Verifiably Encrypted Signatures with Short Keys based on the Decisional Linear Problem and Obfuscation for Encrypted VES

Ryo Nishimaki and Keita Xagawa

NTT Secure Platform Laboratories,
3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585 Japan,
{nishimaki.ryo, xagawa.keita}@lab.ntt.co.jp

Abstract. Verifiably encrypted signatures (VES) are signatures encrypted by a public key of a trusted third party and we can verify their validity without decryption. This paper proposes a new VES scheme which is secure under the decisional linear (DLIN) assumption in the standard model. We also propose new obfuscators for encrypted signatures (ES) and encrypted VES (EVES) which are secure under the DLIN assumption.

All previous efficient VES schemes in the standard model are either secure under standard assumptions (such as the computational Diffie-Hellman assumption) with large verification (or secret) keys or secure under *(non-standard) dynamic q-type assumptions* (such as the q-strong Diffie-Hellman extraction assumption) with short verification keys. Our construction is the first efficient VES scheme with short verification (and secret) keys secure under *a standard assumption (DLIN)*.

As by-products of our VES scheme, we construct new obfuscators for ES/EVES based on our new VES scheme. They are more efficient than previous obfuscators with respect to the public key size. Previous obfuscators for EVES are secure under non-standard assumption and use zero-knowledge (ZK) proof systems and Fiat-Shamir heuristics to obtain non-interactive ZK, i.e., its security is considered in the random oracle model. Thus, our construction also has an advantage with respect to assumptions and security models. Our new obfuscator for ES is obtained from our new obfuscator for EVES.

keywords: verifiably encrypted signature, obfuscation, encrypted verifiably encrypted signature, decisional linear assumption

1 Introduction

1.1 Background

In verifiably encrypted signature (VES) schemes, there are a signer, verifiers, and *a trusted third party, called the adjudicator*. The signer generates a signature, encrypts it under the public key of the adjudicator, and adds extra contents to make it verifiable without decryption. The adjudicator can recover ordinary signatures from encrypted ones by using his/her decryption key.

VES was introduced by Asokan, Shoup, and Waidner [2] and Boneh, Gentry, Lynn, and Shacham proposed an efficient (non-interactive) VES scheme based on Boneh-Lynn-Shacham signature scheme in the random oracle model (ROM) [8, 9]. VES has useful and important applications such as online contract signing and optimistic fair exchange [2, 3]. Suppose a situation that a user, say Alice, wants to buy digital goods from a company online. Alice gives the company her VES for a contract instead of paying money and the company returns the requested digital goods if it receive a valid VES. Alice sends an ordinary signature as effective one to the company if she receives the goods. If a malicious company does not return the requested goods when it receives a VES, Alice can claim that the VES is of no use for the contract since it is encrypted. If malicious Alice does not return a ordinary signature when she receives the goods, the company sends the encrypted signature together with the transcript to the adjudicator and the adjudicator extracts an ordinary signature from the VES by using the secret key of the adjudicator and returns it to the company. The adjudicator is offline, that is, it should be active only when malicious Alice cheats the company. As another application, Fuchsbauer used a certain kind of VES to construct delegatable anonymous credentials [16]. Anonymous credentials are very useful for access control [5]. In some system with access control, users must prove to have the required credential issued by an authority to use the system. The authority may want to delegate its right to other entities to avoid centralization of power.

Lu, Ostrovsky, Sahai, Shacham, and Waters proposed a VES scheme which is secure under the computational Diffie-Hellman (CDH) assumption in the standard model, but the verification key size is quite large [24]. Rückert and Schröder proposed a VES scheme with short verification keys, but its security relies on a non-standard q-type assumption, called q-strong DH extraction assumption [27]. They did not prove its hardness in the generic group model [28]. Thus, there is no VES scheme that achieves constant size verification key and signature based on standard assumptions.

Program Obfuscation and Encrypted Signature/VES. Encrypted VES (EVES) is an extension of encrypted signature (ES) proposed by Hada [23]. ES/EVES functionalities output encryption of signatures/VES. They do not encrypt messages but signatures, and can be used as building blocks of signcryption functionalities as Hada pointed out [23]. If Alice uses free web-mail services to send a mail to Bob on low computational power devices such as smart-phones and her web browsers do not have enough resources to sign messages and encrypt them with Bob's public key, then she wants web-mail providers to carry out its process instead of her. However, she does not want to reveal her signing key. The obfuscation for ES/EVES will give a solution. A program obfuscator is an algorithm which transforms a program into a completely unintelligible program whose functionality is the same as the original one [4, 22]. Informally speaking, obfuscators should guarantee that what is efficiently computed given an obfuscated program is nothing more than what is computed given *black-box access* to the original program. If Alice gives an obfuscated program for ES/EVES functionalities, then she can securely delegate her signing capability to web-mail providers. Moreover, in a situation that president Alice on vacation want to have vice president Carol sign contracts for Bob (only Alice to Bob) instead of her, Alice can give Carol an obfuscated program for EVES functionality. In the case of the obfuscator for ES by Hada, if a malicious party has access to Bob's decryption key, then Alice's signing key is extracted from the obfuscated program [23]. However, in the case of our obfuscator for EVES, even such a malicious party cannot extract Alice's key due to the existence of the adjudicator's key. Thus, obfuscators for EVES have useful applications.

Hada proposed a secure obfuscator for an ES functionality and its application to signcryption [23]. His scheme is secure under the DLIN assumption in the standard model, but the verification key size is quite large. Cheng, Zhang, and Zhang proposed a secure obfuscator for an EVES functionality at ProvSec'11 [13]. Their VES scheme and obfuscator for EVES use zero-knowledge (ZK) proofs and Fiat-Shamir heuristics to crash ZK proofs into non-interactive zero-knowledge (NIZK) proofs. That is, their scheme and obfuscator are *secure in the ROM*. Furthermore, they used a non-standard assumption, called exponent 3-weak DH assumption to prove the unforgeability of their scheme and did not prove opacity (explained in the next section), which is required for secure VES schemes, of their scheme.

In general, obfuscators for ES/EVES can be obtained from fully homomorphic encryption (FHE) schemes [17]. However, existing FHE schemes are still inefficient [11, 12, 14, 18–20, 29]. so we do not rely on expensive FHE schemes but directly construct obfuscators for ES/EVES.

1.2 Our Contributions and Constructions

We propose a new efficient VES scheme based on the *decisional linear assumption (DLIN)* in the standard model. Our main advantages over previous VES schemes are as follows:

1. It is efficient and secure under a standard (i.e., not q-type) assumption in the standard model.
2. The verification key and signature size is small (constant).

As a by-product of our VES scheme, we construct secure obfuscators for ES/EVES functionality based on the DLIN assumption in the standard model. Main advantages of our obfuscators for ES/EVES over previous obfuscators for ES/EVES are as follows: They are *secure under the DLIN assumption in the standard model with short verification keys.*

Comparison and Related Works. Comparisons of our results and previous results of VES schemes and obfuscators for ES/EVES are shown in Table 1 and in Table 2, respectively. Let λ denote the security parameter. In this paper, the CDH assumption is considered in bilinear groups. There is no efficient VES scheme and obfuscator for ES/EVES which are secure under standard assumptions in the standard model with short verification keys prior to our work. The VES scheme by Lu et al. needs a quite large verification key but its signature size is small and its security is based on a standard CDH assumption, so one may think that the scheme of Lu et al. is better than our scheme in terms of signature size. However, we think it is incomparable with our new scheme and we showed a tradeoff between the verification key size and signature size. Rückert proposed a VES scheme based on full-domain hash RSA signature, but it is secure in the ROM [25]. Rückert, Schneider, and Schröder proposed *generic* constructions for VES without NIZKs, pairings, and ROM. Their construction is very insightful, but their schemes use an extra adjudication setup phase and Merkle trees, so they need to setup large parameters and have large keys (non-constant size), that is, they are *inefficient* [26].

Our Construction Technique. Loosely speaking, a VES scheme consists of a signature scheme and a encryption scheme as Lu et al. and Rückert and Schröder [24, 27]. We use a signature scheme presented by Waters at CRYPTO'09 [31] as an underlying signature scheme. We call it the Waters dual signature in this paper to distinguish from Waters' signature at Eurocrypt'05 [30]. Someone may think that a combination of the Waters dual signature and ElGamal encryption easily yields a secure VES

Table 1. A summary of previous efficient schemes and ours for VES

Reference	Key size (vk/sk)	VES size	Assumptions	ROM
BGLS [8]	$1\mathbb{G}/1\mathbb{Z}_p$	$2\mathbb{G}$	CDH	Yes
ZSS [32]	$2\mathbb{G}/2\mathbb{Z}_p$	$1\mathbb{G}$	CDH	Yes
LOSSW [24]	$O(\lambda)\mathbb{G}(> 160\mathbb{G})/1\mathbb{Z}_p$	$3\mathbb{G}$	CDH	No
RS [27]	$4\mathbb{G}/2\mathbb{Z}_p$	$2\mathbb{G} + 1\mathbb{Z}_p$	q-strong DH extraction	No
This work	$16\mathbb{G} + 1\mathbb{G}_T/3\mathbb{G}$	$12\mathbb{G} + 2\mathbb{Z}_p$	*DLIN*	No

Table 2. A summary of previous obfuscation for encrypted ES/EVES

Reference	ES/EVES	Key size (vk)	ROM	Assumptions
Hada [23]	ES	$O(\lambda)$	No	DLIN
CZZ [13]	EVES	$O(\lambda)$	Yes	DLIN and Exponent 3-weak DH
This work	ES	$O(1)$	*No*	*DLIN*
This work	EVES	$O(1)$	*No*	*DLIN*

scheme under the DLIN assumption, *but that is not the case.* The reason is as follows: We can prove unforgeability of VES by relying on unforgeability of the underlying signature scheme as previous schemes [8,24,27], *but opacity is non-trivial.* Opacity means that it is difficult to extract ordinary signatures from VES, i.e., decrypt VES. Moreover, it is highly non-trivial whether we can prove opacity from standard assumptions or not. The reason is as follows: The VES scheme of Lu et al. is a combination of Waters' signature (Eurocrypt'05) [30] and the ElGamal encryption scheme and they proved its opacity from the aggregate extraction assumption [8] (fortunately, it is equivalent to the CDH assumption [15]). On the other hand, the VES scheme of Rückert and Schröder is a combination of Boneh-Boyen signature scheme [6] and the ElGamal encryption scheme, but they proved its opacity from q-strong DH extraction assumption, which is a *stronger assumption than that of underlying Boneh-Boyen signature scheme* [27].

Our construction is a combination of the Waters dual signature scheme and the ElGamal encryption scheme. We encrypt only signature elements *related to signing keys.* The security proof of the Waters dual signature is different from that of many known secure signature schemes such as Boneh-Boyen [6], Waters [30], so we must employ a different proof strategy from that of Lu et al. and Rückert and Schröder. The Waters dual signature has two types of signature, standard signature (which is called type A) and *semi-functional* signature (which is called type B). Semi-functional signatures also pass the verification algorithm as standard ones and are *indistinguishable from standard ones* [21,31]. We extend the proof strategy of this dual form signature technique to prove opacity. First, we employ type B signatures as normal signatures output by a normal signing algorithm and type A signatures are used for simulation. Both type A and B signatures are valid signatures and there is no essential difference in terms of functionality as long as a normal verification algorithm is used. We employ this swap of role since we do not know how to prove that the adversary cannot extract a valid type A signature from given VES when the oracle answers type A signatures.

In the experiment of opacity, the adversary can output a pair of a signature and a message such that the message was queried to an oracle which returns a VES for the queried message. This causes the main difficulty for proving the opacity since the adversary may output a re-randomized signature obtained by using valid signatures from oracles. Unfortunately, the Waters dual signature is re-randomizable. Thus, we modify the Waters dual signature scheme to make it *strongly* unforgeable. Strong unforgeability guarantees that the adversary cannot output a forgery even for a queried message, so it must hold that if the adversary output valid signature for queried message in the experiment of opacity, then the signature is identical to the signature generated by the VES creation oracle (otherwise, contradict to strong unforgeability). This fact can be used to prove the opacity of our scheme.

In the proof of opacity, we must simulate two oracles. One is the creation oracle, which answers VES for queried messages. The other is the adjudication oracle, which extracts ordinary signatures from queried message/VES pairs and returns them. When we answer only encryption of type B signature for VES creation queries of the adversary, we can prove that the adversary cannot extract type B signature from VES under the aggregate extraction assumption. This is the reason why we swap the role of type A signatures for that of type B signature. We have no way to prove that when we answer only encryption of type A signature for VES creation queries of the adversary, adversary cannot extracts type A signature from VES.

Thus, it is showed that the adversary cannot output a valid signature for *queried message* to the VES creation oracle. For non-queried messages, we can use the proof technique for unforgeability of dual form signatures. We show that the adversary cannot output a type A signature when the oracle returns type B signatures (VES).

Next, we change the type of signatures used to generate VES which are answered by the VES creation oracle. Answers of the adjudication oracle depend on the type of the VES creation oracle. Thus, we show that the view of the adversary is indistinguishable even if the type of answers are changed from type B to type A one-by-one for each query. This order of change is reverse to the original proof, but it is not essential difference. Lastly, we show that the adversary cannot output a type B signature when the oracle returns type A signatures (VES).

Secure obfuscations for ES and EVES based on the Waters dual signature scheme are also non-trivial because the signing keys of the Waters dual signature scheme consist of multiple group elements and the signing algorithm computes exponentiation of the signing keys with randomness in contrast to Waters' signature presented at Eurocrypt'05, whose signing key is only one group element and signing algorithm only multiplies it by other group elements [30]. We overcome this hurdle by using additive homomorphic property of ElGamal and the linear encryption schemes [7]. Cheng et al. use the linear encryption scheme for not only encryption of VES but also the construction of VES itself, so their VES scheme cannot check the validity of ciphertext by using only the pairing technique and they need (NI)ZK [13]. We do not need (NI)ZK because our new VES scheme uses the ElGamal encryption scheme and can verify the validity of VES by using only pairings.

Remark. In this extended abstract, we do not have enough space to write complete proofs and all definitions, so we omitted some of them.

2 Preliminaries

Notations and Conventions. For any $n \in \mathbb{N} \setminus \{0\}$, let $[n]$ be the set $\{1, \ldots, n\}$. When D is a random variable or distribution, $y \overset{\mathsf{R}}{\leftarrow} D$ denote that y is randomly selected from D according to its distribution. If S is a set, then $x \overset{\mathsf{U}}{\leftarrow} S$ denotes that x is uniformly selected from S. $y := z$ denotes that y is set, defined or substituted by z. When b is a fixed value, $A(x) \rightarrow b$ (e.g., $A(x) \rightarrow 1$) denotes the event that machine (or algorithm) A outputs a on input x. We say that function $f : \mathbb{N} \rightarrow \mathbb{R}$ is negligible in $\lambda \in \mathbb{N}$ if for every constant $c \in \mathbb{N}$ there exists $k_c \in \mathbb{N}$ such that $f(\lambda) < \lambda^{-c}$ for any $\lambda > k_c$. Hereafter, we use $f < \mathsf{negl}(\lambda)$ to mean that f is negligible in λ. Let $\mathit{\Gamma} := (p, \mathbb{G}, \mathbb{G}_T, e, g)$ be a description of groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p equipped with efficient bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. We often omit common parameters $\mathit{\Gamma}$. Let $\mathcal{G}_{\mathsf{bmp}}$ be a standard parameter generation algorithm for bilinear maps that outputs $\mathit{\Gamma}$.

Definition 1 (DLIN assumption). *The DLIN problem is to guess $\beta \in \{0, 1\}$, given $(\mathit{\Gamma}, g, f, \nu, g^x, f^y, Q_\beta) \overset{\mathsf{R}}{\leftarrow} \mathcal{G}_\beta^{\mathsf{dlin}}(1^\lambda)$, where $\mathcal{G}_\beta^{\mathsf{dlin}}(1^\lambda)$: $\mathit{\Gamma} :=$*

$(p, \mathbb{G}, \mathbb{G}_T, e, g) \overset{\mathsf{R}}{\leftarrow} \mathcal{G}_{\mathsf{bmp}}(1^\lambda)$, $f, \nu \overset{\mathsf{U}}{\leftarrow} \mathbb{G}$, $x, y \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$, $Q_0 := \nu^{x+y}$, $Q_1 \overset{\mathsf{U}}{\leftarrow} \mathbb{G}$, *return* $(\Gamma, g, f, \nu, g^x, f^y, Q_\beta)$. *The advantage is* $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DLIN}}(\lambda) :=$ $\left|\Pr\left[\mathcal{A}(\mathcal{I}) \to 1 \,\middle|\, \mathcal{I} \overset{\mathsf{R}}{\leftarrow} \mathcal{G}_0^{\mathsf{dlin}}(1^\lambda)\right] - \Pr\left[\mathcal{A}(\mathcal{I}) \to 1 \,\middle|\, \mathcal{I} \overset{\mathsf{R}}{\leftarrow} \mathcal{G}_1^{\mathsf{dlin}}(1^\lambda)\right]\right|$. *We say that the DLIN assumption holds if for all probabilistic polynomial-time (PPT) adversary* $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DLIN}}(\lambda) < \mathsf{negl}(\lambda)$.

Definition 2 (Aggregate Extraction (AgExt) assumption [8,15]). *The AgExt problem in bilinear groups is to compute* g^{xy}, *given* $\Gamma := (p, \mathbb{G}, \mathbb{G}_T, e, g) \overset{\mathsf{R}}{\leftarrow} \mathcal{G}_{\mathsf{bmp}}(1^\lambda)$ *and* $(g^x, g^y, g^\beta, g^\delta, g^{xy+\beta\delta})$ *for* $x, y, \beta, \delta \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$. *The advantage is* $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{AgExt}}(\lambda) := \Pr[z = g^{xy} \mid \Gamma \overset{\mathsf{U}}{\leftarrow} \mathcal{G}_{\mathsf{bmp}}(1^\lambda); x, y, \beta, \delta \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p; z \overset{\mathsf{R}}{\leftarrow} \mathcal{A}(\Gamma, g^x, g^y, g^\beta, g^\delta, g^{xy+\beta\delta})]$. *We say that the AgExt assumption holds in bilinear groups if for any PPT* $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{AgExt}}(\lambda) < \mathsf{negl}(\lambda)$.

Definition 3 (CDH assumption). *The CDH problem in bilinear groups is to compute* g^{xy}, *given* $\Gamma := (p, \mathbb{G}, \mathbb{G}_T, e, g) \overset{\mathsf{R}}{\leftarrow} \mathcal{G}_{\mathsf{bmp}}(1^\lambda)$ *and* (g^x, g^y) *for* $x, y \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$. *The advantage is* $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{CDH}}(\lambda) := \Pr[z = g^{xy} \mid \Gamma \overset{\mathsf{U}}{\leftarrow} \mathcal{G}_{\mathsf{bmp}}(1^\lambda); x, y \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p; z \overset{\mathsf{R}}{\leftarrow} \mathcal{A}(\Gamma, g^x, g^y)]$. *We say that the CDH assumption holds in bilinear groups if for any PPT* $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{CDH}}(\lambda) < \mathsf{negl}(\lambda)$.

The AgExt assumption is equivalent to computational Diffie-Hellman (CDH) assumption, which is implied by the DLIN assumption.

Theorem 1 ([15]). *The AgExt and CDH problems are Karp reducible to each other with* $O(1)$ *computation.*

Verifiably Encrypted Signature (VES). A VES scheme consists of following seven algorithms $\mathsf{VES} = \{\mathsf{AdjGen}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy}, \mathsf{Create}, \mathsf{VesVrfy}, \mathsf{Adj}\}$:

Adjudicator Key Generation: Algorithm AdjGen takes as input security parameter 1^λ and outputs a pair of key for an adjudicator, that is, $(apk, ask) \overset{\mathsf{R}}{\leftarrow} \mathsf{AdjGen}(1^\lambda)$.

Key Generation: Algorithm Gen takes as input 1^λ and outputs a pair of keys for a signer, that is, $(vk, sk) \overset{\mathsf{R}}{\leftarrow} \mathsf{Gen}(1^\lambda)$. They are called the verification key and the signing key, respectively.

Signing: Algorithm Sign takes as input a signing key and a message and outputs signature σ. That is, $\sigma \overset{\mathsf{R}}{\leftarrow} \mathsf{Sign}(sk, M)$, where $M \in \mathcal{M}_{vk}$ and $\mathcal{M}_{vk}$ is a message space defined by vk.

Verification: Algorithm Vrfy is deterministic and takes as input vk, M, and σ and outputs bit b. If $b = 1$ then the signature is valid. Else, it is invalid. That is, $\mathsf{Vrfy}(vk, \sigma, m) \to b$.

VES Creation: Algorithm Create takes as input sk, apk, and M and outputs VES ω on M. That is, $\omega \overset{\mathsf{R}}{\leftarrow} \mathsf{Create}(sk, apk, M)$.

VES Verification: Algorithm $\mathsf{VesVrfy}$ is deterministic and takes as input apk, vk, ω, and M and outputs bit b, $\mathsf{VesVrfy}(apk, vk, \omega, M) \rightarrow b$.

Adjudication: Algorithm Adj takes as input ask, apk, vk, ω, and M. If ω is valid, it extracts an ordinary signature σ on M and returns σ, that is $\sigma \overset{\mathsf{R}}{\leftarrow} \mathsf{Adj}(ask, apk, vk, \omega, M)$ if $\mathsf{VesVrfy}(apk, vk, \omega, M) \rightarrow 1$.

For correctnes, it is required that $\forall \lambda\ \forall (apk, ask) \overset{\mathsf{R}}{\leftarrow} \mathsf{AdjGen}(1^\lambda)\ \forall (vk, sk) \overset{\mathsf{R}}{\leftarrow} \mathsf{Gen}(1^\lambda)\ \forall m \in \mathcal{M}_{vk}$ $\mathsf{VesVrfy}(apk, pk, \mathsf{Create}(sk, apk, M), M) \rightarrow 1$ and $\mathsf{Vrfy}(vk, \mathsf{Adj}(ask, apk, vk, \mathsf{Create}(sk, apk, M)), M) \rightarrow 1$.

Experiments $\mathsf{VesForge}_\mathcal{A}(\lambda)$ and $\mathsf{Opac}_\mathcal{A}(\lambda)$ are defined as follows:

Experiment $\mathsf{VesForge}_\mathcal{A}(\lambda)$
$(apk, ask) \overset{\mathsf{R}}{\leftarrow} \mathsf{AdjGen}(1^\lambda)$;
$(vk, sk) \overset{\mathsf{R}}{\leftarrow} \mathsf{Gen}(1^\lambda)$;
$(M^*, \omega^*) \overset{\mathsf{R}}{\leftarrow}$
$\mathcal{A}^{\mathcal{CO}(sk,apk,\cdot),\mathcal{AO}(ask,apk,vk,\cdot,\cdot)}(vk, apk)$;
Return 1 iff
$\mathsf{VesVrfy}(apk, vk, \omega^*, M^*) \rightarrow 1$ and
$M^* \notin Q_\mathsf{C}$ and $M^* \notin Q_\mathsf{A}$.

Experiment $\mathsf{Opac}_\mathcal{A}(\lambda)$
$(apk, ask) \overset{\mathsf{R}}{\leftarrow} \mathsf{AdjGen}(1^\lambda)$;
$(vk, sk) \overset{\mathsf{R}}{\leftarrow} \mathsf{Gen}(1^\lambda)$;
$(M^*, \sigma^*) \overset{\mathsf{R}}{\leftarrow}$
$\mathcal{A}^{\mathcal{CO}(sk,apk,\cdot),\mathcal{AO}(ask,apk,vk,\cdot,\cdot)}(vk, apk)$;
Return 1 iff
$\mathsf{Vrfy}(vk, \sigma^*, M^*) \rightarrow 1$ and
$M^* \notin Q_\mathsf{A}$.

where the creation oracle, $\mathcal{CO}(sk, apk, \cdot)$, returns a VES for a queried message, the adjudication oracle, $\mathcal{AO}(ask, apk, vk, \cdot, \cdot)$, extracts and returns a signature for a queried message/VES pair, and Q_C and Q_A are sets of messages queried by the adversary to $\mathcal{CO}$ and $\mathcal{AO}$, respectively.

Definition 4 (Secure VES [8]). *A VES scheme is secure if it satisfies unforgeability and opacity, i.e., it holds for any PPT $\mathcal{A}$,* $\Pr[\mathsf{VesForge}_\mathcal{A}(\lambda) \rightarrow 1] < \mathsf{negl}(\lambda)$ *and* $\Pr[\mathsf{Opac}_\mathcal{A}(\lambda) \rightarrow 1] < \mathsf{negl}(\lambda)$.

Collision Resistant Hash Functions (CRHF). Let $\mathcal{H} := \{H_k\}$ be a keyed hash family of functions $H_k : \{0,1\}^* \rightarrow \{0,1\}^n$ indexed by $k \in \mathcal{K}_\lambda$ where λ is a security parameter.

Definition 5. *We say that $\mathcal{H}$ is (t, ϵ)-collision-resistant if for any adversary $\mathcal{A}$ running in time t, we have that* $\mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{A},\mathcal{H}}(\lambda) := \Pr[m_0 \neq m_1 \wedge H_k(m_0) = H_k(m_1) \mid (m_0, m_1) \overset{\mathsf{R}}{\leftarrow} \mathcal{A}(k)] < \epsilon$ *where the probability is taken over the random choice of $k \in \mathcal{K}_\lambda$ and random coins of $\mathcal{A}$.*

3 Strongly Unforgeable Waters Dual Signature

Waters Dual Signature Scheme. We review a signature scheme presented by Waters [31] since we use it as a essential building block. However, we add a few minor changes to fit the scheme to this paper. We will explain the differences between the original scheme and modified scheme WdSig.

$\mathsf{Wd.Gen}(1^\lambda, \Gamma)$: On input security parameter λ and $\Gamma := (p, \mathbb{G}, \mathbb{G}_T, e, g) \overset{\mathsf{R}}{\leftarrow} \mathcal{G}_{\mathsf{bmp}}(1^\lambda)$, it chooses generators $v, v_1, v_2, w, u, h \overset{\mathsf{U}}{\leftarrow} \mathbb{G}$ and exponent $a_1, a_2, b, \alpha \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$, computes $\tau_1 := vv_1^{a_1}, \tau_2 := vv_2^{a_2}$, and sets $VK := (\Gamma, g^b, g^{a_1}, g^{a_2}, g^{ba_1}, g^{ba_2}, v, v_1, v_2, \tau_1, \tau_2, \tau_1^b, \tau_2^b, w, u, h, e(g,g)^{\alpha a_1 b})$ and $SK := (VK, g^\alpha, g^{\alpha a_1}, g^{a_1 a_2})$. Hereafter we often omit input 1^λ.

$\mathsf{Wd.Sign}(SK, M)$: On input message $M \in \mathbb{Z}_p$, it selects $r_1, r_2, z_1, z_2, \gamma, \mathsf{stag} \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$, sets $r := r_1 + r_2$, computes $\mathsf{sig} := (\sigma_0, \sigma_1, \ldots, \sigma_7, \mathsf{stag})$, where

$$\begin{array}{lll} \sigma_0 := (u^M w^{\mathsf{stag}} h)^{r_1}, & \sigma_1 := g^{\alpha a_1} v^r g^{-a_1 a_2 \gamma}, & \sigma_2 := g^{-\alpha} v_1^r g^{z_1} g^{a_2 \gamma}, \\ \sigma_3 := (g^b)^{-z_1}, & \sigma_4 := v_2^r g^{z_2} g^{a_1 \gamma}, & \sigma_5 := (g^b)^{-z_2}, \\ \sigma_6 := (g^b)^{r_2}, & \sigma_7 := g^{r_1}. & \end{array}$$

$\mathsf{Wd.Vrfy}(VK, \mathsf{sig}, M)$: On input VK, M, and sig, it outputs 1 if and only if it holds that

$$\begin{aligned} e(u^M w^{\mathsf{stag}} h, \sigma_7) &= e(g, \sigma_0), \\ e(g^b, \sigma_1) e(g^{ba_1}, \sigma_2) e(g^{a_1}, \sigma_3) &= e(\tau_1, \sigma_6)\, e(\tau_1^b, \sigma_7), \\ e(g^b, \sigma_1)\, e(g^{ba_2}, \sigma_4)\, e(g^{a_2}, \sigma_5) &= e(\tau_2, \sigma_6)\, e(\tau_2^b, \sigma_7)\, e(g,g)^{\alpha a_1 b}. \end{aligned}$$

The differences are as follows: In the original Waters dual signature scheme, (1) the verification equation is only one equation and probabilistic, (2) values v, v_1, v_2 are included in secret keys, (3) value $g^{a_1 a_2}$ is not included in the signing key, (4) the (normal) signing algorithm does not multiply $g^{-a_1 a_2 \gamma}$, $g^{a_2 \gamma}$, $g^{a_1 \gamma}$ in σ_1, σ_2, σ_4, respectively.

There are two types of signatures in the Waters dual signature scheme, type A (if $\gamma = 0$) and type B (if $\gamma \neq 0$) signatures. The modified three verification equations above are introduced by Abe et al. [1]. They proved that if a signature passes the equations, then the signature is either type A or B. The original equations use ciphertexts and the decryption procedure of the Waters dual encryption scheme, so it is probabilistic and has a semi-functional verification algorithm that uses semi-functional ciphertexts [31]. Type A signatures are signatures with $\gamma = 0$ and pass both the normal and semi-functional verification equations. Type B signatures are signatures with $\gamma \neq 0$ and cannot pass the semi-functional verification equations (Gerbush, Lewko, O'Neill, and Waters defined them as backdoor verification tests [21]). As long as the verification equations are normal, both type A and type B signatures are valid signatures and there is no essential difference. Thus, we employ type B signatures in the normal signing algorithm.

Even if v, v_1, v_2 are disclosed, we cannot compute v_2^b (and semi-functional ciphertexts of the dual system encryption of Waters [31]). Thus, we add (v, v_1, v_2) to the verification key and this does not affect its security since g^α and $g^{\alpha a_1}$ (and v_2^b) are kept secret and they are essential secret signing keys. This is observed by Abe et al. [1]. For the minor changed version above, the following theorem holds [1, 31].

Theorem 2. *If the DLIN assumption holds, then* $\mathsf{WdSig} := \mathsf{Wd}.\{\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy}\}$ *is existentially unforgeable against adaptive chosen message attacks (EUF-CMA).*

The original Waters dual signature is not strongly unforgeable since it is re-randomizable. "Strong" means that the adversary cannot forge a signature even for a queried message to the signing oracle. In order to make our VES scheme satisfy opacity, we modify the Waters dual signature. We extend the technique by Boneh, Shen, and Waters [10]. They introduced a property called 2-partitioned to convert unforgeable signature schemes into *strongly* unforgeable signature schemes. We extend 2-partitioned to 3-partitioned.

Definition 6. *A signature scheme is* 3-partitioned *if it satisfies the following two properties:*

- *The signing algorithm consists of three deterministic algorithms F_1, F_2, and F_3*
  1. *chooses random $R \in \mathcal{R}$ ($\mathcal{R}$ is a space for randomness),*
  2. *computes $\Sigma_1 := F_1(M, R, VK)$, $\Sigma_2 := F_2(R, VK)$, $\Sigma_3 := F_3(R, SK)$,*
  3. *and outputs signature $\sigma := (\Sigma_1, \Sigma_2, \Sigma_3)$.*
- *Given M and Σ_2 there is at most one (Σ_1, Σ_3) such that $(\Sigma_1, \Sigma_2, \Sigma_3)$ is a valid signature on M under VK.*

A 2-partitioned signature is $\sigma = (\Sigma_1', \Sigma_2')$ where $\Sigma_1' = F_1'(M, R, SK)$ and $\Sigma_2' = F_2'(R, SK)$ [10]. Value Σ_2' binds all randomness R, so M and R fully determine Σ_1'. For VES, signature elements related to the secret signing key (i.e., Σ_3) should be encrypted, so we cannot use such elements as inputs to hash functions (we will use hash functions to obtain strongly secure signature) and want to isolate the secret signing key from Σ_2'. Otherwise, encrypted signatures are not verifiable. If Σ_3 is not used as an input of hash function, then hash values are not changed even if Σ_3 is encrypted. This is the reason why we introduced 3-partitioned and Σ_1 and Σ_2 are independent of the secret signing key.

Let $\Pi := (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be an existentially unforgeable signature scheme. New signature scheme $\Pi' := (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Vrfy}')$ is as follows:

$\mathsf{Gen}'(1^\lambda)$**:** It generates $(VK, SK) \overset{\mathsf{R}}{\leftarrow} \mathsf{Gen}(1^\lambda)$, chooses $\bar{h} \overset{\mathsf{U}}{\leftarrow} \mathbb{G}$ and random hash key $k \in \mathcal{K}$, and sets $(VK', SK') := ((VK, \bar{h}, k), SK)$.

$\mathsf{Sign}'(SK', M)$**:** On input message $M \in \{0,1\}^\ell$, it chooses exponent $\varphi \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$ and randomness $R \in \mathcal{R}$, computes $\Sigma_2 := F_2(R, VK)$, $\vartheta := H_k(M \parallel \Sigma_2)$ (view ϑ as an element in $\mathbb{Z}_p$), $m := H_k(g^\vartheta \bar{h}^\varphi)$, $\Sigma_1 := F_1(m, R, VK)$ and $\Sigma_3 := F_3(R, SK)$, and outputs a signature $\mathsf{sig} := (\Sigma_1, \Sigma_2, \Sigma_3, \varphi)$.

$\mathsf{Vrfy}'(VK', \mathsf{sig}, M)$**:** On input VK', M, and signature $\mathsf{sig} = (\Sigma_1, \Sigma_2, \Sigma_3, \varphi)$, it computes $\vartheta' := H_k(M \parallel \Sigma_2)$ (view ϑ' as an element $\mathbb{Z}_p$), $m' := H_k(g^{\vartheta'} \bar{h}^\varphi)$, It outputs 1 if and only if $\mathsf{Vrfy}(VK, (\Sigma_1, \Sigma_2, \Sigma_3), m') \rightarrow 1$.

Theorem 3. *Signature scheme Π' is (t, q, ϵ)-strongly existentially unforgeable if Π is $(t, q, \epsilon/3)$-existentially unforgeable, the $(t, \epsilon/3)$-DL assumption holds in $\mathbb{G}$, and H is $(t, \epsilon/3)$-collision-resistant.*

This is easily proved by extending the proof of Boneh, Shen, and Waters [10]. The DL assumption means the discrete logarithm assumption. The essential point is that given message M and partial signature Σ_2, the randomness which is used to generate the whole signature is determined and there is at most one (Σ_1, Σ_3) such that $(\Sigma_1, \Sigma_2, \Sigma_3)$ is a valid signature on M under VK. Intuitively, in the construction of Π', we sign not only message M but also randomness R to bind the randomness and prevent re-randomization. Moreover, in order to prevent message m being determined by randomness R, new randomness φ is introduced and chameleon hash functions $(g^\vartheta \bar{h}^\varphi)$ are used. Value m will be signed.

Theorem 4. *The Waters dual signature is* 3*-partitioned.*

Proof. Let $\mathcal{R} := \{(r_1, r_2, z_1, z_2, \mathsf{stag}, \gamma) | \; r_1, r_2, z_1, z_2, \mathsf{stag}, \gamma \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p\}$, then functions F_1, F_2, and F_3 are defined as follows: $R \overset{\mathsf{R}}{\leftarrow} \mathcal{R}$, $F_1(M, R, VK) := \sigma_0 = (u^M w^{\mathsf{stag}} h)^{r_1}$, $F_2(R, VK) := (\sigma_3, \ldots, \sigma_7, \mathsf{stag}) = (g^{-bz_1}, v_2^r g^{z_2} \cdot g^{a_1\gamma}, g^{-bz_2}, g^{br_2}, g^{r_1}, \mathsf{stag})$, $F_3(R, SK) := (\sigma_1, \sigma_2) = (g^{\alpha a_1} v^r \cdot g^{-a_1 a_2 \gamma}, g^{-\alpha} v_1^r g^{z_1} \cdot g^{a_2\gamma})$ where $\gamma \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$ is chosen for type B signatures. If the signature is type A, then $\gamma := 0$. We can interpret $\sigma_3, \sigma_5, \sigma_6, \sigma_7$ (outputs of F_2) as $g^{-bz_1}, g^{-bz_2}, g^{br_2}, g^{r_1}$, respectively and it follows $\sigma_0 = (u^M w^{\mathsf{stag}} h)^{r_1}$ from the first verification equation, that is, the output of F_1 is fixed. If we interpret σ_4 as $v_2^r g^{z_2} \cdot g^{a_1\gamma}$, then by the second and third equations two unknowns σ_1 and σ_2 are fixed to $g^{\alpha a_1} v^r \cdot g^{-a_1 a_2 \gamma}$ and $g^{-\alpha} v_1^r g^{z_1}$, respectively, that is, the output of F_3 is fixed. Thus, if the output of F_2 and M are fixed, then the outputs of F_1 and F_3 are also fixed.

We can see that even if (σ_1, σ_2) is encrypted by the ElGamal encryption, hash value $\vartheta = H_k(M \parallel (\sigma_3, \ldots, \sigma_7, \mathsf{stag}))$ is not changed, so it can be fitted to VES schemes. Note that we assume that each element $g \in \mathbb{G}$ has a unique encoding. We can obtain strongly secure scheme sWdSig:

$\mathsf{sWd.Gen}(1^\lambda, \mathit{\Gamma})$**:** It generates $(VK', SK') \xleftarrow{\mathsf{R}} \mathsf{Wd.Gen}(1^\lambda, \mathit{\Gamma})$, chooses $\bar{h} \xleftarrow{\mathsf{U}} \mathbb{G}$ and random hash key $k \in \mathcal{K}$, and sets $(VK, SK) := ((VK', \bar{h}, k), SK')$.

$\mathsf{sWd.Sign}(SK, M)$**:** On input message $M \in \mathbb{Z}_p$, it selects $r_1, r_2, z_1, z_2, \gamma, \mathsf{stag}, \varphi \xleftarrow{\mathsf{U}} \mathbb{Z}_p$, sets $r := r_1 + r_2$, computes $\sigma_1 := g^{\alpha a_1} v^r \cdot g^{-a_1 a_2 \gamma}$, $\sigma_2 := g^{-\alpha} v_1^r g^{z_1} \cdot g^{a_2 \gamma}$, $\sigma_3 := (g^b)^{-z_1}$, $\sigma_4 := v_2^r g^{z_2} \cdot g^{a_1 \gamma}$, $\sigma_5 := (g^b)^{-z_2}$, $\sigma_6 := (g^b)^{r_2}$, $\sigma_7 := g^{r_1}$, $\vartheta := H_k(M \parallel \mathit{\Sigma}_2)$ where $\mathit{\Sigma}_2 = (\sigma_3, \ldots, \sigma_7, \mathsf{stag})$ and view ϑ as an element in $\mathbb{Z}_p$, $m := H_k(g^{\vartheta} \bar{h}^{\varphi})$, $\sigma_0 := (u^m w^{\mathsf{stag}} h)^{r_1}$, and outputs $\mathsf{sig} := (\sigma_0, \sigma_1, \ldots, \sigma_7, \mathsf{stag}, \varphi)$.

$\mathsf{sWd.Vrfy}(VK, \mathsf{sig}, M)$**:** On input VK, M, and signature $\mathsf{sig} = (\sigma_0, \sigma_1, \ldots, \sigma_7, \mathsf{stag}, \varphi)$, it computes $\vartheta' := H_k(M \parallel (\sigma_3, \ldots, \sigma_7, \mathsf{stag}))$, $m' := H_k(g^{\vartheta'} \bar{h}^{\varphi})$, and $\mathsf{Wd.Vrfy}(VK', \mathsf{sig}', m') \to b$ where $\mathsf{sig}' := (\sigma_0, \ldots, \sigma_7, \mathsf{stag})$, and outputs b.

Corollary 1. *The scheme above is strongly unforgeable against adaptive chosen message attacks if the DLIN assumption holds. In particular, for any PPT adversary $\mathcal{F}$ against* sWdSig *that makes at most q signing queries, there exists PPT algorithm $\mathcal{B}'$ for DLIN and $\mathcal{C}$ for CRHF,* $\mathsf{Adv}^{\mathsf{sEUF\text{-}CMA}}_{\mathcal{F},\mathsf{sWdSig}}(\lambda) \le \{(q+3)/3\}\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}'} + (1/3)\mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{C},\mathcal{H}}$ *where* $\mathsf{Adv}^{\mathsf{sEUF\text{-}CMA}}_{\mathcal{F},\mathsf{sWdSig}}(\lambda)$ *and* $\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{F}',\mathsf{WdSig}}(\lambda)$ *is the advantage of the adversary for* sWdSig.

Note that the DL assumption is implied by the DLIN assumption.

4 Construction of Our VES

We present our VES scheme, sWdVES, based on the strongly secure variant the Waters dual signature scheme in this section. The proposed scheme is basically the same as the strongly unforgeable Waters dual signature scheme in Section 3 except that we encrypt signature elements which include secret keys $(g^{\alpha}, g^{\alpha a_1}, g^{a_1 a_2})$ by the ElGamal encryption scheme. That is, in our creation algorithm, only σ_1 and σ_2 are encrypted. In order to verify encrypted signatures, we add extra elements and cancel out group elements which are generated by pairing computation of encrypted signatures in the verification equation. sWdVES is as follows:

$\mathsf{AdjGen}(1^\lambda)$**:** It selects $\beta \xleftarrow{\mathsf{U}} \mathbb{Z}_p$ and sets $apk := \zeta := g^{\beta}$ and $ask := \beta$.

$\mathsf{Gen}(1^\lambda)$**:** It generates $(VK', SK') := ((g, g^b, g^{a_1}, g^{a_2}, g^{ba_1}, g^{ba_2}, \tau_1, \tau_2, \tau_1^b, \tau_2^b, v, v_1, v_2, w, u, h, \bar{h}, k, e(g,g)^{\alpha a_1 b}), (g^\alpha, g^{\alpha a_1}, g^{a_1 a_2})) \xleftarrow{\mathsf{R}} \mathsf{sWd.Gen}(1^\lambda)$ and sets $vk := VK'$ and $sk := (VK', SK')$.

Sign **and** Vrfy**:** Same as $\mathsf{sWd}.\{\mathsf{Sign}, \mathsf{Vrfy}\}$ in Section 3, respectively.

$\mathsf{Create}(sk, apk, M)$**:** It generates $(\sigma_0, \ldots, \sigma_7, \mathsf{stag}, \varphi) \xleftarrow{\mathsf{R}} \mathsf{sWd.Sign}(SK', M)$, selects $\rho_1, \rho_2 \xleftarrow{\mathsf{U}} \mathbb{Z}_p$, outputs $\omega := (K_0, \ldots, K_7, K_1', K_2', \hat{K}_1, \hat{K}_2, \mathsf{stag}, \varphi)$, where $(K_0, K_3, \ldots, K_7) := (\sigma_0, \sigma_3, \ldots, \sigma_7)$ and

$$K_1 := \sigma_1 \cdot \zeta^{\rho_1}, \qquad K_1' := g^{\rho_1}, \qquad \hat{K}_1 := (g^b)^{\rho_1},$$
$$K_2 := \sigma_2 \cdot \zeta^{\rho_2}, \qquad K_2' := g^{\rho_2}, \qquad \hat{K}_2 := (g^{ba_1})^{\rho_2}.$$

$\mathsf{VesVrfy}(apk, vk, \omega, M)$**:** It parses $\omega = (K_0, \ldots, K_7, K_1', K_2', \hat{K}_1, \hat{K}_2, \mathsf{stag}, \varphi)$, and computes $\vartheta' := H_k(M \parallel (K_3, \ldots, K_7, \mathsf{stag}))$, $m' := H_k(g^{\vartheta'} \bar{h}^{\varphi})$, It outputs 1 if and only if it holds that

$$e(K_1', g^b) = e(g, \hat{K}_1) \ , \ e(K_2', g^{ba_1}) = e(g, \hat{K}_2)$$
$$e(u^{m'} w^{\mathsf{stag}} h, K_7) = e(g, K_0)$$
$$\frac{e(g^b, K_1)}{e(\zeta, \hat{K}_1)} \cdot \frac{e(g^{ba_1}, K_2)}{e(\zeta, \hat{K}_2)} \cdot e(g^{a_1}, K_3) = e(\tau_1, K_6)\, e(\tau_1^b, K_7)$$
$$\frac{e(g^b, K_1)}{e(\zeta, \hat{K}_1)} \cdot e(g^{ba_2}, K_4)\, e(g^{a_2}, K_5) = e(\tau_2, K_6)\, e(\tau_2^b, K_7)\, e(g,g)^{\alpha a_1 b}$$

$\mathsf{Adj}(ask, apk, vk, \omega, M)$**:** It parses $\omega = (K_0, \ldots, K_7, \mathsf{stag}, \varphi)$ and computes $\sigma_1 := K_1 \cdot (K_1')^{-\beta}$, $\sigma_2 := K_2 \cdot (K_2')^{-\beta}$, $\sigma_3 := K_3$, $\sigma_4 := K_4$, $\sigma_5 := K_5$, $\sigma_6 := K_6$, $\sigma_7 := K_7$, $\sigma_0 := K_0$. If $\mathsf{VesVrfy}(apk, vk, \omega, M) \to 1$, then it outputs $(\sigma_0, \ldots, \sigma_7, \mathsf{stag}, \varphi)$. These are valid signatures.

Intuitively, the scheme above is secure because underlying signature scheme is strongly unforgeable. The adversary has no choice but to decrypt valid VES given by oracles to output a valid signature, but it contradicts to the one-wayness of the ElGamal encryption scheme.

Rückert and Schröder defined key-independence and extractability of VES to prove unforgeability and collusion-resistance of VES in a modular way [26,27]. Key-independence means that a VES creation algorithm consists of a signature generation part and a transformation (into VES) part and they are independent. Extractability means that if VES ω is valid, then the adjudicator can extract a valid (ordinary) signature σ with except negligible probability. Collusion-resistance means that no adversary can forge VES even if the adjudicator is corrupted, i.e., adversary obtains the secret decryption key of the adjudicator. Rückert and Schröder showed the following theorem.

Theorem 5 ([27]). *Let* VES *be an extractable and key-independent verifiably encrypted signature scheme.* VES *is unforgeable if and only if the underlying signature scheme* Sig *is unforgeable.*

As a corollary, sWdVES is unforgeable under the DLIN assumption since we can easily show that our sWdVES based on sWdSig is key-independent and extractable though we omit proofs in this extended abstract.

Theorem 6. sWdVES *is opaque if the DLIN assumption holds and there exists CRHF.*

Proof. If adversary $\mathcal{A}$ outputs forgery $\sigma^* = (\sigma_0^*, \ldots, \sigma_7^*, \mathsf{stag}^*, \varphi^*)$ and M^* such that M^* is not queried to $\mathcal{AO}$, then it means that $\mathcal{A}$ breaks opacity of sWdVES. $\mathcal{A}$ directly forges a signature of underlying sWdSig or extracts a signature by breaking the one-wayness of the ElGamal encryption scheme. In order to prove opacity, we introduce the following games: Let $\mathsf{Game}\text{-}(i)$ denote a game where $\mathcal{CO}$ answers encryption of type A signatures for the first i ($i \in [q_{\mathsf{C}}]$ and q_{C} is the number of creation query by $\mathcal{A}$) queries and encryption of type B signatures for the remaining $(q_{\mathsf{C}} - i)$ queries and $\mathcal{AO}$ answers signatures extracted from queried VES for all q_{A} (the number of adjudication query) queries. Let $\mathsf{Adv}_i^{\mathsf{forge\text{-}A}}$ (resp. $\mathsf{Adv}_i^{\mathsf{forge\text{-}B}}$) denote the advantage of the adversary in $\mathsf{Game}\text{-}(i)$ for outputting type A (resp. B) forgery for a non-queried message (a message which is not queried to $\mathcal{CO}$). Let $\mathsf{Adv}_0^{\mathsf{extract\text{-}B}}$ denote the advantage of the adversary in $\mathsf{Game}\text{-}0$ for extracting a type B signature from a VES for a queried message (a message which is queried to $\mathcal{CO}$).

1. In $\mathsf{Game}\text{-}(0)$, $\mathcal{CO}$ returns encryption of type B signature and $\mathcal{AO}$ returns type B signature. First, we show Lemma 1: If $\mathcal{A}$ outputs a valid type B signature for message M_i which has been *already queried to* $\mathcal{CO}$, then we can construct algorithm $\mathcal{E}$ which solves the AgExt problem. Thus, in the remaining games, we only consider $\mathcal{A}$ which outputs forgery for message M^* such that $M^* \neq M_i$ for all $i \in [q]$. We can show that if $\mathcal{A}$ outputs forgery of type A signature, then we can construct algorithm $\mathcal{B}_1$ which solves the CDH problem.
2. Next, we consider $\mathsf{Game}\text{-}(i)$. We can show that if $\mathcal{A}$ detects the change from type B answer to type A answer by $\mathcal{CO}$, then we can construct algorithm $\mathcal{B}_2$ which solves the DLIN problem.
3. Last, we consider $\mathsf{Game}\text{-}(q_{\mathsf{C}})$, where all answers for VES queries of $\mathcal{A}$ are encryption of type A signature. We can show that if $\mathcal{A}$ outputs a forgery of type B signature, then we can construct algorithm $\mathcal{B}_3$ which solves the DLIN problem.

Thus, if the DLIN assumption holds, the signature scheme is opaque. The core part is Lemma 1. By statements described above except Lemma 1, we can show $\mathsf{Adv}^{\mathsf{Opac}}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{forge\text{-}A}}_{0} + \mathsf{Adv}^{\mathsf{extract\text{-}B}}_{0} + \mathsf{Adv}^{\mathsf{forge\text{-}B}}_{0} < \mathsf{Adv}^{\mathsf{extract\text{-}B}}_{0} + \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{F},\mathsf{WdSig}} < \mathsf{Adv}^{\mathsf{extract\text{-}B}}_{0} + (q_{\mathsf{C}}+2)\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}}$. By Lemma 1, we can show

$$\begin{aligned}\mathsf{Adv}^{\mathsf{extract\text{-}B}}_{0} &< q_{\mathsf{C}}\mathsf{Adv}^{\mathsf{AgExt}}_{\mathcal{E}} + \mathsf{Adv}^{\mathsf{sEUF\text{-}CMA}}_{\mathcal{F}',\mathsf{sWdSig}} + \mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{C}}\\ &< \frac{4q_{\mathsf{C}}+3}{3}\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}} + \frac{4}{3}\mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{C}}.\end{aligned}$$

Thus, it holds $\mathsf{Adv}^{\mathsf{Opac}}_{\mathcal{A}}(\lambda) < ((7q_{\mathsf{C}}+9)/3)\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}} + (4/3)\mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{C}}$.

Lemma 1. *If there exists adversary $\mathcal{A}$ that outputs a type B forgery for a queried message M_i in* Game-(0)*, then we can construct algorithm $\mathcal{E}$ that solves the AgExt problem.*

Proof of lemma. $\mathcal{E}$ is given instance $(\varGamma, g^x, g^y, g^\beta, g^\delta, g^{xy+\beta\delta})$ of the AgExt problem. $\mathcal{E}$ generates the verification key as follows: Chooses exponents $a_1, b, y_v, y_{v_1}, y_{v_2}, y_w, y_h, y_u, \eta \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$ and hash key $k \in \mathcal{K}$, computes $g := g$, $g^b := g^b$, $g^{a_1} := g^{a_1}$, $g^{a_2} := g^y$, $g^{ba_2} := (g^y)^b$, $g^{ba_1} := g^{ba_1}$, $v := g^{y_v}$, $v_1 := g^{y_{v_1}}$, $v_2 := g^{y_{v_2}}$, $w := g^{y_w}$, $u := g^{y_u}$, $h := g^{y_h}$, $\bar{h} := g^\eta$, $\zeta := g^\beta$, $e(g,g)^{\alpha a_1 b} := e(g^x, g^y)^{a_1 \cdot b}$ (it implicitly holds $\alpha = xy$ though $\mathcal{E}$ does not have α), $\tau_1 := vv_1^{a_1}$, τ_1^b, $\tau_2 := v(g^y)^{y_{v_2}}$, and τ_2^b, and sets $VK := (g, g^b, g^{a_1}, g^{a_2}, g^{ba_1}, g^{ba_2}, \tau_1, \tau_2, \tau_1^b, \tau_2^b, w, u, h, \bar{h}, k, e(g,g)^{\alpha a_1 b})$ and $apk := \zeta = g^\beta$. Note that $\mathcal{E}$ does not have $a_2 = y$ and $g^\alpha = g^{xy}$, so $\mathcal{E}$ cannot directly compute Type B signature.

Simulation of Creation Oracle: $\mathcal{E}$ initializes list $\mathsf{QList} := \emptyset$. $\mathcal{E}$ chooses random index $j \overset{\mathsf{U}}{\leftarrow} [q_{\mathsf{C}}]$, i.e., guesses which VES $\mathcal{A}$ selects and outputs its extraction. $\mathcal{E}$ outputs encryption of Type B signatures for i-th VES creation query M_i as follows: If $i \neq j$, then chooses $r_1, r_2, z_1, z_2, \gamma', \mathsf{stag}, \varphi_i, \rho_1, \rho_2 \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$, sets $r := r_1 + r_2$ (we want to set $\gamma := x + \gamma'$), computes $\sigma_{i,1} := (g^y)^{-\gamma' a_1} \cdot v^r = (g^{\alpha a_1} v^r) \cdot g^{-a_1 a_2 \gamma}$ (where $a_2 = y$ and $xy = \alpha$), $\sigma_{i,2} := (g^y)^{\gamma'} v_1^r g^{z_1} = (g^\alpha v_1^r g^{z_1}) \cdot g^{a_2\gamma}$, $K_3 := \sigma_{i,3} := (g^b)^{-z_1}$, $K_4 := \sigma_{i,4} := (g^x)^{a_1} g^{a_1\gamma'} v_2^r g^{z_2} = (v_2^r g^{z_2}) \cdot g^{a_1\gamma}$, $K_5 := \sigma_{i,5} := (g^b)^{-z_2}$, $K_6 := \sigma_{i,6} := g^{r_2 b}$, $K_7 := \sigma_{i,7} := g^{r_1}$, $\vartheta_i := H_k(M_i \parallel \varSigma_{i,2})$ where $\varSigma_{i,2} := (\sigma_{i,3}, \ldots, \sigma_{i,7})$, $m_i := H_k(g^{\vartheta_i}\bar{h}^{\varphi_i})$, $K_0 := \sigma_{i,0} := (u^{m_i} w^{\mathsf{stag}} h)^{r_1}$, $K_1 := \sigma_{i,1} \cdot \zeta^{\rho_1}$, $K_1' := g^{\rho_1}$, $\hat{K}_1 := (g^b)^{\rho_1}$, $K_2 := \sigma_{i,2} \cdot \zeta^{\rho_2}$, $K_2' := g^{\rho_2}$, $\hat{K}_2 := (g^{ba_1})^{\rho_2}$, stores $(M_i, \sigma_i, R_i := (r_1, r_2, z_1, z_2, \mathsf{stag}, \gamma_i := \gamma'))$ in QList where $\sigma_i := (\sigma_{i,0}, \ldots, \sigma_{i,7}, \mathsf{stag}, \varphi_i)$ and outputs $\omega := (K_0, \ldots, K_7, K_1', K_2', \hat{K}_1, \hat{K}_2, \mathsf{stag}, \varphi_i)$ for M_i. We can verify σ_i is a correct type B signature.

Embedding Instance: If $i = j$, then $\mathcal{E}$ chooses $r_1^*, r_2^*, z_1^*, z_2^*, \gamma^*, \mathsf{stag}^*, \varphi^*, \rho_1^*, \rho_2^* \overset{\mathsf{U}}{\leftarrow} \mathbb{Z}_p$, sets $r^* = r_1^* + r_2^*$, answers $K_1^* := (g^{xy+\beta\delta})^{a_1} \cdot v^{r^*} \cdot (g^y)^{-a_1\gamma^*} \zeta^{\rho_1^*} =$

$(g^{\alpha a_1} v^{r^*}) \cdot g^{-a_1 a_2 \gamma^*} \zeta^{\rho'_1}$ (where $a_2 = y, xy = \alpha, \rho'_1 := a_1\delta + \rho^*_1$), $K^{*\prime}_1 := (g^{\delta})^{a_1} g^{\rho^*_1} = g^{\rho'_1}$, $\hat{K}^*_1 := (g^{\delta})^{ba_1} g^{b\rho^*_1} = (g^{b})^{\rho'_1}$, $K^*_2 := (g^{xy+\beta\delta})^{-1} v_1^{r^*} g^{z^*_1} \cdot (g^{a_2})^{\gamma^*} (g^{\beta})^{\rho^*_2} = (g^{-\alpha} v_1^{r^*} g^{z^*_1}) \cdot g^{a_2\gamma^*} \zeta^{\rho'_2}$ (where $\rho'_2 := -\delta + \rho^*_2$), $K^{*\prime}_2 := (g^{\delta})^{-1} g^{\rho^*_2} = g^{\rho'_2}$, $\hat{K}^*_2 := (g^{\delta})^{-ba_1} g^{ba_1\rho^*_2} = (g^{ba_1})^{\rho'_2}$, $K^*_3 := (g^{b})^{-z^*_1}$, $K^*_4 := v_2^{r^*} g^{z^*_2} g^{a_1\gamma^*}$, $K^*_5 := (g^{b})^{-z^*_2}$, $K^*_6 := g^{r^*_2 b}$, $K^*_7 := g^{r^*_1}$, $\vartheta^* := H_k(K^*_3, \ldots, K^*_7, \mathsf{stag}^*)$, $m^* := H_k(g^{\vartheta^*} \bar{h}^{\varphi^*})$, and $K^*_0 := (u^{m^*} w^{\mathsf{stag}^*} h)^{r^*_1}$ and records $(M_j, \omega^* := (K^*_0, \ldots, \varphi^*), j, \gamma^*)$ as the challenge instance. It can be verified ω^* is a correct encryption of type B signature.

Simulation of Adjudication Oracle: When $\mathcal{A}$ makes ℓ-th adjudication query (M_ℓ, ω_ℓ), then we know that $\mathcal{A}$ must have queried M_ℓ to $\mathcal{CO}$ by the theorem of Rückert and Schröder (Otherwise, it is a forgery. This is the same argument by Rückert and Schröder in [27]). First, $\mathcal{E}$ verifies the query and returns $\perp$ if it is invalid. Otherwise, $\mathcal{E}$ acts as follows: If $M_\ell = M_j$, that is, the guessed index ($(M_j, \ldots) \notin \mathsf{QList}$), then $\mathcal{E}$ aborts. Otherwise, there exists $(M_i, \sigma_i, R_i) \in \mathsf{QList}$ for some $i \neq j$ such that $M_\ell = M_i$ and the signature is Type B. In this case ($M_\ell = M_i$), for query $(M_\ell, \omega = (K_0, \ldots, K_7, K'_1, K'_2, \hat{K}_1, \hat{K}_2, \mathsf{stag}, \varphi))$, if $\varphi \neq \varphi_i$, then $\mathcal{A}$ breaks strong unforgeability of our modified Waters dual signature. We consider an intermediate game where if $\varphi \neq \varphi_i$, then $\mathcal{E}$ aborts. The probability $\mathcal{E}$ aborts with this condition is less than the success probability of breaking strong unforgeability of sWdSig. That is, it holds $\varphi = \varphi_i$ without negligible probability. If $\varphi = \varphi_i$, then it holds $K_3 = \sigma_{i,3}$, $K_4 = \sigma_{i,4}$, $K_5 = \sigma_{i,5}$, $K_6 = \sigma_{i,6}$, $K_7 = \sigma_{i,7}$, $\mathsf{stag} = \mathsf{stag}_i$ since otherwise it means $\mathcal{A}$ outputs $(K_3, \ldots, K_7, \mathsf{stag})$ such that $H_k(\sigma_{i,3}, \ldots, \sigma_{i,7}, \mathsf{stag}_i) = \varphi_i = \varphi = H_k(K_3, \ldots, K_7, \mathsf{stag})$ and $(K_3, \ldots, K_7, \mathsf{stag}) \neq (\sigma_{i,3}, \ldots, \sigma_{i,7}, \mathsf{stag}_i)$. This is a collision of the hash function and contradicts to the collision-resistant property. We consider an intermediate game where if $H_k(\sigma_{i,3}, \ldots, \sigma_{i,7}, \mathsf{stag}_i) = \varphi_i = \varphi = H_k(K_3, \ldots, K_7, \mathsf{stag})$ and $(K_3, \ldots, K_7, \mathsf{stag}) \neq (\sigma_{i,3}, \ldots, \sigma_{i,7}, \mathsf{stag}_i)$, then $\mathcal{E}$ aborts. The probability $\mathcal{E}$ aborts with this condition is less than the success probability of breaking the CRHF. That is, randomness of $(K_3, \ldots, K_7, \mathsf{stag})$ is the same as that of $(\sigma_{i,3}, \ldots, \sigma_{i,7}, \mathsf{stag}_i)$ without negligible probability.

By using $K_3 = g^{-bz_1}$, $K_5 = g^{-bz_2}$, $K_6 = g^{br_2}$, $K_7 = g^{r_1}$, $\mathcal{E}$ can compute $g^{r_2} = (K_6)^{1/b}$, $g^{r_1} = K_7$, $g^{z_1} = (K_3)^{-1/b}$, $g^{z_2} = (K_5)^{-1/b}$, $v^r = (g^{r_1} \cdot g^{r_2})^{y_v}$, $v_1^r = (g^{r_1} \cdot g^{r_2})^{y_{v_1}}$, and $v_2^r = (g^{r_1} \cdot g^{r_2})^{y_{v_2}}$ since $\mathcal{E}$ has b, y_v, y_{v_1}, y_{v_2} and it holds that $v = g^{y_v}$ $v_1 = g^{y_{v_1}}$ $v_2 = g^{y_{v_2}}$. $\mathcal{E}$ can use the same computation procedure in the simulation of $\mathcal{CO}$ above by using γ_i stored in QList. Therefore, $\mathcal{E}$ can return valid Type B signature $(\sigma_0, \ldots, \sigma_7, \mathsf{stag}, \varphi)$ such that the randomness r in σ_1, σ_2 is the same as that in K_1, K_2 by using stored information σ_i. That is, $\mathcal{AO}$ is perfectly simulated by $\mathcal{E}$.

Solving the Problem: At some point, $\mathcal{A}$ outputs a Type B extraction, $M^* = M_j$, stag^*, $\sigma_1^* = g^{\alpha a_1} v^{r^*} g^{-a_1 a_2 \gamma^*}$, $\sigma_2^* = g^{-\alpha} v_1^{r^*} g^{z_1^*} g^{a_2 \gamma^*}$, $\sigma_3^* = (g^b)^{-z_1^*}$, $\sigma_4^* = v_2^{r^*} g^{z_2^*} g^{a_1 \gamma^*}$, $\sigma_5^* = (g^b)^{-z_2^*}$, $\sigma_6^* = g^{r_2^* b}$, $\sigma_7^* = g^{r_1^*}$, $\vartheta^* = H_k(\sigma_3, \ldots, \sigma_7, \mathsf{stag}^*)$, $m^* = H_k(g^{\vartheta^*} \bar{h}^{\varphi})$, and $\sigma_0^* = (u^{m^*} w^{\mathsf{stag}^*} h)^{r_1^*}$ (not queried to $\mathcal{AO}$ but $\mathcal{CO}$) such that *randomness are the same as those used when $\mathcal{B}$ embedded the problem instance at j-th query* and $(M_j, \omega^*, j, \gamma^*)$ is recorded as the challenge instance. This is guaranteed by the strong unforgeability and collision-resistance as we discussed above. By using these values, $\mathcal{E}$ can compute $g^{r_2^*} = (\sigma_6^*)^{1/b}$, $g^{r_1^*} = \sigma_7^*$, $g^{z_1^*} = (\sigma_3^*)^{-1/b}$, $g^{z_2^*} = (\sigma_5^*)^{-1/b}$, $v^{r^*} = (g^{r_1^*} \cdot g^{r_2^*})^{y_v}$, $v_1^{r^*} = (g^{r_1^*} \cdot g^{r_2^*})^{y_{v_1}}$, $v_2^{r^*} = (g^{r_1^*} \cdot g^{r_2^*})^{y_{v_2}}$, since $\mathcal{E}$ has b, y_v, y_{v_1}, y_{v_2} and it holds that $v = g^{y_v}$ $v_1 = g^{y_{v_1}}$ $v_2 = g^{y_{v_2}}$. Thus, $\mathcal{E}$ can compute $g^{z_1^*} \cdot v_1^{r^*} g^{a_2 \gamma^*} / \sigma_2^* = g^{\alpha} = g^{xy}$ since $\mathcal{E}$ has a_1 and γ^* is recorded. That is, $\mathcal{E}$ can output solution g^{xy} of the AgExt problem if the adversary outputs a Type A extraction for queried message M_j to $\mathcal{CO}$. $\mathcal{E}$ guesses index j, so its success probability is degraded by a factor of $1/q_{\mathsf{C}}$. However, it still breaks the AgExt problem with non-negligible probability ϵ/q_{C} where ϵ is the success probability of $\mathcal{A}$. ■

5 Application to Obfuscators for ES and EVES

Our VES scheme can be used to construct new obfuscators for ES and EVES. Hada constructed an obfuscator for ES by combining Waters's signature (2005) and the linear encryption scheme [23]. The linear encryption scheme proposed by Boneh, Boyen, and Shacham [7] and is as follows:

$\mathsf{L.Gen}(1^\lambda)$: It generates $\Gamma \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{bmp}}(1^\lambda)$, selects exponents $x_e, y_e \xleftarrow{\mathsf{U}} \mathbb{Z}_p$, and outputs $pk := (f_e, h_e) := (g^{x_e}, g^{y_e})$, $dk := (x_e, y_e)$.

$\mathsf{L.Enc}(pk_e, m)$: On input $m \in \mathbb{G}$ and $pk = (f_e, h_e)$ it selects $r, s \xleftarrow{\mathsf{U}} \mathbb{Z}_p$ and outputs $c := (f_e^r, h_e^s, g^{r+s} m)$.

Hada's idea is as follows: Suppose that signature σ is computed as $\sigma = sk \cdot G(m)$ where $sk \in \mathbb{G}$ is the signing key, $m \in \mathbb{Z}_p$ is the message and $G : \mathbb{Z}_p \to \mathbb{G}$ is an efficiently computable function. Then, for ciphertext $c = \mathsf{L.Enc}(pk, sk)$, we can compute $\tilde{c} := c \cdot G(m) = \mathsf{L.Enc}(pk, sk \cdot G(m))$ by homomorphic property of the linear encryption scheme. This is exactly an encrypted signature. The ciphertext of sk can be seen as an obfuscated circuit for encrypted signatures since the linear encryption scheme is semantically secure and no information about sk is revealed.

We extend Hada's construction, that is, we combine our VES scheme based on the strongly unforgeable Waters dual signature and the linear encryption scheme. However, our VES scheme is based on the Waters dual signature, which is more complex than Waters' signature at Eurocrypt'05, so it is non-trivial whether we can use Hada's technique directly or not. Especially, in Waters' signature at Eurocrypt'05, the signing algorithm does not exponentiate sk, but in the Waters dual signature, it does. We can resolve this problem by using the multiplicatively homomorphic property of the linear encryption scheme, that is, we can compute $c^r \cdot G(m) = \mathsf{L.Enc}(pk, sk^r \cdot G(m))$. Therefore, if we encrypt $sk = (g^{\alpha}, g^{\alpha a_1}, g^{a_1 a_2})$ by linear encryption, then we can construct an obfuscator for ES/EVES. We omit details of these constructions since we do not have space to present them. We will present them in a full version.

References

1. M. Abe, M. Chase, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Constant-size structure-preserving signatures: Generic constructions and simple assumptions. In *ASIACRYPT*, volume 7658 of *LNCS*, pages 4–24. Springer, 2012. full version available from http://eprint.iacr.org/2012/285.
2. N. Asokan, V. Shoup, and M. Waidner. Optimistic fair exchange of digital signatures (extended abstract). In *EUROCRYPT*, volume 1403 of *Lecture Notes in Computer Science*, pages 591–606. Springer, 1998.
3. F. Bao, R. H. Deng, and W. Mao. Efficient and practical fair exchange protocols with off-line TTP. In *IEEE Symposium on Security and Privacy*, pages 77–85. IEEE Computer Society, 1998.
4. B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6, 2012.
5. M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. Randomizable proofs and delegatable anonymous credentials. In *CRYPTO*, volume 5677 of *LNCS*, pages 108–125. Springer, 2009.
6. D. Boneh and X. Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *J. Cryptology*, 21(2):149–177, 2008.
7. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *CRYPTO'04*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004.
8. D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432. Springer, 2003.
9. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In *ASIACRYPT*, volume 2248 of *LNCS*, pages 514–532. Springer, 2001.
10. D. Boneh, E. Shen, and B. Waters. Strongly unforgeable signatures based on computational Diffie-Hellman. In *PKC'06*, volume 3958 of *LNCS*, pages 229–240, 2006.
11. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, pages 309–325. ACM, 2012.
12. Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS*, pages 97–106. IEEE, 2011.

13. R. Cheng, B. Zhang, and F. Zhang. Secure obfuscation of encrypted verifiable encrypted signatures. In *ProvSec*, volume 6980 of *LNCS*, pages 188–203. Springer, 2011.
14. J.-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 487–504. Springer, 2011.
15. J.-S. Coron and D. Naccache. Boneh et al.'s k-element aggregate extraction assumption is equivalent to the Diffie-Hellman assumption. In *ASIACRYPT*, volume 2894 of *LNCS*, pages 392–397. Springer, 2003.
16. G. Fuchsbauer. Commuting signatures and verifiable encryption. In *EUROCRYPT*, volume 6632 of *LNCS*, pages 224–245. Springer, 2011.
17. C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178. ACM, 2009.
18. C. Gentry and S. Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In *FOCS*, pages 107–109. IEEE, 2011.
19. C. Gentry and S. Halevi. Implementing Gentry's fully-homomorphic encryption scheme. In *EUROCRYPT*, volume 6632 of *LNCS*, pages 129–148. Springer, 2011.
20. C. Gentry, S. Halevi, and N. P. Smart. Fully homomorphic encryption with polylog overhead. In *EUROCRYPT*, volume 7237 of *LNCS*, pages 465–482. Springer, 2012.
21. M. Gerbush, A. B. Lewko, A. O'Neill, and B. Waters. Dual form signatures: An approach for proving security from static assumptions. In *ASIACRYPT*, volume 7658 of *LNCS*, pages 25–42. Springer, 2012.
22. S. Hada. Zero-knowledge and code obfuscation. In *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 443–457. Springer, 2000.
23. S. Hada. Secure obfuscation for encrypted signatures. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 92–112. Springer, 2010.
24. S. Lu, R. Ostrovsky, A. Sahai, H. Shacham, and B. Waters. Sequential aggregate signatures and multisignatures without random oracles. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 465–485. Springer, 2006.
25. M. Rückert. Verifiably encrypted signatures from RSA without NIZKs. In *INDOCRYPT*, volume 5922 of *LNCS*, pages 363–377. Springer, 2009.
26. M. Rückert, M. Schneider, and D. Schröder. Generic constructions for verifiably encrypted signatures without random oracles or NIZKs. In *ACNS*, volume 6123 of *Lecture Notes in Computer Science*, pages 69–86, 2010.
27. M. Rückert and D. Schröder. Security of verifiably encrypted signatures and a construction without random oracles. In *Pairing*, volume 5671 of *Lecture Notes in Computer Science*, pages 17–34. Springer, 2009.
28. V. Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT*, volume 1233 of *LNCS*, pages 256–266, 1997.
29. M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer, 2010.
30. B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT'05*, volume 3494 of *LNCS*, pages 114–127. Springer, 2005.
31. B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *CRYPTO'09*, volume 5677 of *LNCS*, pages 619–636. Springer, 2009. full version available from http://eprint.iacr.org/2009/385.
32. F. Zhang, R. Safavi-Naini, and W. Susilo. Efficient verifiably encrypted signature and partially blind signature from bilinear pairings. In *INDOCRYPT*, volume 2904 of *LNCS*, pages 191–204. Springer, 2003.