At the end of Phase 1, $\mathcal{C}$ performs the algorithm KeyEnc by using the private key $sk_s^\star$ and the public key $pk_r^\star$, and obtains the result $(K_0^\star, C^\star) = \mathsf{KeyEnc}(sk_s^\star, pk_r^\star)$. $\mathcal{C}$ also chooses a random bit $b \in \{0,1\}$ and a random symmetric key $K_1^\star$ with the requirement that $K_1^\star$ and $K_0^\star$ are of the same length. Lastly, $\mathcal{C}$ gives $\mathcal{A}$ the tuple $(K_b^\star, C^\star)$ as the challenge.
- **Phase 2**: During this phase, $\mathcal{A}$ may make the queries as in **Phase 1**, while differently we do not allow $\mathcal{A}$ to query the key decapsulation for the ciphertext $C^\star$ under the the sender's public key $pk_s^\star$.
- **Guess**: Eventually, $\mathcal{A}$ outputs a bit b', and it wins the game if $b = b'$.

The advantage of the adversary $\mathcal{A}$ is defined as the probability $\mathrm{Adv}_{\mathcal{A}}^{\mathsf{IND}} = |2\Pr[b = b'] - 1|$.

Definition 5 (Confidentiality). *We say $\mathcal{A}$ (t, q_d, ε)-breaks the IND-CCA2 security of the signcryption KEM, if $\mathcal{A}$ wins the* $\mathsf{IND\text{-}CCA2}$ *game with the advantage ε in time t after making q_d key decapsulation queries. A signcryption KEM is said to achieve the IND-CCA2 security if no polynomially bounded adversary has a non-negligible advantage in winning the* $\mathsf{IND\text{-}CCA2}$ *game.*

Unforgeability. The notion of strongly existential unforgeability for a signcryption KEM [18,23] is defined by the SUF game, played between a hypothetical challenger $\mathcal{C}$ and an attacker $\mathcal{F}$ below. For a given security parameter λ:

- **Setup:** The challenger $\mathcal{C}$ runs the key generation algorithm $\mathsf{KeyGen}(1^\lambda)$ (defined in definition 4) to produce the sender's key pair $(pk_s^\star, sk_s^\star)$ and the receiver's key pair $(pk_r^\star, sk_r^\star)$, and sends $pk_s^\star$ and $(pk_r^\star, sk_r^\star)$ to the attacker $\mathcal{F}$, while keeping $sk_s^\star$ secret.

- **Attack:** During this phase, $\mathcal{F}$ may make the polynomially bounded queries of key encapsulation. In a key encapsulation query, the challenger $\mathcal{C}$ performs key encapsulation operation for $\mathcal{F}$ in the algorithm KeyEnc by using the private key $sk_s^\star$ and the public key $pk_r^\star$, obtains the result $(K, C) = \mathsf{KeyEnc}(sk_s^\star, pk_r^\star)$, and sends C to $\mathcal{F}$.
- **Forgery:** Eventually, the attacker $\mathcal{F}$ outputs a ciphertext $C^\star$ with the requirement that $C^\star$ is not one of the outputs of the key encapsulation queries. $\mathcal{F}$ wins the game if $C^\star$ is valid, i.e., $\mathsf{KeyDec}(pk_s^\star, sk_r^\star, C^\star) \neq \perp$.

The advantage of $\mathcal{F}$ is defined as the probability of success in winning the game: $\mathrm{Adv}_{\mathcal{F}}^{\mathsf{SUF}} = \Pr[\mathsf{Win}]$.

Definition 6 (Unforgeability). *We say the signcryption KEM is (t, q_e, ε)-forgeable if $\mathcal{F}$ wins the* SUF *game with the advantage ε in time t after making q_e key encapsulation queries. A signcryption KEM achieves strongly existential unforgeability if no polynomially-bounded adversary can win the* SUF *game with non-negligible advantage.*

Definition 7 (Public Verifiability). *We say the signcryption KEM has public verifiability of ciphertexts if any member of the public can independently fully verify the accuracy of a ciphertext without relying on any secret information.*

3 The Proposed Signcryption KEM

Let $\mathbb{G}$ be a group of prime order p, for which there exists an efficiently computable bilinear map into $\mathbb{G}$. The size of the group is determined by the security parameter. Additionally, let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ denote the bilinear map and g be the corresponding generator, along with u', u_1, u_2, $\ldots$, u_n, f, h, v, $w \in \mathbb{G}$. Let $G : \{0,1\}^* \to \{0,1\}^n$, $H : \{0,1\}^* \to \mathbb{Z}_p$ be two collision resistant hash functions. Our construction is described as follows.

$\mathsf{KeyGen}(1^\lambda)$: A probabilistic polynomial-time sender/receiver key generation algorithm, chooses $x_s, x_r \in_R \mathbb{Z}_p$, sets $sk_s = x_s$, $pk_s = g^{x_s}$, $sk_r = x_r$, $pk_r = g^{x_r}$, and outputs the public/private key pair (pk_s, sk_s) for the sender and the public/private key pair (pk_r, sk_r) for the receiver.

$\mathsf{keyEnc}(sk_s, pk_r)$:

1. Randomly choose $k, \ell \in_R \mathbb{Z}_p$.
2. Compute $K = e(h, pk_r)^k$, $\sigma_1 = g^k$, $\sigma_2 = g^\ell$, $t_1 = G(\sigma_1, pk_s, pk_r)$,
3. Let $\mathcal{T} \subset \{1, 2, \ldots, n\}$ be the set of indices such that $t_1[i] = 1$ where $t_1[i]$ is the i-th bit of t_1.
4. Compute $t_2 = H(\sigma_1, \sigma_2, pk_s, pk_r)$, $\sigma_3 = f^{x_s} \cdot \left(u' \prod_{i \in \mathcal{T}} u_i\right)^\ell (v^{t_2} w)^k$.
5. Let $C = (\sigma_1, \sigma_2, \sigma_3)$ and return (K, C).

Different from the constructions in [18,23], a ciphertext C of our SC-KEM scheme is only composed of three elements in $\mathbb{G}$.

$\mathsf{KeyDec}(pk_s, sk_r, C)$:

1. Compute $t_1 = G(\sigma_1, pk_s, pk_r)$, $t_2 = H(\sigma_1, \sigma_2, pk_s, pk_r)$.
2. If
$$e(g, \sigma_3) = e(f, pk_s) \cdot e\left(\sigma_2, u' \prod_{i \in \mathcal{T}} u_i\right) \cdot e\left(\sigma_1, v^{t_2} w\right),$$
return
$$K = e(\sigma_1, h^{x_r});$$
otherwise return $\perp$.

It can be verified easily that the construction satisfies the correctness. Note that unlike previous SC-KEM constructions, the validity of the ciphertext of ours can get checked by anyone who only knows the public keys.

4 Comparisons

Prior to proving the security of our construction, we compare in this section our SC-KEM with the SC-KEM schemes in the literature [9,10,23,18]. Table 1 shows the comparisons.

The schemes in [18] and [23] need strongly unforgeable signatures as building block. Without random oracles, several signature schemes can be shown to be strongly unforgeable under relatively strong or standard assumptions.

Gennaro, Halevi, and Rabin [13], and Cramer and Shoup [7] constructed strongly unforgeable signatures based on the Strong-RSA assumption, and the signatures are composed of one element in $\mathbb{Z}_n$ (n is the RSA modulus) [13], and one element in $\mathbb{Z}_n$ and two elements in a group $\mathbb{G}$ (n is the RSA modulus, $|\mathbb{G}| \geq 160$) [7], respectively; Boneh and Boyen [3] constructed a strongly unforgeable signature based on the Strong-Diffie-Hellman assumption, and the signature consists of an element in $\mathbb{G}$ and an element in $\mathbb{Z}_p$; Boneh, Shen, and Waters [5] constructed a strongly unforgeable signature based on the standard computational Diffie-Hellman assumption, and the signature contains two elements in $\mathbb{G}$ and an element in $\mathbb{Z}_p$; Kang et al. [15] constructed a short signature scheme based on the computational Diffie-Hellman assumption, and the signature is composed of an element in $\mathbb{G}$ and an element in $\mathbb{Z}_p$.

We also notice that there are several generic transformations proposed to convert weak unforgeability into strong unforgeability. According to the shortest generic transformation [19] so far, in terms of signature size expansion, the transformation increases the resulting signature by one group element. For example, when we use the transformation to convert the Waters signature, the only known signature secure under standard CDH assumption without random oracles, into a strongly unforgeable one, the resultant signature contains three elements in $\mathbb{G}$.

The idea behind our SC-KEM construction is to combine the technique in the Waters signatures [24] with that in transforming identity-based encryption to CCA-secure public key encryption [6,17]. Different from the previous SC-KEM methods, we present a direct construction which does not rely on the use of any strongly unforgeable signature scheme. All known constructions for

Table 1. Comparisons of SC-KEM Schemes

	S/R KG	Size	ROM/Standard model	SA	PV	FIS
Dent[9]	1A	$1\|\mathbb{G}\|$	ROM	CDH	NO	NO
Dent[10]	1A	$2\|\mathbb{Z}_q\|$	ROM	GDH	NO	NO
Tan[23]	2A	$3\|\mathbb{G}\| + \|\mathsf{sig}\|$	Standard	DDH+SUF	NO	YES
Li[18]	2A	$2\|\mathbb{G}\| + \|\mathsf{sig}\|$	Standard	GHDH +SUF	NO	YES
Ours	1A	$3\|\mathbb{G}\|$	Standard	DBDH+CDH	YES	YES

S/R KG: Sender/Receiver Key Generation Algorithms; Size: Ciphertext Expansion Size; 1A: the key generation algorithm generates the public/private key pairs for both the sender and the receiver; 2A: two separate key generation algorithms are required, one for the sender, another for the receiver; $|\mathbb{G}|$, $|\mathsf{sig}|$: the bit lengths of the representation for elements in the underlying group $\mathbb{G}$, and for the signature generated by the underlying signature scheme, respectively (and $|\mathsf{sig}| \geq |\mathbb{G}| + |\mathbb{Z}_p|$ according to the state of the art on standard model based signatures); SA: Security Assumption; PV: Public Verifiability; FIS: Full Insider Security; GDH: Gap Diffie-Hellman; SUF: Strong Unforgeability of the underlying signature scheme; GHDH: Gap Hashed Diffie-Hellman.

SC-KEM schemes in the standard model are 'generic': they involve running a standard strongly unforgeable signature scheme and are thus not very efficient in ciphertext expansion as well as the computational performance.

5 Proving The Security

5.1 Confidentiality

Theorem 1. *If there exists an adversary $\mathcal{A}$ that can (t, q_d, ε)-break the IND-CCA2 security of our SC-KEM (q_d is the total number of the key decapsulation queries), then one can construct an algorithm $\mathcal{B}$ that (t', ε')-breaks the DBDH problem assuming that H is (t, ε_{cr})-collision resistant, where T_e, T_p are the running-time of the exponentiation in $\mathbb{G}$ and the pairing respectively, and*

$$\varepsilon' \geq \frac{\varepsilon}{2} - \varepsilon_{cr} - \frac{q_d}{p}, t' \leq t + \mathcal{O}(6 \cdot q_d + n + 12)T_e + \mathcal{O}(6 \cdot q_d)T_p, \quad (1)$$

Proof. Our idea of the proof is to utilize the adversary $\mathcal{A}$ that (t, q_d, ε)-breaks the IND-CCA2 security of our signcryption KEM, to construct an algorithm $\mathcal{B}$ that first simulates the environment of the $\mathsf{IND\text{-}CCA2}$ game, and then uses the output of $\mathcal{A}$ to solve the DBDH problem.

Assume that algorithm $\mathcal{B}$ is given as input a random 5 tuple (g, g^a, g^b, g^c, Z) where $Z = e(g,g)^{abc}$ or $e(g,g)^z$ for a, b, c, z randomly chosen from $\mathbb{Z}_p$. Algorithm $\mathcal{B}$'s goal is to output 1 if $Z = e(g,g)^{abc}$ and 0 otherwise. $\mathcal{B}$ does the following to achieve the goal.

Setup. $\mathcal{B}$ randomly chooses α_0, α_1, α_2, $\ldots$, α_n, α_v, α_w, β_v, s, γ and x_s from $\mathbb{Z}_p$, then sets

$$\sigma_2^\star = g^s, u' = g^{\alpha_0}, u_1 = g^{\alpha_1}, u_2 = g^{\alpha_2}, \ldots, u_n = g^{\alpha_n}, h = g^b, f = g^\gamma, v = g^{\alpha_v} h^{\beta_v},$$

$$t_2^\star = H(g^c, g^s, pk_s^\star, pk_r^\star), w = g^{\alpha_w} h^{-\beta_v t_2^\star}, pk_s^\star = g^{x_s}, sk_s^\star = x_s, pk_r^\star = g^a.$$

Finally $\mathcal{B}$ gives $\mathcal{A}$ the parameters u', u_1, u_2, ..., u_n, f, h, v, w and keys $pk_s^\star$, $sk_s^\star$, $pk_r^\star$.

All the parameters and keys we give here have the same distribution as those used in our construction. Thus, $\mathcal{B}$ provides a perfect simulation in this phase.

Phase 1. When $\mathcal{A}$ submits a query $(pk_s, C = (\sigma_1, \sigma_2, \sigma_3))$ for key decapsulation, $\mathcal{B}$ responds as follows:

1. Compute $t_1 = G(\sigma_1, pk_s, pk_r^\star)$, $t_2 = H(\sigma_1, \sigma_2, pk_s, pk_r^\star)$.
2. Check
$$e(g, \sigma_3) \overset{?}{=} e(f, pk_s) \cdot e\left(\sigma_2, u' \prod_{i \in \mathcal{T}} u_i\right) \cdot e(\sigma_1, v^{t_2} w), \tag{2}$$
if not, return $\perp$.
3. If $t_2 = t_2^\star$, abort (this event is denoted as CRFail); otherwise randomly choose r from $\mathbb{Z}_p$ and compute
$$\begin{aligned} D_1 &= (g^a)^{-\frac{\alpha_v \cdot t_2 + \alpha_w}{\beta_v (t_2 - t_2^\star)}} \cdot (v^{t_2} w)^r = h^a \cdot (h^a)^{-\frac{\beta_v(t_2 - t_2^\star)}{\beta_v(t_2 - t_2^\star)}} \cdot (g^a)^{-\frac{\alpha_v \cdot t_2 + \alpha_w}{\beta_v (t_2 - t_2^\star)}} \cdot (v^{t_2} w)^r \\ &= h^a \cdot \left(g^{\alpha_v \cdot t_2 + \alpha_w} h^{\beta_v (t_2 - t_2^\star)}\right)^{-\frac{a}{\beta_v(t_2 - t_2^\star)}} \cdot (v^{t_2} w)^r \\ &= h^a \cdot (v^{t_2} w)^{-\frac{a}{\beta_v(t_2 - t_2^\star)}} \cdot (v^{t_2} w)^r = h^a \cdot (v^{t_2} w)^{r - \frac{a}{\beta_v(t_2 - t_2^\star)}}, \\ D_2 &= g^r \cdot (g^a)^{-\frac{1}{\beta_v(t_2 - t_2^\star)}} = g^{r - \frac{a}{\beta_v(t_2 - t_2^\star)}}. \end{aligned}$$
Let $\eta = r - \frac{a}{\beta_v(t_2 - t_2^\star)}$, we have $D_1 = h^a \cdot (v^{t_2} w)^\eta$, $\quad D_2 = g^\eta$.
4. Compute
$$\varDelta = \sigma_3 \cdot (pk_s)^{-\gamma} \cdot (\sigma_2)^{-\alpha_0 - \sum_{i \in \mathcal{T}} \alpha_i}. \tag{3}$$
Since $C = (\sigma_1, \sigma_2, \sigma_3)$ can pass the verification equation (2), we have
$$pk_s = g^x,\ \sigma_1 = g^k,\ \sigma_2 = g^\ell,\ \sigma_3 = f^x \cdot \left(u' \prod_{i \in \mathcal{T}} u_i\right)^\ell (v^{t_2} w)^k,$$
for some $x, k, \ell \in \mathbb{Z}_p$. Thus, we know that
$$\begin{aligned} \varDelta &= \sigma_3 \cdot (pk_s)^{-\gamma} \cdot (\sigma_2)^{-\alpha_0 - \sum_{i \in \mathcal{T}} \alpha_i} = \sigma_3 \cdot (g^x)^{-\gamma} \cdot \left(g^\ell\right)^{-\alpha_0 - \sum_{i \in \mathcal{T}} \alpha_i} \\ &= \sigma_3 \cdot (g^\gamma)^{-x} \cdot \left(u' \prod_{i \in \mathcal{T}} u_i\right)^{-\ell} \\ &= f^x \cdot \left(u' \prod_{i \in \mathcal{T}} u_i\right)^\ell (v^{t_2} w)^k \cdot (f)^{-x} \cdot \left(u' \prod_{i \in \mathcal{T}} u_i\right)^{-\ell} = (v^{t_2} w)^k. \end{aligned}$$
5. Return
$$K = \frac{e(\sigma_1, D_1)}{e(D_2, \varDelta)}.$$
Note that K is correct because
$$\begin{aligned} e(\sigma_1, D_1) &= e\left(\sigma_1, h^a \cdot (v^{t_2} w)^\eta\right) = e\left(\sigma_1, h^a\right) \cdot e\left(g^k, (v^{t_2} w)^\eta\right) \\ &= K \cdot e\left(g^\eta, (v^{t_2} w)^k\right) = K \cdot e(D_2, \varDelta). \end{aligned} \tag{4}$$

Challenge. In this phase, $\mathcal{B}$ generates the challenge ciphertext for the adversary $\mathcal{A}$ as follows.

1. Set $\sigma_1^\star = g^c$ and compute $t_1^\star = G(\sigma_1^\star, pk_s^\star, pk_r^\star)$;
2. Compute $\sigma_3^\star = (g^\gamma)^{x_s} \cdot \left(u' \prod_{i \in \mathcal{T}^\star} u_i\right)^s \cdot (g^c)^{\alpha_v \cdot t_2^\star + \alpha_w}$;
3. Set $K_0^\star = Z$, $C^\star = (\sigma_1^\star, \sigma_2^\star, \sigma_3^\star)$;
4. Choose a random bit $\theta \in \{0,1\}$ and a random key $K_1^\star \in \mathbb{G}_T$;
5. Return $(K_\theta^\star, C^\star)$ as the challenge.

The ciphertext $C^\star$ is valid and can pass the Equation (2) since

$$\begin{aligned}\sigma_3^\star &= (g^\gamma)^{x_s} \cdot \left(u' \prod_{i \in \mathcal{T}^\star} u_i\right)^s \cdot (g^c)^{\alpha_v \cdot t_2^\star + \alpha_w} = f^{x_s} \cdot \left(u' \prod_{i \in \mathcal{T}^\star} u_i\right)^s \cdot (g^{\alpha_v \cdot t_2^\star + \alpha_w})^c \\ &= f^{x_s} \cdot \left(u' \prod_{i \in \mathcal{T}^\star} u_i\right)^s \cdot \left((g^{\alpha_v} \cdot h^{\beta_v})^{t_2^\star} \cdot (g^{\alpha_w} \cdot h^{-\beta_v t_2^\star})\right)^c \\ &= f^{x_s} \cdot \left(u' \prod_{i \in \mathcal{T}^\star} u_i\right)^s \cdot \left(v^{t_2^\star} w\right)^c.\end{aligned}$$

Phase 2. $\mathcal{B}$ responds to the queries of $\mathcal{A}$ as it does in Phase 1, except denying to answer the query of the challenge ciphertext $C^\star$ w.r.t. $pk_s^\star$.

Guess. Eventually $\mathcal{A}$ outputs a bit θ' as its guess for θ.

Algorithm $\mathcal{B}$ outputs 1 if $\theta' = \theta$ (denoted by ASuc), and 0 if $\theta' \neq \theta$.

Analysis. In the following, we analyze $\mathcal{B}$'s probability of success in solving the Decisional Bilinear Diffie-Hellman problem. We first present the following claim.

Claim. $\Pr[\mathsf{CRFail}] \leq \varepsilon_{cr} + \frac{q_d}{p}$, where q_d is the number of the key decapsulation queries made by $\mathcal{A}$.

Proof. For any valid ciphertext $C = (\sigma_1, \sigma_2, \sigma_3)$, event CRFail happens only when one of the following two events takes place:

1. Event CR, $(\sigma_1, \sigma_2, pk_s, pk_r^\star) \neq (\sigma_1^\star, \sigma_2^\star, pk_s^\star, pk_r^\star) \wedge t_2 = t_2^\star$;
2. Event Fail, $(\sigma_1, \sigma_2, pk_s, pk_r^\star) = (\sigma_1^\star, \sigma_2^\star, pk_s^\star, pk_r^\star)$.

Actually, event Fail can't happen in **Phase 2** because if $(\sigma_1, \sigma_2, pk_s, pk_r^\star) = (\sigma_1^\star, \sigma_2^\star, pk_s^\star, pk_r^\star)$ and $C = (\sigma_1, \sigma_2, \sigma_3)$ is valid (which can be verified by Equation (2)), then $\sigma_3 = \sigma_3^\star$ must hold. However, the challenge ciphertext $(\sigma_1^\star, \sigma_2^\star, \sigma_3^\star)$ with respect to $(pk_s^\star, pk_r^\star)$ is not allowed to be queried. Thus we know $(pk_s, C = (\sigma_1, \sigma_2, \sigma_3))$ can't be queried as well in **Phase 2.** Therefore, event Fail may happen in **Phase 1,** but must not happen in **Phase 2.**

The adversary cannot know the challenge ciphertext in Phase 1 because it is information-theoretically hidden in Phase 1. Then, the event $\mathcal{A}$ submits a ciphertext identical to the challenge one with the same sender's public key happens with probability at most $\frac{1}{p}$. And event Fail happens with probability at most $\frac{q_d}{p}$ for the q_d queries in **Phase 1**, i.e., $\Pr[\mathsf{Fail}] \leq \frac{q_d}{p}$.

Event CR, $(\sigma_1, \sigma_2, pk_s, pk_r^\star) \neq (\sigma_1^\star, \sigma_2^\star, pk_s^\star, pk_r^\star) \wedge t_2 = t_2^\star$, implies $\mathcal{B}$ finds a collision for H by utilizing $\mathcal{A}$. Therefore, $\Pr[\mathsf{CR}] \leq \varepsilon_{cr}$.

Thus, we know $\mathcal{B}$'s abortion probability is bounded by $\Pr[\mathsf{CRFail}] = \Pr[\mathsf{CR}] + \Pr[\mathsf{Fail}] \leq \varepsilon_{cr} + \frac{q_d}{p}$. □

Now we can compute the probability that $\mathcal{B}$ in the above game outputs 1 given Z with either $Z = e(g,g)^{abc}$ or $Z = e(g,g)^z$ where a, b, c, z are randomly chosen from $\mathbb{Z}_p$. Let ASuc be the event that the adversary $\mathcal{A}$ succeeds in guessing θ (i.e., $\theta' = \theta$).

Due to the simulation, it follows that if $Z = e(g,g)^{abc}$ then the challenge ciphertext $C^\star = (\sigma_1^\star, \sigma_2^\star, \sigma_3^\star)$ is a valid key encapsulation of $K_0^\star = Z$ under $(sk_s^\star, pk_r^\star)$. Therefore, $\mathcal{B}$ provides a perfect simulation unless event CRFail happens. Namely, $\mathcal{A}$'s view is identical to that in the real attack game unless event CRFail happens. So we have the following result.

$$\begin{aligned}&\Pr\left[\mathcal{B}(g^a, g^b, g^c, Z = e(g,g)^{abc}) = 1\right] = \Pr\left[(\mathsf{ASuc}|Z = e(g,g)^{abc}) \bigwedge (\neg\mathsf{CRFail})\right] \\ &\geq \Pr\left[\mathsf{ASuc}|Z = e(g,g)^{abc}\right] - \Pr\left[\mathsf{CRFail}\right] \geq \Pr\left[\theta = \theta'|Z = e(g,g)^{abc}\right] - \varepsilon_{cr} - \tfrac{q_d}{p} \\ &= \tfrac{\mathrm{Adv}_{\mathcal{A}}^{\mathsf{IND}}+1}{2} - \varepsilon_{cr} - \tfrac{q_d}{p} = \tfrac{\varepsilon+1}{2} - \varepsilon_{cr} - \tfrac{q_d}{p}.\end{aligned} \tag{5}$$

If $Z = e(g,g)^z$, then the challenge ciphertext $C^\star = (\sigma_1^\star, \sigma_2^\star, \sigma_3^\star)$ is an invalid key encapsulation of $K_0^\star = Z$ under $(sk_s^\star, pk_r^\star)$. In this case, both $K_0^\star = Z$ and $K_1^\star$ are random. Therefore, $\mathcal{A}$ succeeds in guessing θ with probability at most $\frac{1}{2}$. Thus, we have

$$\begin{aligned}&\Pr\left[\mathcal{B}(g^a, g^b, g^c, Z = e(g,g)^z) = 1\right] = \Pr\left[(\mathsf{ASuc}|Z = e(g,g)^z) \bigwedge (\neg\mathsf{CRFail})\right] \\ &\leq \Pr\left[\mathsf{ASuc}|Z = e(g,g)^z\right] = \Pr\left[\theta = \theta'|Z = e(g,g)^z\right] = \tfrac{1}{2}.\end{aligned} \tag{6}$$

Combining Equation (5) and Equation (6), we conclude that

$$\begin{aligned}\varepsilon' &= \mathrm{Adv}_{\mathcal{B}}^{\mathsf{DBDH}} = \left|\Pr[\mathcal{A}(g^a, g^b, g^c, e(g,g)^{abc}) = 1] - \Pr[\mathcal{A}(g^a, g^b, g^c, e(g,g)^z) = 1]\right| \\ &\geq \tfrac{\varepsilon+1}{2} - \varepsilon_{cr} - \tfrac{q_d}{p} - \tfrac{1}{2} = \tfrac{\varepsilon}{2} - \varepsilon_{cr} - \tfrac{q_d}{p}.\end{aligned}$$

Finally, for the running-time of $\mathcal{B}$, we mainly take into account the running-time t of $\mathcal{A}$, the exponentiations and the pairings in the key decapsulation queries, and the exponentiation of generating the parameters. This takes time at most $t + \mathcal{O}(6 \cdot q_d + n + 12)T_e + \mathcal{O}(6 \cdot q_d)T_p$, where T_e is the running-time of the exponentiation in $\mathbb{G}$, T_p is the running-time of the pairing, and q_d is the number of key decapsulation queries.

5.2 Unforgeability

Our signcryption KEM satisfies strong unforgeability as defined in definition 6. The following theorem formally proves its unforgeability. Note that we can conclude that the proposed construction is asymptotically unforgeable under the CDH assumption if the underlying hash function is collision resistant, as the Waters signature [24] itself can be reduced to the CDH assumption.

Theorem 2 (Unforgeability). *Our signcryption KEM is (t, q_s, ε)-strongly unforgeable assuming the Waters signature is $(t + \mathcal{O}(q_s), q_s, \varepsilon/2)$-existentially unforgeable, the CDH assumption $(t + \mathcal{O}(q_s), (\varepsilon - \varepsilon_{cr})/2q_s)$-holds in $\mathbb{G}$, and H is (t, ε_{cr})-collision resistant.*

Proof. The SUF game defines the strong unforgeability for signcryption KEM, and is played by an adversary and the challenger. Suppose there is an adversary $\mathcal{A}$ which can win the SUF game in time t with probability ε. $\mathcal{A}$ is first equipped with the public parameters and the keys $pk_s^\star, pk_r^\star, sk_r^\star$. $\mathcal{A}$ can make q_s key encapsulation queries and will be given $\Sigma = \{C_i = (\sigma_{i1}, \sigma_{i2}, \sigma_{i3}) | i = 1, 2, \ldots, q_s\}$ on these queries. Let $\Sigma_1 = \{\sigma_{i1} | i = 1, 2, \ldots, q_s\}$, and let $C^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*)$ be the forgery $\mathcal{A}$ eventually produces. As $C^* \notin \Sigma$, we can then distinguish between two types of forgeries:

Type I. A forgery where $\sigma_1^* \notin \Sigma_1$. In this case we denote the adversary as type I forger $\mathcal{A}_{\mathrm{I}}$.

Type II. A forgery where $\sigma_1^* = \sigma_{l1}$ and $\sigma_2^* \neq \sigma_{l2}$ for some $l \in \{1, 2, ..., q_s\}$. In this case we denote the adversary as type II forger $\mathcal{A}_{\mathrm{II}}$.

Note that if $\sigma_1^* = \sigma_{l1}$ and $\sigma_2^* = \sigma_{l2}$, then $\sigma_3^* = \sigma_{l3}$ because given $(pk_s^\star, pk_r^\star)$, σ_1^* and σ_2^* (resp., σ_{l1} and σ_{l2}) uniquely determines σ_3^* (resp., σ_{l3}) that implies $C^*(= C_l)$ is not a valid forgery.

A successful adversary $\mathcal{A}$ must output a forgery of either Type I or Type II. We will show that a Type I forger $\mathcal{A}_{\mathrm{I}}$ can be used to break the existential unforgeability of the Waters signature, and a Type II forger $\mathcal{A}_{\mathrm{II}}$ can be used to solve the CDH problem if H is collision resistant. The simulator can flip a coin at the beginning of the simulation to guess which type of forgery the adversary will produce and set up the simulation appropriately. In both cases the simulation is perfect. We start by describing how to use a Type II forgery which is the more interesting case.

Type II Forgery. Suppose $\mathcal{A}_{\mathrm{II}}$ is a Type II adversary which (t, q_s, ε)-breaks strong unforgeability of our signcryption KEM, producing a Type II forgery. We construct an adversary $\mathcal{B}_{\mathrm{II}}$ that can $(t, \frac{1}{q_s}(\varepsilon - \varepsilon_{cr}))$-break the Computational Diffie-Hellman problem if the hash function is (t, ε_{cr})-collision resistant.

Suppose $\mathcal{B}_{\mathrm{II}}$ is given (g, g^a, g^b) associated with the bilinear group parameters $\mathsf{pp} = (\mathbb{G}, \mathbb{G}_T, e, g)$ and its goal is to output g^{ab}. To utilize the forger $\mathcal{A}_{\mathrm{II}}$, the simulator $\mathcal{B}_{\mathrm{II}}$ simulates the environment of the SUF game.

Setup. $\mathcal{B}_{\mathrm{II}}$ generates the parameters, the public key of the sender, and the private/public key pair of the receiver.

1. Randomly choose α_0, α_1, …, α_n, α_v, α_w, x_s, s, γ and x_r from $\mathbb{Z}_p$.
2. Set

$$u' = g^{\alpha_0}, u_1 = g^{\alpha_1}, u_2 = g^{\alpha_2}, \ldots, u_n = g^{\alpha_n}, f = g^b, h = g^\gamma, v = g^{\alpha_v} f,$$

$$pk_r^\star = g^{x_r},\ sk_r^\star = x_r,\ pk_s^\star = g^{x_s},\ t_2^\star = H(g^a, g^s, pk_s^\star, pk_r^\star),\ w = g^{\alpha_w} f^{-t_2^\star}.$$

3. Give $\mathcal{A}_{\mathrm{II}}$ the parameters u', u_1, u_2, …, u_n, f, h, v, w and the keys $pk_s^\star$, $sk_r^\star$, $pk_r^\star$.

Encapsulation Queries. Suppose $\mathcal{A}_{\mathrm{II}}$ issues q_s key encapsulation queries. $\mathcal{B}_{\mathrm{II}}$ first picks up $j^\star \in \{1, 2, \ldots, q_s\}$ randomly, then responds to the i-th query as follows ($i = 1, 2, ..., q_s$):

1. If $i \neq j^\star$, select k, η randomly from $\mathbb{Z}_p$, and return $C_i = (\sigma_{i1}, \sigma_{i2}, \sigma_{i3})$ where $\sigma_{i1} = g^k$, $\sigma_{i2} = g^\eta$, $t_1 = G(g^k, pk_s^\star, pk_r^\star)$, $t_2 = H(g^k, g^\eta, pk_s^\star, pk_r^\star)$ and
$$\sigma_{i3} = (g^b)^{x_s} \cdot \left(u' \prod_{i \in \mathcal{T}} u_i\right)^\eta \cdot (v^{t_2} \cdot w)^k;$$
2. If $i = j^\star$, return $C_i = (\sigma_{j^\star 1}, \sigma_{j^\star 2}, \sigma_{j^\star 3})$ where $\sigma_{j^\star 1} = g^a$, $\sigma_{j^\star 2} = g^s$, $t_1^\star = G(g^a, pk_s^\star, pk_r^\star)$,
$$\sigma_{j^\star 3} = (g^b)^{x_s} \cdot \left(u' \prod_{i \in \mathcal{T}^\star} u_i\right)^s \cdot (g^a)^{\alpha_v t_2^\star + \alpha_w}.$$
3. Update $\Sigma = \Sigma \bigcup \{C_i\}$ (where we let Σ be initially empty).

Indeed, the ciphertext $C_{j^\star} = (\sigma_{j^\star 1}, \sigma_{j^\star 2}, \sigma_{j^\star 3})$ is valid because $\sigma_{j^\star 1} = g^a$, $\sigma_{j^\star 2} = g^s$,
$$\begin{aligned}\sigma_{j^\star 3} &= (g^b)^{x_s} \cdot \left(u' \prod_{i \in \mathcal{T}^\star} u_i\right)^s \cdot (g^a)^{\alpha_v t_2^\star + \alpha_w} \\ &= (g^b)^{x_s} \cdot \left(u' \prod_{i \in \mathcal{T}^\star} u_i\right)^s \cdot \left((g^{\alpha_v} f)^{t_2^\star} \cdot (g^{\alpha_w} f^{-t_2^\star})\right)^a \\ &= (g^b)^{x_s} \cdot \left(u' \prod_{i \in \mathcal{T}^\star} u_i\right)^s \cdot \left(v^{t_2^\star} \cdot w\right)^a.\end{aligned} \tag{7}$$

Output. In this phase, $\mathcal{A}_{\mathrm{II}}$ eventually outputs its forgery $C^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*)$ of Type II (implying $\sigma_1^* \in \Sigma_1$), $\mathcal{B}_{\mathrm{II}}$ does the following to extract g^{ab} for solving the CDH problem.

1. If $(\sigma_1^* = \sigma_{j^\star 1}$ and $\sigma_2^* \neq \sigma_{j^\star 2})$, compute $t_1^\star = G(\sigma_1^*, pk_s^\star, pk_r^\star)$, $t_2 = H(\sigma_1^*, \sigma_2^*, pk_s^\star, pk_r^\star)$.
2. If $t_2 = t_2^\star$, abort (we denote this as event ColF); otherwise compute
$$\Delta = \frac{\sigma_3^*}{(g^b)^{x_s} \cdot (\sigma_2^*)^{\alpha_0 + \sum_{i \in \mathcal{T}^\star} \alpha_i} \cdot (\sigma_1^*)^{\alpha_w + t_2 \alpha_v}}. \tag{8}$$
3. Return $(\Delta)^{\frac{1}{t_2 - t_2^\star}}$.

As $C^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*)$ is a valid forgery, we have, for some $\ell \in \mathbb{Z}_p$:
$$\sigma_1^* = g^a, \quad \sigma_2^* = g^\ell, \quad \sigma_3^* = (g^b)^{x_s} \cdot \left(u' \prod_{i \in \mathcal{T}^\star} u_i\right)^\ell \cdot (v^{t_2} w)^a,$$
$$\begin{aligned}\Delta &= \frac{\sigma_3^*}{(g^b)^{x_s} \cdot (\sigma_2^*)^{\alpha_0 + \sum_{i \in \mathcal{T}^\star} \alpha_i} \cdot (\sigma_1^*)^{\alpha_w + t_2 \alpha_v}} \\ &= \frac{(g^b)^{x_s} \cdot \left(u' \prod_{i \in \mathcal{T}^\star} u_i\right)^\ell \cdot (v^{t_2} w)^a}{(g^b)^{x_s} \cdot (\sigma_2^*)^{\alpha_0 + \sum_{i \in \mathcal{T}^\star} \alpha_i} \cdot (\sigma_1^*)^{\alpha_w + t_2 \alpha_v}} \\ &= \frac{(g^b)^{x_s} \cdot \left(g^{\alpha_0 + \sum_{i \in \mathcal{T}^\star} \alpha_i}\right)^\ell \cdot \left((g^{\alpha_v} \cdot f)^{t_2} \cdot (g^{\alpha_w} f^{-t_2^\star})\right)^a}{(g^b)^{x_s} \cdot (\sigma_2^*)^{\alpha_0 + \sum_{i \in \mathcal{T}^\star} \alpha_i} \cdot (\sigma_1^*)^{\alpha_w + t_2 \alpha_v}} \\ &= \frac{(g^{\alpha_v t_2 + \alpha_w} f^{t_2 - t_2^\star})^a}{(\sigma_1^*)^{\alpha_w + t_2 \alpha_v}} = \frac{(\sigma_1^*)^{\alpha_w + t_2 \alpha_v} \cdot (f^a)^{t_2 - t_2^\star}}{(\sigma_1^*)^{\alpha_w + t_2 \alpha_v}} = (g^{ab})^{t_2 - t_2^\star}.\end{aligned}$$

Thus, $\Delta^{\frac{1}{t_2 - t_2^\star}} = g^{ab}$. Namely, when $\mathcal{A}_{\rm II}$ outputs a valid forgery C^* of Type II (denoted as event ASuc), $\mathcal{B}_{\rm II}$ can successfully solve the CDH problem if $\sigma_1^* = \sigma_{j^\star 1}$ and event ColF doesn't happen.
Since $j^\star$ is information theoretically hidden from $\mathcal{A}_{\rm II}$, both event ASuc and event ColF are independent from event $\sigma_1^* = \sigma_{j^\star 1}$. Then we have $\Pr[\sigma_1^* = \sigma_{j^\star 1}] \geq \frac{1}{q_s}$, and

$$\begin{aligned}\Pr[g^{ab} \leftarrow \mathcal{B}(g, g^a, g^b)] &= \Pr[\mathsf{ASuc} \bigwedge \neg\mathsf{ColF} \bigwedge \sigma_1^* = \sigma_{j^\star 1}] \\ &= \Pr[\mathsf{ASuc} \bigwedge \neg\mathsf{ColF}] \cdot \Pr[\sigma_1^* = \sigma_{j^\star 1}] \\ &\geq \frac{\Pr[\mathsf{ASuc} \bigwedge \neg\mathsf{ColF}]}{q_s} \\ &\geq \frac{\Pr[\mathsf{ASuc}] - \Pr[\mathsf{ColF}]}{q_s} \\ &= \frac{\varepsilon - \Pr[\mathsf{ColF}]}{q_s}\end{aligned}$$

If event ColF happens, we get a collision of H. Thus $\Pr[\mathsf{ColF}] \leq \varepsilon_{cr}$. From Equation (5.2), we have

$$\Pr[g^{ab} \leftarrow \mathcal{B}(g, g^a, g^b)] \geq \frac{\varepsilon - \varepsilon_{cr}}{q_s}.$$

The running time of $\mathcal{B}_{\rm II}$ is close to that of $\mathcal{A}_{\rm II}$ except $(4q_s + 12) \cdot T_e$ in simulation where T_e is the running time of the exponentiation in $\mathbb{G}$.

Type I Forgery. Suppose $\mathcal{A}_{\rm I}$ is a Type I forger which (t, q_s, ε)-breaks the strong unforgeability of our signcryption KEM, producing a Type I forgery. We can construct an adversary $\mathcal{B}_{\rm I}$ that (t, ε)-breaks (existential unforgeability of) the Waters signature of the form $\left(g^r, g_2^\alpha \left(u' \prod_{i \in \mathcal{M}} u_i\right)^r\right)$. Refer to [24] for more details on the Waters signatures.

Suppose $\mathcal{B}_{\rm I}$ is given a public key $g_1 = g^a$ along with the parameters $\mathsf{pp} = (\mathbb{G}, \mathbb{G}_T, e, g, u', u_1, u_2, \ldots, u_n, g_2, G)$ and a signing oracle $\mathcal{O}_w$ that returns the Waters signatures on requested messages. Its goal is to output a Waters signature on some fresh message which is not among $\mathcal{B}_{\rm I}$'s chosen messages. To utilize $\mathcal{A}_{\rm I}$, the adversary $\mathcal{B}_{\rm I}$ simulates the environment of the SUF game.

Setup. In this phase, $\mathcal{B}_{\rm I}$ generates the remaining parameters and the public key of the sender and the private/public key pair of the receiver.

1. Randomly choose α_v, α_w, γ and x_r from $\mathbb{Z}_p$.
2. Set $f = g_2, h = g^\gamma, v = g^{\alpha_v}, w = g^{\alpha_w}, pk_r^\star = g^{x_r},\ sk_r^\star = x_r,\ pk_s^\star = g_1$.
3. Give $\mathcal{A}_{\rm I}$ the parameters u', u_1, $u_2, \ldots$, u_n, f, h, v, w and the keys $pk_s^\star$, $sk_r^\star$, $pk_r^\star$.

Encapsulation Queries. When $\mathcal{A}_{\rm I}$ makes key encapsulation queries, $\mathcal{B}_{\rm I}$ simulates the encapsulation oracle as follows:

1. Select k randomly from $\mathbb{Z}_p$, and compute $\sigma_1 = g^k$.
2. Submit $M = (g^k, pk_s^\star, pk_r^\star)$ to the oracle $\mathcal{O}_w$ and obtain the signature $(\sigma_{w1}, \sigma_{w2})$ on M.
3. Set $t_2 = H(g^k, \sigma_{w1}, pk_s^\star, pk_r^\star)$, $\sigma_1 = g^k$, $\sigma_2 = \sigma_{w1}$.
4. Return $C = (\sigma_1, \sigma_2, \sigma_3)$ where $\sigma_3 = \sigma_{w2} \cdot (\sigma_1)^{\alpha_v t_2 + \alpha_w} = \sigma_{w2} \cdot (v^{t_2} w)^k$.
5. Update $\mathbb{M} = \mathbb{M} \bigcup \{M\}$ (where we let $\mathbb{M}$ be initially empty).

Output. Eventually $\mathcal{A}_{\mathrm{I}}$ outputs its forgery $C^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*)$ of Type I (namely, σ_1^* is not included in any $(\sigma_1, \sigma_2, \sigma_3)$ returned by the encapsulation oracle), $\mathcal{B}_{\mathrm{I}}$ does the following to obtain a new forgery for the Waters signature:

1. Set $M^\star = (\sigma_1^*, pk_s^\star, pk_r^\star)$ and $t_2^* = H(\sigma_1^*, \sigma_2^*, pk_s^\star, pk_r^\star)$;
2. Compute $\sigma_{w2}^\star = \sigma_3^* \cdot (\sigma_1^*)^{-\alpha_v t_2^* - \alpha_w}$, and return $(M^\star, (\sigma_{w1}^\star = \sigma_2^*, \sigma_{w2}^\star))$.

Note that $M^\star \notin \mathbb{M}$ as σ_1^* is not included in any $(\sigma_1, \sigma_2, \sigma_3)$ returned by the encapsulation oracle. Meanwhile, $(\sigma_{w1}^\star = \sigma_2^*, \sigma_{w2}^\star)$ is a valid forgery of the Waters signature because, for $k = \log_g \sigma_1^*$ and $\ell = \log_g \sigma_2^*$, we have

$$\sigma_{w2}^\star = \sigma_3^* \cdot (\sigma_1^*)^{-\alpha_v t_2^* - \alpha_w} = \sigma_3^* \cdot (v^{t_2^*} w)^{-k} = g_2^a \cdot (u' \prod_{i \in \mathcal{T}^\star} u_i)^\ell,$$

where $\mathcal{T}^\star \subset \{1, 2, \ldots, n\}$ is the set of indices such that $G(M^\star)[i] = 1$, and $G(M^\star)[i]$ is the i-th bit of $G(M^\star)$.

The probability of $\mathcal{B}_{\mathrm{I}}$'s success in forging a Waters signature is the same as that of $\mathcal{A}_{\mathrm{I}}$'s success in outputting a forgery of Type I. The running times of $\mathcal{A}_{\mathrm{I}}$ and $\mathcal{B}_{\mathrm{I}}$ are almost the same except for $2q_s$ exponentiation computations in simulation. □

Acknowledgement. This work has been supported by the National Natural Science Foundation of China (Grant Nos. 61272536, 61172085, 61103221, 61021004, 61070249, U1135004, 61170080, 11061130539 and 60703031).

References

1. El Aimani, L.: Generic constructions for verifiable signcryption. In: Kim, H. (ed.) ICISC 2011. LNCS, vol. 7259, pp. 204–218. Springer, Heidelberg (2012)
2. An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 83–107. Springer, Heidelberg (2002)
3. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)
4. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. SIAM Journal on Computing 32(3), 586–615 (2003)
5. Boneh, D., Shen, E., Waters, B.: Strongly unforgeable signatures based on computational diffie-hellman. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 229–240. Springer, Heidelberg (2006)
6. Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: Proceedings of ACM CCS 2005, pp. 320–329 (2005)
7. Cramer, R., Shoup, V.: Signature schemes based on the strong RSA assumption. ACM TISSEC 3(3), 161–185 (2000)
8. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing 33(1), 167–226 (2004)

9. Dent, A.W.: Hybrid signcryption schemes with outsider security. In: Zhou, J., López, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, vol. 3650, pp. 203–217. Springer, Heidelberg (2005a)
10. Dent, A.W.: Hybrid signcryption schemes with insider security. In: Boyd, C., González Nieto, J.M. (eds.) ACISP 2005. LNCS, vol. 3574, pp. 253–266. Springer, Heidelberg (2005b)
11. Dent, A., Zheng, Y.: Practical signcryption. In: Information Security and Cryptography. Springer (2010)
12. Gamage, C., Leiwo, J., Zheng, Y.: Encrypted message authentication by firewalls. In: Imai, H., Zheng, Y. (eds.) PKC 1999. LNCS, vol. 1560, pp. 69–81. Springer, Heidelberg (1999)
13. Gennaro, R., Halevi, S., Rabin, T.: Secure hash-and-sign signatures without the random oracle. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 123–139. Springer, Heidelberg (1999)
14. Ji, P., Yang, M.: Verifiable short signcryption without random oracle. In: Wireless Communications, Networking and Mobile Computing, pp. 2270–2273 (2007)
15. Kang, L., Tang, X., Lu, X., Fan, J.: A short signature scheme in the standard model. IACR Eprint archive (2007), `http://eprint.iacr.org/2007/398`
16. Kiltz, E.: Chosen-ciphertext secure key-encapsulation based on gap hashed Diffie-Hellman. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 282–297. Springer, Heidelberg (2007)
17. Kiltz, E., Galindo, D.: Direct chosen-ciphertext secure identity-based key encapsulation without random oracles. In: Batten, L.M., Safavi-Naini, R. (eds.) ACISP 2006. LNCS, vol. 4058, pp. 336–347. Springer, Heidelberg (2006)
18. Li, F., Shirase, M., Takagi, T.: Efficient signcryption key encapsulation without random oracles. In: Yung, M., Liu, P., Lin, D. (eds.) Inscrypt 2008. LNCS, vol. 5487, pp. 47–59. Springer, Heidelberg (2009)
19. Liu, J.K., Au, M.H., Susilo, W., Zhou, J.: Short generic transformation to strongly unforgeable signature in the standard model. In: Gritzalis, D., Preneel, B., Theoharidou, M. (eds.) ESORICS 2010. LNCS, vol. 6345, pp. 168–181. Springer, Heidelberg (2010)
20. Matsuura, K., Zheng, Y., Imai, H.: Compact and flexible resolution of CBT multicast key-distribution. In: Masunaga, Y., Tsukamoto, M. (eds.) WWCA 1998. LNCS, vol. 1368, pp. 190–205. Springer, Heidelberg (1998)
21. Park, B., Lee, W.: ISMANET: a secure routing protocol using identity-based signcryption scheme for mobile ad-hoc networks. IEICE Transactions on Communications E88-B(6), 2548–2556 (2005)
22. Park, N., Moon, K., Chung, K.-I., Won, D.H., Zheng, Y.: A security acceleration using XML signcryption scheme in mobile grid web services. In: Lowe, D.G., Gaedke, M. (eds.) ICWE 2005. LNCS, vol. 3579, pp. 191–196. Springer, Heidelberg (2005)
23. Tan, C.: Insider-secure signcryption KEM/tag-KEM schemes without random oracles. In: Proceedings of International Conference on Availability, Reliability and Security - ARES 2008, pp. 1275–1281 (2008)
24. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
25. Zheng, Y.: Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost (signature)+ cost(encryption). In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 165–179. Springer, Heidelberg (1997)