Tagged One-Time Signatures: Tight Security and Optimal Tag Size

Masayuki Abe[1], Bernardo David[2], Markulf Kohlweiss[3], Ryo Nishimaki[1], and Miyako Ohkubo[4]

[1)] NTT Secure Platform Laboratories
{abe.masayuki,nishimaki.ryo}@lab.ntt.co.jp

[2)] University of Brasilia
bernardo.david@aluno.unb.br

[3)] Microsoft Research, Cambridge
markulf@microsoft.com

[4)] Security Architecture Laboratory, NSRI, NICT
m.ohkubo@nict.go.jp

Abstract. We present an efficient structure-preserving tagged one-time signature scheme with tight security reductions to the decision-linear assumption. Our scheme features short tags consisting of a single group element and gives rise to the currently most efficient structure-preserving signature scheme based on the decision-liner assumption with constant-size signatures of only 14 group elements, where the record-so-far was 17 elements.

To demonstrate the advantages of our scheme, we revisit the work by Hofheinz and Jager (CRYPTO 2012) and present the currently most efficient tightly secure public-key encryption scheme. We also obtain the first structure-preserving public-key encryption scheme featuring both tight security and public verifiability.

Keywords. Tagged One-Time Signatures, Structure-Preserving Signatures, Tight Security Reduction, Decision Linear Assumption

1 Introduction

Background. A tagged one-time signature (TOS, [1]) scheme is a signature scheme that includes a fresh random tag in each signature. It is unforgeable if creating a signature on a new message but with an old tag picked by an honest signer is hard. A TOS is a special type of partial one-time signature (POS, [1], also known as two-tier signatures [9]), that involves one-time keys and long-term keys. Namely, a TOS is a POS with an empty one-time secret-key. For this reason the one-time public-key is called a tag.

A TOS is structure-preserving [3] if its long-term public-keys, tags, messages, and signatures consist only of elements of the base bilinear groups and the verification only evaluates pairing product equations. Structure-preservation grants interoperability among building blocks over the same bilinear groups and allows modular constructions of conceptually complex cryptographic schemes, in particular when combined with the

Groth-Sahai (GS) proof system [26]. So far, structure-preserving constructions have been developed for signature [24, 15, 3, 4, 16, 12, 1] commitments [5, 6], and public-key encryption schemes [13]. The growing list of their applications include universally composable adaptive oblivious transfer [23, 22], anonymous proxy signatures [19], delegatable anonymous credentials [8], transferable e-cash [20], compact verifiable shuffles [17], and network coding [7].

Efficiency and tight security is of general interest for cryptographic primitives. In [27], a tightly-secure structure-preserving POS is used as a central building block for constructing pubic-key encryption scheme secure against adaptive chosen-ciphertext attacks with multiple challenges and users. Replacing the POS with a TOS gives an immediate improvement. It is however seemingly more difficult to construct a TOS with high efficiency and tight security at the same time due to the absence of one-time secrets. To the best of our knowledge, the scheme in [1] is the only structure-preserving TOS in the literature bases on the decision-linear assumption (DLIN, [10]). Unfortunately, their reduction is not tight. For q_s signing queries, it suffers a factor of $1/q_s$. Moreover, a tag requires two group elements for technical reasons. This contrasts to the case of POS, where tight reductions to DLIN or SXDH are known, and the one-time public-key can be a single group element [1].

Our Contribution. The main contribution of this paper is a structure-preserving TOS with 1) optimally short tags consisting only of one group element, and 2) a tight security reduction to a computational assumption tightly implied by DLIN. Thus, when compared with the TOS scheme in [1], our scheme improves both tag size and tightness. The first application of our new TOS is a more efficient structure-preserving signature (SPS) scheme based on DLIN. The signature consists of 14 group elements and the verification evaluates 7 pairing product equations. It saves 3 group elements and 2 equations over previous SPS in [1]. Our second application is a more efficient tightly secure public-key encryption scheme. As a stepping stone we also obtain a more efficient and tight secure structure-preserving tree-based signature schemes. We obtain these results by revisiting the framework of [27]. In addition to the efficiency and key-management improvements, our contributions include the first structure-preserving CCA-secure encryption schemes featuring a tight security reduction, public verifiability, and leakage resilience which we inherit from [18].

The combined length of a tag and the long-term public-key in the new TOS is shorter than the one-time public-key of other structure-preserving one-time signature schemes (OTS) in the literature. (It saves 2 group elements over the OTS in [5].) Using our TOS as OTS is therefore beneficial even for applications that use the whole public-key only once. Typical examples include the IBE-to-PKE transformation [14], NIZK-to-SS-NIZK transformation [24], CCA-secure Group Signatures [25] where one-time signatures are used to add non-malleability, and delegatable anonymous credentials [8]. Though the improvement in this direction is small, it is often considerably amplified in applications, e.g., in delegatable anonymous credentials where the whole public key needs to be concealed in Groth-Sahai commitments.

Many of the applications in this paper require hundreds of group elements and are not necessarily practical. Nevertheless, the concrete efficiency assessment should serve as a reference that shows how efficient instantiation of generic modular constructions can be. In particular, as the constants in generic constructions can be large, we observed that small gains in the building blocks can result in significant efficiency improvements in applications.

2 Preliminaries

2.1 Bilinear Groups

We work in a setting with a symmetric bilinear pairing (the Type-I setting of [21]) and use multiplicative notation. Let $\mathcal{G}$ be a bilinear group generator that takes security parameter λ as input and outputs a description of bilinear groups $\Lambda := (p, \mathbb{G}, \mathbb{G}_T, e)$, where $\mathbb{G}$ and $\mathbb{G}_T$ are groups of prime order p, and e is an efficient and non-degenerating bilinear map $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. We count the number of group elements to measure the size of cryptographic objects such as keys, messages, and signatures. By $\mathbb{Z}_p$ and $\mathbb{Z}_p^*$, we denote $\mathbb{Z}/p\mathbb{Z}$ and $\mathbb{Z}/p\mathbb{Z} \setminus \{0\}$, respectively. We abuse the notation and denote $\mathbb{G} \setminus \{1_{\mathbb{G}}\}$ by $\mathbb{G}^*$.

The security of our schemes is based on the following computational assumption.

Definition 1 (Simultaneous Double Pairing Assumption : SDP [15]). *For the bilinear group generator $\mathcal{G}$ and any polynomial time $\mathcal{A}$ the probability*

$$\mathrm{Adv}^{\mathsf{sdp}}_{\mathcal{G},\mathcal{A}}(\lambda) := \Pr\left[\begin{array}{ll} \Lambda \leftarrow \mathcal{G}(1^\lambda) & Z \in \mathbb{G}^* \wedge \\ (G_z, G_r, H_z, H_s) \leftarrow \mathbb{G}^{*4} & : 1 = e(G_z, Z)\, e(G_r, R) \wedge \\ (Z, R, S) \leftarrow \mathcal{A}(\Lambda, G_z, G_r, H_z, H_s) & \ \ 1 = e(H_z, Z)\, e(H_s, S) \end{array}\right]$$

is negligible in λ.

SDP is random-self reducible. Given (G_z, G_r, H_z, H_s), another random instance $(G_z^a G_r^b, G_r^c, H_z^a H_s^d, H_s^e)$ can be generated by choosing a, b, c, d, and e uniformly from $\mathbb{Z}_p^*$. Given an answer (Z, R, S) to the new instance, $(Z^a, R^c Z^b, S^e Z^d)$ is the answer to the original instance. Furthermore, SDP is tightly reduced from DLIN as observed in [15]. For a DLIN instance $(G_1, G_2, G_3, G_1^a, G_2^b, G_3^c)$ for deciding $c = a + b$ or not, construct an SDP instance (G_1^a, G_1, G_2^b, G_2). Then, given an answer (Z, R, S) that satisfies $1 = e(G_1^a, Z)\ e(G_1, R)$ and $1 = e(G_2^b, Z)\ e(G_2, S)$, one can conclude that $c = a + b$ if $e(G_3, R \cdot S) = e(G_3^c, Z)$ since $R = Z^a$ and $S = Z^b$. We restate this observation as a lemma below.

Lemma 1 (DLIN $\Rightarrow$ SDP). *If there exists adversary $\mathcal{A}$ that solves SDP, then there exists adversary $\mathcal{B}$ that solves DLIN with the same advantage and a runtime overhead of a few exponentiations and pairings.*

2.2 Syntax and Security Notions

We follow the syntax and security notions for TOS in [1]. Let $\mathsf{Setup}(1^\lambda)$ be an algorithm that takes security parameter λ and outputs common parameter gk. Parameters gk are (sometimes implicit) input to all algorithms.

Definition 2 (Tagged One-Time Signature Scheme). *A tagged one-time signature scheme* TOS *is a set of polynomial-time algorithms* $\mathsf{TOS}.\{\mathsf{Key}, \mathsf{Tag}, \mathsf{Sign}, \mathsf{Vrf}\}$ *that takes gk generated by* Setup. *Each function works as follows.*

- $\mathsf{TOS.Key}(gk)$ *generates a long-term public-key pk and a secret-key sk. Message space $\mathcal{M}_t$ and tag space $\mathcal{T}$ are determined by gk.*
- $\mathsf{TOS.Tag}(gk)$ *takes gk as input and outputs $tag \in \mathcal{T}$.*
- $\mathsf{TOS.Sign}(sk, msg, tag)$ *outputs signature σ for message msg based on secret-key sk and tag tag.*
- $\mathsf{TOS.Vrf}(pk, tag, msg, \sigma)$ *outputs 1 for acceptance, or 0 for rejection.*

For any key $(pk, sk) \leftarrow \mathsf{TOS.Key}(\mathsf{Setup}(1^\lambda))$, any message $msg \in \mathcal{M}_t$, any tag $tag \leftarrow \mathsf{TOS.Tag}(gk)$, and any signature $\sigma \leftarrow \mathsf{TOS.Sign}(sk, msg, tag)$, verification $\mathsf{TOS.Vrf}(pk, tag, msg, \sigma)$ outputs 1.

TOS is called uniform-tag if the output distribution of tag is uniform over $\mathcal{T}$. TOS is structure-preserving over Λ if gk contains Λ and the public-keys, messages, tags, and signatures consist only of elements of base groups of Λ and $\mathsf{TOS.Vrf}$ consists of evaluating pairing product equations.

Definition 3 (Unforgeability against One-Time Tag Chosen-Message Attacks). *For tagged one-time signature scheme* TOS *and algorithm $\mathcal{A}$, let* $\mathrm{Expr}^{\mathsf{ot\text{-}cma}}_{\mathsf{TOS},\mathcal{A}}$ *be an experiment that:*

$$
\begin{aligned}
&\mathrm{Expr}^{\mathsf{ot\text{-}cma}}_{\mathsf{TOS},\mathcal{A}}(1^\lambda) := \\
&\quad gk \leftarrow \mathsf{Setup}(1^\lambda),\ (pk, sk) \leftarrow \mathsf{TOS.Key}(gk) \\
&\quad (tag^\dagger, \sigma^\dagger, msg^\dagger) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{tag}}, \mathcal{O}_{\mathsf{sig}}}(pk) \\
&\quad \textit{If } \exists (tag, msg, \sigma) \in Q_m \textit{ s.t.} \\
&\qquad tag^\dagger = tag \ \wedge\ msg^\dagger \neq msg \ \wedge\ 1 = \mathsf{TOS.Vrf}(pk, tag^\dagger, \sigma^\dagger, msg^\dagger) \\
&\quad \textit{return 1. Return 0, otherwise.}
\end{aligned}
$$

$\mathcal{O}_{\mathsf{tag}}$ and $\mathcal{O}_{\mathsf{sig}}$ are tag and signature generation oracles, respectively. On receiving i-th query, $\mathcal{O}_{\mathsf{tag}}$ returns tag tag_i generated by $\mathsf{TOS.Tag}(gk)$. *On receiving j-th query with message msg_j as input (if at this point $\mathcal{O}_{\mathsf{tag}}$ has been received $i < j$ requests, $\mathcal{O}_{\mathsf{tag}}$ is invoked to generate tag_j), $\mathcal{O}_{\mathsf{sig}}$ performs $\sigma_j \leftarrow \mathsf{TOS.Sign}(sk, msg_j, tag_j)$, appends (tag_j, msg_j, σ_j) to Q_m, and returns σ_j (and tag_j if generated) to $\mathcal{A}$.*

A tagged one-time signature scheme is unforgeable against one-time tag adaptive chosen message attacks (OT-CMA) if for all polynomial-time oracle algorithms $\mathcal{A}$ the advantage function $\mathrm{Adv}^{\mathsf{ot\text{-}cma}}_{\mathsf{TOS},\mathcal{A}} := \Pr[\mathrm{Expr}^{\mathsf{ot\text{-}cma}}_{\mathsf{TOS},\mathcal{A}}(1^\lambda) = 1]$ is negligible in λ.

Strong unforgeability is a variation on this definition obtained by replacing the condition $msg^\dagger \neq msg$ in the experiment with $(msg^\dagger, \sigma^\dagger) \neq (msg, \sigma)$. Another variation is non-adaptive attack unforgeability (OT-NACMA) defined by integrating $\mathcal{O}_{\mathsf{tag}}$ into $\mathcal{O}_{\mathsf{sig}}$ so that tag_j and σ_j are returned to $\mathcal{A}$ at the same time. Namely, $\mathcal{A}$ must submit msg_j before seeing tag_j. It is obvious that if a scheme is secure in the sense of OT-CMA, the scheme is also secure in the sense of OT-NACMA. By $\mathrm{Adv}^{\mathsf{ot\text{-}nacma}}_{\mathsf{TOS},\mathcal{A}}(\lambda)$ we denote the advantage of $\mathcal{A}$ in this non-adaptive case. We use labels sot-cma and sot-nacma for adaptive and non-adaptive strong unforgeability respectively.

For signatures we follow the standard syntax of digital signatures with common setup. Namely, a signature scheme consists of three algorithms $\mathsf{SIG}.\{\mathsf{Key}, \mathsf{Sign}, \mathsf{Vrf}\}$ that take gk generated by Setup as additional input. SIG.Key is a key generation algorithm, SIG.Sign is a signing algorithm and SIG.Vrf is a verification algorithm. We also follow the standard security notion of existential unforgeability against adaptive chosen message attacks.

2.3 A Framework of TOS + RMA-SIG

We review the framework of combining TOS and RMA signatures in [1] to obtain a CMA-secure signature scheme. Let rSIG be a signature scheme with message space $\mathcal{M}_{\mathsf{r}}$, and TOS be a tagged one-time signature scheme with tag space $\mathcal{T}$ such that $\mathcal{M}_{\mathsf{r}} = \mathcal{T}$. We construct a signature scheme SIG from rSIG and TOS. Let gk be a global parameter generated by $\mathsf{Setup}(1^\lambda)$.

[Generic Construction: SIG**]**

$\mathsf{SIG.Key}(gk)$: Run $(pk_t, sk_t) \leftarrow \mathsf{TOS.Key}(gk)$, $(vk_r, sk_r) \leftarrow \mathsf{rSIG.Key}(gk)$. Output $vk := (pk_t, vk_r)$ and $sk := (sk_t, sk_r)$.

$\mathsf{SIG.Sign}(sk, msg)$: Parse sk into (sk_t, sk_r). Output $\sigma := (tag, \sigma_t, \sigma_r)$ where $tag \leftarrow \mathsf{TOS.Tag}(gk)$, $\sigma_t \leftarrow \mathsf{TOS.Sign}(sk_t, msg, tag)$, and $\sigma_r \leftarrow \mathsf{rSIG.Sign}(sk_r, tag)$.

$\mathsf{SIG.Vrf}(vk, \sigma, msg)$: Parse vk and σ accordingly. Output 1, if $1 = \mathsf{TOS.Vrf}(pk_t, tag, \sigma_t, msg)$ and $1 = \mathsf{rSIG.Vrf}(vk_r, \sigma_r, tag)$. Output 0, otherwise.

The following theorems are due to [1].

Theorem 1. SIG *is unforgeable against adaptive chosen message attacks (UF-CMA) if* TOS *is uniform-tag and unforgeable against one-time non-adaptive chosen message attacks (OT-NACMA), and* rSIG *is unforgeable against random message attacks (UF-RMA). In particular,* $\mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{SIG},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathsf{ot\text{-}nacma}}_{\mathsf{TOS},\mathcal{B}}(\lambda) + \mathrm{Adv}^{\mathsf{uf\text{-}rma}}_{\mathsf{rSIG},\mathcal{C}}(\lambda)$. *The overhead of adversary* $\mathcal{B}$ *against* rSIG *and* $\mathcal{C}$ *against* TOS *is proportional to the running time of the key generation and signing operations of* rSIG *and* TOS *respectively.*

Theorem 2. *If* TOS.Tag *produces constant-size tags and signatures in the size of input messages, the resulting* SIG *produces constant-size signatures as well. Furthermore, if* TOS *and* rSIG *are structure-preserving, so is* SIG.

3 Tightly-Secure TOS based on DLIN

Let gk be a global parameter that specifies $\Lambda = (p, \mathbb{G}, \mathbb{G}_T, e)$ generated by group generator $\mathcal{G}(1^\lambda)$. It also includes a generator $G \in \mathbb{G}^*$. We construct $\mathsf{TOS}.\{\mathsf{Key}, \mathsf{Tag}, \mathsf{Sign}, \mathsf{Vrf}\}$ as shown in Fig. 1.

[Scheme TOS**]**

$\mathsf{TOS.Key}(gk)$: Parse $gk = (\Lambda, G)$. Choose $w_z,\ w_r, \mu_z,\ \mu_s, \tau$ randomly from $\mathbb{Z}_p^*$ and compute $G_z := G^{w_z}$, $G_r := G^{w_r}$, $H_z := G^{\mu_z}$, $H_s := G^{\mu_s}$, $G_t := G^\tau$ and For $i = 1, \ldots, k$, uniformly choose $\chi_i, \gamma_i, \delta_i$ from $\mathbb{Z}_p$ and compute

$$G_i := G_z^{\chi_i} G_r^{\gamma_i}, \quad \text{and} \quad H_i := H_z^{\chi_i} H_s^{\delta_i}. \tag{1}$$

Output $pk := (G_z, G_r, H_z, H_s, G_t, G_1, \ldots, G_k, H_1, \ldots, H_k) \in \mathbb{G}^{2k+5}$ and $sk := (\chi_1, \gamma_1, \delta_1, \ldots, \chi_k, \gamma_k, \delta_k, w_z, w_r, \mu_z, \mu_s, \tau) \in \mathbb{Z}_p^{3k+5}$.

$\mathsf{TOS.Tag}(gk)$: Choose $t \leftarrow \mathbb{Z}_p^*$ and output $tag := T = G^t \in \mathbb{G}$.

$\mathsf{TOS.Sign}(sk, msg, tag)$: Parse msg into $(M_1, \ldots, M_k) \in \mathbb{G}^k$. Take T_1 from tag. Parse sk accordingly. Output $\sigma := (Z, R, S) \in \mathbb{G}^3$ that, for $\zeta \leftarrow \mathbb{Z}_p$,

$$Z := G^\zeta \textstyle\prod_{i=1}^k M_i^{-\chi_i}, \quad R := (T^\tau G_z^{-\zeta})^{\frac{1}{w_r}} \prod_{i=1}^k M_i^{-\gamma_i}, \text{ and}$$
$$S := (H_z^{-\zeta})^{\frac{1}{\mu_s}} \textstyle\prod_{i=1}^k M_i^{-\delta_i}.$$

$\mathsf{TOS.Vrf}(pk, tag, msg, \sigma)$: Parse σ as $(Z, R, S) \in \mathbb{G}^3$, msg as $(M_1, \ldots, M_k) \in \mathbb{G}^k$, and take T from tag. Return 1 if the following equations hold. Return 0, otherwise.

$$e(T, G_t) = e(G_z, Z)\ e(G_r, R) \prod_{i=1}^k e(G_i, M_i) \tag{2}$$

$$1 = e(H_z, Z)\ e(H_s, S) \prod_{i=1}^k e(H_i, M_i) \tag{3}$$

Fig. 1. Tagged One-Time Signature Scheme.

Correctness is verified by inspecting the following relations.

For (2): $$e(G_z, G^\zeta \prod_{i=1}^k M_i^{-\chi_i})\ e(G_r, (T^\tau G_z^{-\zeta})^{\frac{1}{w_r}} \prod_{i=1}^k M_i^{-\gamma_i}) \prod_{i=1}^k e(G_z^{\chi_i} G_r^{\gamma_i}, M_i)$$
$$= e(G_z, G^\zeta)\ e(G, T^\tau) e(G, G_z^{-\zeta}) = e(G, T^\tau) = e(T, G_t)$$

For (3): $$e(H_z, G^\zeta \prod_{i=1}^k M_i^{-\chi_i})\ e(H_s, (H_z^{-\zeta})^{\frac{1}{\mu_s}} \prod_{i=1}^k M_i^{-\delta_i}) \prod_{i=1}^k e(H_z^{\chi_i} H_s^{\delta_i}, M_i)$$
$$= e(H_z, G^\zeta)\ e(G, H_z^{-\zeta}) = 1$$

We state the following theorems, of which the first one is immediate from the construction.

Theorem 3. *Above* TOS *is structure-preserving, and yields uniform tags and constant-size signatures.*

Theorem 4. *Above* TOS *is strongly unforgeable against one-time tag adaptive chosen message attacks (SOT-CMA) if the SDP assumption holds. In particular,* $\mathrm{Adv}^{\mathsf{sot\text{-}cma}}_{\mathsf{TOS},\mathcal{A}} \leq \mathrm{Adv}^{\mathsf{sdp}}_{\mathcal{G},\mathcal{B}} + 1/p$ *and the runtime overhead of the reduction* $\mathcal{B}$ *is a small number of multi-exponentiations per signing query.*

Proof. Given successful forger $\mathcal{A}$ against TOS as a black-box, we construct $\mathcal{B}$ that breaks SDP as follows. Let $I_{\mathsf{sdp}} = (\Lambda, G_z, G_r, H_z, H_s)$ be an instance of SDP. Algorithm $\mathcal{B}$ simulates the attack game against TOS as follows. It first build $gk := (\Lambda, G, G)$ by choosing G randomly from $\mathbb{G}^*$. This yields a gk in the same distribution as produced by Setup. Next $\mathcal{B}$ simulates $\mathsf{TOS.Key}$ by taking (G_z, G_r, H_z, H_s) from I_{sdp} and computing $G_t := H_s^{\tau}$ for random τ in $\mathbb{Z}_p^*$. It then generates G_i and H_i according to (1). This perfectly simulates $\mathsf{TOS.Key}$.

On receiving the j-th query to $\mathcal{O}_{\mathsf{tag}}$, algorithm $\mathcal{B}$ computes

$$T := (G_z^{\zeta} G_r^{\rho})^{\frac{1}{\tau}} \tag{4}$$

for $\zeta, \rho \leftarrow \mathbb{Z}_p^*$. If $T = 1$, $\mathcal{B}$ sets $Z^\star := H_s$, $S^\star := H_z^{-1}$, and $R^\star := (Z^\star)^{\rho/\zeta}$, outputs $(Z^\star, R^\star, S^\star)$ and stop. Otherwise, $\mathcal{B}$ stores (ζ, ρ) and returns $tag_j := T$ to $\mathcal{A}$.

On receiving signing query $msg_j = (M_1, \ldots, M_k)$, algorithm $\mathcal{B}$ takes ζ and ρ used for computing tag_j (if they are not yet defined, invoke the procedure for $\mathcal{O}_{\mathsf{tag}}$) and computes

$$Z := H_s^{\zeta} \prod_{i=1}^{k} M_i^{-\chi_i}, \quad R := H_s^{\rho} \prod_{i=1}^{k} M_i^{-\gamma_i}, \quad \text{and} \quad S := H_z^{-\zeta} \prod_{i=1}^{k} M_i^{-\delta_i}. \tag{5}$$

Then $\mathcal{B}$ returns $\sigma_j := (Z, R, S)$ to $\mathcal{A}$ and record (tag_j, σ_j, msg_j).

When $\mathcal{A}$ outputs a forgery $(tag^\dagger, \sigma^\dagger, msg^\dagger)$, algorithm $\mathcal{B}$ searches the records for (tag, σ, msg) such that $tag^\dagger = tag$ and $(msg^\dagger, \sigma^\dagger) \neq (msg, \sigma)$. If no such entry exists, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ computes

$$Z^\star := \frac{Z^\dagger}{Z} \prod_{i=1}^{k} \left(\frac{M_i^\dagger}{M_i}\right)^{\chi_i}, \quad R^\star := \frac{R^\dagger}{R} \prod_{i=1}^{k} \left(\frac{M_i^\dagger}{M_i}\right)^{\gamma_i}, \text{ and } \quad S^\star := \frac{S^\dagger}{S} \prod_{i=1}^{k} \left(\frac{M_i^\dagger}{M_i}\right)^{\delta_i}$$

where (Z, R, S), $(M_1, \ldots, M_k)$ and their dagger counterparts are taken from (σ, msg) and $(\sigma^\dagger, msg^\dagger)$, respectively. $\mathcal{B}$ finally outputs $(Z^\star, R^\star, S^\star)$ and stops. This completes the description of $\mathcal{B}$.

We claim that $\mathcal{B}$ solves the problem by itself or the view of $\mathcal{A}$ is perfectly simulated. The correctness of key generation has been already inspected. In the simulation of $\mathcal{O}_{\mathsf{tag}}$, there is a case of $T = 1$ that happens with probability $1/q$. If it happens, $\mathcal{B}$ outputs a correct answer to I_{sdp}, which is inspected by observing $G_z = G_r^{-\rho/\zeta}$, $Z^\star = H_s \neq 1$, $e(G_z, Z^\star)e(G_r, R^\star) = e(G_r^{-\rho/\zeta}, Z^\star)e(G_r, (Z^\star)^{\rho/\zeta}) = 1$ and $e(H_z, Z^\star)e(H_s, S^\star) = e(H_z, H_s)e(H_s, H_z^{-1}) = 1$. Otherwise, tag T uniformly distributes over $\mathbb{G}^*$ and the simulation is perfect.

Oracle $\mathcal{O}_{\mathsf{sig}}$ is simulated perfectly as well. Correctness of simulated $\sigma_j = (Z,\, R,\, S)$ can be verified by inspecting the following relations.

$$
\begin{aligned}
\text{(Right-hand of (2))} &= e(G_z, H_s^{\zeta}\prod_{i=1}^{k} M_i^{-\chi_i})\; e(G_r, H_s^{\rho}\prod_{i=1}^{k} M_i^{-\gamma_i})\; \prod_{i=1}^{k} e(G_z^{\chi_i}G_r^{\gamma_i}, M_i) \\
&= e(G_z^{\zeta}G_r^{\rho}, H_s) = e((G_z^{\zeta}G_r^{\rho})^{\frac{1}{\tau}}, H_s^{\tau}) = e(T_1, G_t) \\
\text{(Right-hand of (3))} &= e(H_z, H_s^{\zeta}\prod_{i=1}^{k} M_i^{-\chi_i})\; e(H_s, H_z^{-\zeta}\prod_{i=1}^{k} M_i^{-\delta_i})\; \prod_{i=1}^{k} e(H_z^{\chi_i}H_s^{\delta_i}, M_i) \\
&= e(H_z, H_s^{\zeta})\; e(H_s, H_z^{-\zeta}) \;= 1
\end{aligned}
$$

Every Z distributes uniformly over $\mathbb{G}$ due to the uniform choice of ζ. Then R and S are uniquely determined by following the distribution of Z.

Accordingly, $\mathcal{A}$ outputs successful forgery with noticeable probability and $\mathcal{B}$ finds a corresponding record (tag, σ, msg). We show that output $(Z^\star, R^\star, S^\star)$ from $\mathcal{B}$ is a valid solution to I_{sdp}. First, equation (2) is satisfied because

$$
\begin{aligned}
1 &= e\left(G_z, \frac{Z^\dagger}{Z}\right)\; e\left(G_r, \frac{R^\dagger}{R}\right) \prod_{i=1}^{k} e\left(G_z^{\chi_i}G_r^{\gamma_i}, \frac{M_i^\dagger}{M_i}\right) \\
&= e\left(G_z, \frac{Z^\dagger}{Z}\prod_{i=1}^{k}\left(\frac{M_i^\dagger}{M_i}\right)^{\chi_i}\right)\; e\left(G_r, \frac{R^\dagger}{R}\prod_{i=1}^{k}\left(\frac{M_i^\dagger}{M_i}\right)^{\gamma_i}\right) \\
&= e\left(G_z, Z^\star\right)\; e\left(G_r, R^\star\right),
\end{aligned}
$$

holds. Equation (3) is verified similarly.

It remains to prove that $Z^\star \neq 1$. Since $msg^\dagger \neq msg$, there exists $\ell \in \{1, \ldots, k\}$ such that $M_\ell^\dagger / M_\ell \neq 1$. We claim that, parameter $\chi_1, \ldots, \chi_k$ are independent of the view of $\mathcal{A}$. We prove it by showing that, for every possible assignment to $\chi_1, \ldots, \chi_k$, there exists an assignment to other coins, i.e., $(\gamma_1, \ldots, \gamma_k,\, \delta_1, \ldots, \delta_k)$ and $(\zeta^{(1)}, \rho^{(1)}, \ldots, \zeta^{(q_s)}, \rho^{(q_s)})$ for q_s queries, that is consistent to the view of $\mathcal{A}$. (By $\zeta^{(j)}$, we denote ζ with respect to the j-th query. We follow this convention hereafter. Without loss of generality, we assume that $\mathcal{A}$ makes q_s tag queries and the same number of signing queries.) Observe that the view of $\mathcal{A}$ consists of independent group elements $(G, G_z, G_r, H_z, H_s, G_t, G_1, H_1, \ldots, G_k, H_k)$ and $(T^{(j)}, Z^{(j)}, M_1^{(j)}, \ldots, M_k^{(j)})$ for $j = 1, \ldots, q_s$. (Note that $R^{(j)}$ and $S^{(j)}$ are not in the view since they are uniquely determined from other components.) We represent the view by the discrete-logarithms of these group elements with respect to base G. Namely, the view is $(1, w_z, w_r, \mu_z, \mu_s, \tau, w_1, \mu_1, \ldots, w_k, \mu_k)$ and $(t^{(j)}, z^{(j)}, m_1^{(j)}, \ldots, m_k^{(j)})$ for $j = 1, \ldots, q_s$. The view and the random coins follow relations from (1), (4), and (5) translated to

$$w_i = w_z\chi_i + w_r\gamma_i, \quad \mu_i = \mu_z\chi_i + \mu_s\delta_i \quad \text{for } i = 1, \ldots, k, \tag{6}$$

$$\tau t^{(j)} = w_z\zeta^{(j)} + w_r\rho^{(j)}, \text{ and} \tag{7}$$

$$z^{(j)} = \mu_s\,\zeta^{(j)} - \sum_{i=1}^{k} m_i^{(j)}\chi_i \quad \text{for } j = 1, \ldots, q_s. \tag{8}$$

Consider χ_ℓ for some $\ell \in \{1, \ldots, k\}$. For every value of χ_ℓ in $\mathbb{Z}_p$, the linear equations in (6) determine γ_ℓ and δ_ℓ. Then, if $m_\ell^{(j)} \neq 0$, equations in (8) determine $\zeta^{(j)}$, $\rho^{(j)}$. If $m_\ell^{(j)} = 0$, then $\zeta^{(j)}$, $\rho^{(j)}$ can be assigned independently from χ_ℓ. The above holds for every ℓ in $\{1, \ldots, k\}$. Thus, if $\chi_1, \ldots, \chi_k$ distributes uniformly over $\mathbb{Z}_p^k$, then other coins distribute uniformly as well retaining the consistency with the view of $\mathcal{A}$.

Now we see that $\left(M_\ell^\dagger / M_\ell\right)^{\chi_\ell}$ distributes uniformly over $\mathbb{G}$. Therefore $Z^\star = 1$ happens only with probability $1/p$. Thus, $\mathcal{B}$ outputs correct answer with probability $\mathrm{Adv}^{\mathsf{sdp}}_{\mathcal{G},\mathcal{B}} = 1/p + (1-1/p)(1-1/p)\mathrm{Adv}^{\mathsf{sot\text{-}cma}}_{\mathsf{TOS},\mathcal{A}}$, which leads to $\mathrm{Adv}^{\mathsf{sot\text{-}cma}}_{\mathsf{TOS},\mathcal{A}} \leq \mathrm{Adv}^{\mathsf{sdp}}_{\mathcal{G},\mathcal{B}} + 1/p$ as claimed. $\square$

Remark 1. On tag extension. The tag can be easily extended to the form $(G^t, G_1^t, G_2^t, \ldots)$ for extra bases $G_1, G_2, \ldots$ provided as a part of gk. (In the security proof, the extended part is computed from the first element by using $\log_G G_i$. This is possible since the extra generators in gk are chosen by the reduction algorithm.) Such an extension is in particular needed when the TOS is coupled with other signature schemes whose message space is structured as above. Indeed, it is the case for an application in Section 4.

Remark 2. Signing lengthy messages. The TOS can be used to sign messages of unbound length by chaining the signatures. Every message block except for the last one is followed by a tag used to sign the next block. The signature consists of all internal signatures and tags. The initial tag is considered as the tag. For a message consisting of m group elements, it repeats $\tau := 1 + \max(0, \lceil \frac{m-k}{k-1} \rceil)$ times. The signature consists of $4\tau - 2$ elements.

4 Efficient SPS based on DLIN

As the first application of our TOS, we present an efficiet structure-preserving signature scheme. The construction follows the framework suggested in Theorem 1. We begin with introducing an RMA-secure SPS as a building block. The scheme in Fig. 2 is an RMA-secure SPS for messages in the form $(C^m, F^m, U^m) \in \mathbb{G}^3$ defined by generators (C, F, U) provided in gk. The scheme is a modification of the one in Sec.5.3 of [1] that signs longer message of the form $\{(C^{m_1}, C^{m_2}, F^{m_1}, F^{m_2}, U^{m_1}, U^{m_2})\}$. Our scheme is obtained by restricting $m_2 = 0$ and removing useless operations relevant to m_2. The security is stated in Theorem 5 below, whose proof is obtained as a trivial modification of the proof of Theorem 24 in [1].

Theorem 5. *The above* rSIG *scheme is secure against random message attacks under the DLIN assumption. In particular, for any polynomial-time adversary* $\mathcal{A}$ *against* rSIG *that makes at most* q_s *signing queries, there exists polynomial-time algorithm* $\mathcal{B}$ *for DLIN such that* $\mathrm{Adv}^{\mathsf{uf\text{-}rma}}_{\mathsf{rSIG},\mathcal{A}}(\lambda) \leq (q_s + 2) \cdot \mathrm{Adv}^{\mathsf{dlin}}_{\mathcal{G},\mathcal{B}}(\lambda)$.

[Scheme rSIG**]**

Let gk be a common parameter that consists of group description $\Lambda = (p, \mathbb{G}, \mathbb{G}_T, e)$ and default generator G. It also includes randomly chosen generators C, F, and U.

$\mathsf{rSIG.Key}(gk)$**:** Given $gk := (\Lambda, G, C, F, U)$ as input, uniformly select V, V_1, V_2, H from $\mathbb{G}^*$ and a_1, a_2, b, α, and ρ from $\mathbb{Z}_p^*$. Then compute and output $vk := (B, A_1, A_2, B_1, B_2, R_1, R_2, W_1, W_2, V, V_1, V_2, H, X_1, X_2)$ and $sk := (vk, K_1, K_2)$ where

$$B := G^b, \quad A_1 := G^{a_1}, \quad A_2 := G^{a_2}, \quad B_1 := G^{b \cdot a_1}, \quad B_2 := G^{b \cdot a_2}$$
$$R_1 := V V_1^{a_1}, \quad R_2 := V V_2^{a_2}, \quad W_1 := R_1^b, \quad W_2 := R_2^b,$$
$$X_1 := G^{\rho}, \quad X_2 := G^{\alpha \cdot a_1 \cdot b/\rho}, \quad K_1 := G^{\alpha}, \quad K_2 := G^{\alpha \cdot a_1}.$$

$\mathsf{rSIG.Sign}(sk, msg)$**:** Parse msg into (M_1, M_2, M_3). Pick random $r_1, r_2, z_1, z_2 \in \mathbb{Z}_p$. Let $r = r_1 + r_2$. Compute and output signature $\sigma := (S_0, S_1, \ldots S_7)$ where

$$S_0 := (M_3 H)^{r_1}, \quad S_1 := K_2 V^r, \quad S_2 := K_1^{-1} V_1^r G^{z_1}, \quad S_3 := B^{-z_1},$$
$$S_4 := V_2^r G^{z_2}, \quad S_5 := B^{-z_2}, \quad S_6 := B^{r_2}, \quad S_7 := G^{r_1}.$$

$\mathsf{rSIG.Vrf}(vk, \sigma, msg)$**:** Parse msg into (M_1, M_2, M_3) and σ into $(S_0, S_1, \ldots, S_7)$. Also parse vk accordingly. Verify the following pairing product equations:

$$e(S_7, M_3 H) = e(G, S_0),$$
$$e(S_1, B)\, e(S_2, B_1)\, e(S_3, A_1) = e(S_6, R_1)\, e(S_7, W_1),$$
$$e(S_1, B)\, e(S_4, B_2)\, e(S_5, A_2) = e(S_6, R_2)\, e(S_7, W_2)\, e(X_1, X_2),$$
$$e(F, M_1) = e(C, M_2), \quad e(U, M_1) = e(C, M_3)$$

Fig. 2. RMA-secure SPS for 1 message block based on DLIN.

According to Theorem 1, combining TOS in Section 3 and rSIG in Fig. 2 results in a chosen-message-secure SPS. (Note that tags of TOS are extended as explained in the remark in the end of Section 3 so that they fit to the message space of rSIG. Concretely, by using generator C from rSIG as G in the description of TOS, and also using extra generators F and U, a tag is defined as $(T_1, T_2, T_3) := (C^t, F^t, U^t)$.) The resulting

SPS yields signatures consisting of 14 group elements $(T_1, T_2, T_3, Z, R, S, S_0, \ldots, S_7)$ and evaluates 7 pairing product equations in the verification. Since both TOS and rSIG are based on DLIN, the resulting SPS is secure under DLIN as well. (They are actually based on SDP that is a seemingly weaker computational assumption.)

The efficiency is summarised in Table 1. It is compared to existing efficient structure-preserving schemes over symmetric bilinear groups. We measure efficiency by counting the number of group elements and the number of pairing product equations for verifying a signature. The figures do not count default generator G in gk.

To see how a small difference in the size of signatures and the number of PPEs impacts the efficiency in applications, we assess the cost of proving possession of valid signatures and messages by using Groth-Sahai NIWI proof system. Column "Proof Cost σ" shows the number of group elements in the commitment of a signature and the proof. If there are randomizable parts in a signature, they are put in the clear. It is the case for the scheme in [3]. Similarly, column "Proof Cost (σ, msg)" shows the size when both messages and signatures are committed as witnesses.

Scheme	$\|msg\|$	$\|gk\| + \|vk\|$	$\|\sigma\|$	#(PPE)	Assumption	Reduction Cost	Proof Cost σ	Proof Cost (msg, σ)
[3]	k	$2k + 12$	7	2	q-SFP	1	19	3k+19
[1]	k	$2k + 25$	17	9	DLIN	$(2q)^{-1}$	84	3k+84
this paper	k	$2k + 20$	14	7	DLIN	$(q+1)^{-1}$	69	3k+69

Table 1. Comparison of constant-size SPS over symmetric bilinear groups. "Reduction" shows the loss factor to the underlying assumptions. "Proof Size" is the number of group elements in the Groth-Sahai NIWI proof of knowledge about a valid signature.

5 Chosen-Ciphertext Secure Public-Key Encryption

5.1 Simulation Extractable NIZK

A non-interactive zero-knowledge argument system $\mathsf{NIZK} = \mathsf{NIZK}.\{\mathsf{Crs}, \mathsf{Prv}, \mathsf{Vrf}\}$ for a relation R consists of three algorithms: $\mathsf{NIZK.Crs}$ that takes a common setup parameter and generates a common reference string crs, the proof algorithm $\mathsf{NIZK.Prv}$ which on input crs, an instance x and a witness w for the truth of the statement R, outputs a proof π, and the verification algorithm $\mathsf{NIZK.Vrf}$ that on input crs, an instance x, and a proof π either accepts or rejects the proof. It is equipped with a pair of algorithms, $\mathsf{NIZK.CrsSim}$ and $\mathsf{NIZK.PrvSim}$, that simulates $\mathsf{NIZK.Crs}$ and $\mathsf{NIZK.Prv}$, respectively. $\mathsf{NIZK.CrsSim}$ outputs crs and a simulation-trapdoor, τ_{zk}, and $\mathsf{NIZK.PrvSim}$ produces proofs by using the trapdoor. NIZK is (unbounded multi-theorem) zero-knowledge, if given oracle access to either $\mathsf{NIZK.PrvSim}(\tau_{\mathsf{zk}}, \cdot)$ or $\mathsf{NIZK.Prv}(crs, \cdot, \cdot)$, with true statements as inputs, any polynomial-time adversary trying to distinguish the oracles has advantage upper bounded by a negligible function, ϵ_{zk}, in the security parameter. A NIZK is strongly simulation-sound if adversary $\mathcal{A}$ is given oracle access to

$\mathsf{NIZK.PrvSim}(\tau_{\mathsf{zk}}, \cdot)$ and outputs valid (x, π) only with negligible probability. It can be relaxed to standard simulation soundness by requiring that only x is not reused.

A NIZK is a non-interactive proof of knowledge [29] if $\mathsf{NIZK.Crs}$ additionally outputs an extraction trapdoor, τ_{ex}, and there exists an efficient algorithm, $\mathsf{NIZK.Ext}$, that extracts a correct witness w from any (x, π) that $1 = \mathsf{NIZK.Vrf}(crs, x, \pi)$ with probability $1 - \epsilon_{\mathsf{ks}}$ for some negligible function ϵ_{ks}. This property is called knowledge soundness. A simulation-extractable NIZK extends a NIZK proof of knowledge so that $\mathsf{NIZK.Crs}$ outputs extraction trapdoor τ_{ex} and simulation trapdoor τ_{zk} at the same time. Then it is simulation-extractable if $\mathsf{NIZK.Ext}$ works even if an adversary is given oracle access to $\mathsf{NIZK.PrvSim}(\tau_{\mathsf{zk}}, \cdot)$. More precisely,

$$\Pr\left[\begin{array}{l|l} (crs, \tau_{\mathsf{ex}}, \tau_{\mathsf{zk}}) \leftarrow \mathsf{NIZK.Crs} & \mathsf{NIZK.Vrf}(crs, x, \pi) = 1 \wedge \\ (x, \pi) \leftarrow \mathcal{A}^{\mathsf{NIZK.PrvSim}(\tau_{\mathsf{zk}}, \cdot)}(crs) & R(x, w) \neq 1 \wedge \\ w \leftarrow \mathsf{NIZK.Ext}(crs, x, \pi, \tau_{\mathsf{ex}}) & x \notin Q \end{array}\right] < \epsilon_{\mathsf{se}} \quad (9)$$

where Q is the query tape of the adversary, holds for a negligible function ϵ_{se}.

Recall that simulation soundness only guarantees that x is a true statement whereas simulation extractability additionally guarantees that the witness be efficiently extractable. When the number of oracle access is unlimited (limited to only once, resp.), it is called unbounded (one-time, resp.) simulation extractability.

We show that the simulation-sound NIZK of [27] is simulation extractable if the underlying NIZK is a proof of knowledge system. Let $\mathsf{POK} = \mathsf{POK}.\{\mathsf{Crs}, \mathsf{Prv}, \mathsf{Vrf}, \mathsf{Ext}\}$ be a NIZK proof of knowledge system, $\mathsf{SIG} = \mathsf{SIG}.\{\mathsf{Key}, \mathsf{Sign}, \mathsf{Vrf}\}$ be a signature scheme, and $\mathsf{OTS} = \mathsf{OTS}.\{\mathsf{Key}, \mathsf{Sign}, \mathsf{Vrf}\}$ be a one-time signature scheme. Their construction of $\mathsf{SE\text{-}NIZK} = \mathsf{SE\text{-}NIZK}.\{\mathsf{Crs}, \mathsf{Prv}, \mathsf{Vrf}, \mathsf{PrvSim}, \mathsf{Ext}\}$ is shown in Fig. 3

Theorem 6. *If* POK *is a witness indistinguishable proof of knowledge system with knowledge-soundness error* ϵ_{ks}*,* SIG *is unforgeable against non-adaptive chosen message attacks with advantage* ϵ_{sig}*, and* OTS *is strongly one-time unforgeable against chosen message attacks with advantage* ϵ_{ots}*, then* $\mathsf{SE\text{-}NIZK}$ *is strongly simulation-extractable NIZK with simulation-extraction error* $\epsilon_{\mathsf{se}} \leq \epsilon_{\mathsf{ots}} + \epsilon_{\mathsf{ks}} + \epsilon_{\mathsf{sig}}$.

Proof. Correctness of the scheme and zero-knowledge property is verified by inspecting the construction. Computational zero-knowledge is not hard to verify due to the witness indistinguishability of POK and the construction of $\mathsf{SE\text{-}NIZK.PrvSim}$.

We focus on showing simulation extractability. Suppose that adversary $\mathcal{A}$ accesses $\mathsf{SE\text{-}NIZK.PrvSim}(crs, \tau_{\mathsf{zk}}, \cdot)$ as an oracle and eventually outputs $x^\star$ and $\pi^\star = (\pi^\star, opk^\star, \sigma^\star)$ that passes $\mathsf{SE\text{-}NIZK.Vrf}$. For $\mathcal{A}$ to be successful, it must be the case that $(x^\star, \pi^\star) \notin \{x_i, \pi_i\}$ and $(x^\star, \pi^\star) \notin R$. Recall that $\pi^\star_{\mathsf{se}} = (\pi^\star, opk^\star, \sigma^\star)$. We distinguish two cases:

Case 1: $opk^\star = opk_i$ happens for opk_i returned from the oracle. In this case, $(x^\star, \pi^\star, \sigma^\star) \neq (x_i, \pi_i, \sigma_i)$ and we have a valid forgery for OTS. This happens with probability at most ϵ_{ots} due to the strong one-time unforgeability of OTS.

Case 2: $opk^\star \neq opk_i$ for all opk_i. By executing $\mathsf{SE\text{-}NIZK.Ext}(crs, \tau_{\mathsf{ex}}, x^\star, \pi^\star)$, we have $w_{\mathsf{se}} = (w, \sigma)$ that either $R(x^\star, w) = 1$ or $\mathsf{SIG.Vrf}(vk, \sigma, opk^\star) = 1$ for

[Scheme $\mathsf{SE\text{-}NIZK}$**]**

$\mathsf{SE\text{-}NIZK.Crs}(gk)$: It takes gk and runs $(crs_{\mathsf{pok}}, \tau_{\mathsf{ex}}) \leftarrow \mathsf{POK.Crs}(gk)$, $(vk, sk) \leftarrow \mathsf{SIG.Key}(gk)$. It then outputs $crs := (gk, crs_{\mathsf{pok}}, vk)$, $\tau_{\mathsf{ex}} := \tau_{\mathsf{ex}}$, and $\tau_{\mathsf{zk}} := sk$.

$\mathsf{SE\text{-}NIZK.Prv}(crs, x, w)$: Run $opk \leftarrow \mathsf{OTS.Key}(gk)$. Set $\sigma = \bot$. Let $x_{\mathsf{se}} := (x, opk)$ and $w_{\mathsf{se}} := (w, \sigma)$. Set relation R_{se} be

$$R_{\mathsf{se}}(x_{\mathsf{se}}, w_{\mathsf{se}}) := (R(x, w) = 1) \vee (\mathsf{SIG.Vrf}(vk, \sigma, opk) = 1)\,.$$

Run $\pi \leftarrow \mathsf{POK.Prv}(crs_{\mathsf{pok}}, x_{\mathsf{se}}, w_{\mathsf{se}})$, and $\sigma_o \leftarrow \mathsf{OTS.Sign}(osk, \pi)$. Output $\pi_{\mathsf{se}} := (\pi, opk, \sigma_o)$.

$\mathsf{SE\text{-}NIZK.Vrf}(crs, x, \pi_{\mathsf{se}})$: Parse $(\pi, opk, \sigma_o) \leftarrow \pi_{\mathsf{se}}$. Verify both σ_o and π.

$\mathsf{SE\text{-}NIZK.PrvSim}(crs, \tau_{\mathsf{zk}}, x)$: Parse $(gk, crs_{\mathsf{pok}}, vk) \leftarrow crs$ and $sk \leftarrow \tau_{\mathsf{zk}}$. Run $opk \leftarrow \mathsf{OTS.Key}(gk)$ and $\sigma \leftarrow \mathsf{SIG.Sign}(sk, opk)$. Set $w_{\mathsf{se}} := (\bot, \sigma)$. Run $\pi \leftarrow \mathsf{POK.Prv}(crs_{\mathsf{pok}}, x_{\mathsf{se}}, w_{\mathsf{se}})$ and $\sigma_o \leftarrow \mathsf{OTS.Sign}(osk, \pi)$. Output $\pi_{\mathsf{se}} := (\pi, opk, \sigma_o)$.

$\mathsf{SE\text{-}NIZK.Ext}(crs, \tau_{\mathsf{ex}}, x, \pi_{\mathsf{se}})$: Parse $(gk, crs_{\mathsf{pok}}, vk) \leftarrow crs$ and $(\pi, opk, \sigma_o) \leftarrow \pi_{\mathsf{se}}$. Run $w_{\mathsf{se}} \leftarrow \mathsf{POK.Ext}(crs_{\mathsf{pok}}, \tau_{\mathsf{ex}}, \pi, (x, opk))$ and return w in $w_{\mathsf{se}} = (w, \sigma)$.

Fig. 3. Simulation-Extractable Non-Interactive Zero-Knowledge Proof System.

vk included in crs. The extraction is successful with probability $1 - \epsilon_{\mathsf{ks}}$ due to the knowledge-soundness of NIZK. Then, if the former happens, we have extracted correct witness for $x^\star$ and $\mathcal{A}$ is unsuccessful. Otherwise, we have a valid forgery for SIG since its message $opk^\star$ is fresh. This happens with probability at most ϵ_{sig} due to the unforgeability against non-adaptive chosen-message attacks for SIG. (The non-adaptiveness is due to the fact that all opk_i can be generated in advance.)

In total, the extraction is successful with probability $(1 - \epsilon_{\mathsf{se}}) = (1 - \epsilon_{\mathsf{ots}})(1 - \epsilon_{\mathsf{ks}})(1 - \epsilon_{\mathsf{sig}})$ which leads to $\epsilon_{\mathsf{se}} \leq \epsilon_{\mathsf{ots}} + \epsilon_{\mathsf{ks}} + \epsilon_{\mathsf{sig}}$ as stated. $\square$

Instantiating SE-NIZK. We instantiate the above generic SE-NIZK in several ways. The result is several SE-NIZKs that have different sets of properties as summarised in Table 5.1.

- $\mathsf{SE\text{-}NIZK0}$**:** The original instantiation in [27]. SIG is a tree-based signature scheme with their original one-time signature scheme, and OTS is instantiated with the Pedersen commitment as a one-time signature that is not structure-preserving. The result is a unbounded SE-NIZK.
- $\mathsf{SE\text{-}NIZK1}$**:** SIG remains a tree-based scheme but we replace the internal one-time signatures with our TOS in plug-in manner. The result is a more efficient unbounded SE-NIZK. This shows how plug-in replacement of low-level building block impacts to the efficiency.
- $\mathsf{SE\text{-}NIZK2}$**:** The same as $\mathsf{SE\text{-}NIZK1}$ but we instantiate OTS with our TOS as well. Since that OTS is the only non-structure-preserving component in $\mathsf{SE\text{-}NIZK1}$, the

result is structure-preserving unbounded SE-NIZK. A problem is that the TOS must be able to sign the entire proof that linearly grows in the size of the public-key of the TOS itself. We therefore use the technique of chaining the signatures as mentioned in Remark 2 in Section 3. The same technique is used when the one-time key is signed at the bottom of the tree-based signing. The resulting SE-NIZK is used in constructing structure-preserving publicly verifiable CCA-secure PKE tightly-secure with multiple challenges.

SE-NIZK3: We use TOS for SIG. No tree-based construction here. This means the signature can be generated only once for simulation and the result is structure-preserving one-time SE-NIZK. As well as SE-NIZK2, we use the signatrue chaining. The resulting scheme can be used in constructing efficient structure-preserving publicly verifiable CCA-secure PKE. We can add leakage resilience (LR) if desired.

SE-NIZK4: As well as SE-NIZK3 we instantiate SIG with our TOS but leave OTS with the one based on the Pedersen commitment for the sake of efficiency. In exchange of losing structure-preservation, it results in a very efficient one-time SE-NIZK. It will be used for publicly verifiable CCA-secure PKE (with LR if desired).

scheme	efficiency	simulatability	structure-preservation
SE-NIZK0	less efficient	unbounded	no
SE-NIZK1	moderate	unbounded	no
SE-NIZK2	less efficient	unbounded	yes
SE-NIZK3	efficient	one-time	yes
SE-NIZK4	very efficient	one-time	no

Table 2. Properties of the instantiations of SE-NIZK. Efficiency is presented in subjective term. Objective evaluation of efficiency is in Table 3.

We give a general formula that evaluate the cost of the generic construction. The generic SE-NIZK uses the S_0-or-S_1 structure so that real proof is done for statement S_0 whereas simulation is done with a witness for statement S_1. It is however believed that the OR structure with Groth-Sahai proof system is as costly as doubling the number of elements in a proof. It is true for general statements. But for the specific construction of SE-NIZK, it can be done much less costly. taking the advantage of the fact that there is no common witnesses shared by statements S_0 and S_1.

Regarding the proof of disjunction, we sketch the construction of [11] and refer to [11] for details. The prover commits to $1_{\mathbb{G}}$ or G with X, and show its correctness by proving a single non-linear relation $e(X, X) = e(X, G)$. We call X a switcher as it switches the statement that is really proven. Let $X_0 = X$ and $X_1 = G \cdot X^{-1}$. Then for every pairing product equation in S_b, if pairing $e(A, B)$ with some constants A and B is involved, one of them say A is transformed to variable Y and prove its correctness by showing $e(Y, G) = e(A, X_b)$ holds. (Observe that if $X_b = G$, it guarantees that $Y = A$. Otherwise, if $X_b = 1$, it holds for $Y = 1$.) After that, every pairing in every relation in S_b includes at least one variable. Now, if $X_b = G$, one can still satisfy the relations with the legitimate witnesses. Otherwise, if $X_b = 1_{\mathbb{G}}$, they can be satisfied by setting $1_{\mathbb{G}}$ to all variables, which allows zero-knowledge simulation.

Now the number of group elements in a proof of SE-NIZK is counted as follows. Let $S_0 : (R(x, w) = 1)$ and $S_1 : (\mathsf{SIG.Vrf}(vk, \sigma, opk) = 1)$ be the statements represented by pairing product equations. The proof size of SE-NIZK is as follows:

$$\begin{aligned}
&(\text{cost for } S_0) + (\text{cost for switcher}) + (\text{cost for } S_1) + (\text{cost for } \mathsf{OTS}) \\
&= (\text{cost for } S_0) && (10) \\
&+ (|com| \times 1 + |\pi_{NL}| \times 1) && (11) \\
&+ (|com| \times (|\sigma_{\mathsf{sig}}| + S_1(C)) + |\pi_L| \times (S_1(L) + S_1(C)) + |\pi_{NL}| \times S_1(NL) && (12) \\
&+ (|opk_o| + |\sigma_o|) && (13)
\end{aligned}$$

Here, parameters $|\pi_{L/NL}|$, $|opk_o|$, $|\sigma_o|$, $|\sigma_{\mathsf{sig}}|$, $|com|$ are the size of a proof for a linear/non-linear relation, a one-time public-key of OTS, a signature of OTS, a signature of SIG, and commitment per variable, respectively. Also, $S_1(L/NL)$ and $S_1(C)$, denote the number of linear/non-linear relations and constant pairings, respectively, in $\mathsf{SIG.Vrf}$ where signatures are considered as variables. By "overhead", we mean the size of (11)+(12)+ (13) since it is the cost for achieving simulation extractability on top of simply proving the original statement S_0.

With the Groth-Sahai proof over the DLIN setting, we have $(|com|, |\pi_L|, |\pi_{NL}|) = (3, 3, 9)$. Other parameters $(|\sigma_{\mathsf{sig}}|, S_1(C), S_1(L), S_1(NL), |opk_o|, |\sigma_o|)$ differ in every instantiation and summarised as in Table 3. For $\mathsf{SE\text{-}NIZK2,3}$ that uses the signature chaining, let k_1 and k_2 be block size of a message for SIG and OTS, respectively. Also let τ_1 and τ_2 be the length of the chains determined by $\tau_2 := 1 + \max(0, \lceil \frac{m_2 - k_2}{k_2 - 1} \rceil)$ and $\tau_1 := 1 + \max(0, \lceil \frac{m_1 - k_1}{k_1 - 1} \rceil)$ where $m_2 := |opk_o|$ and $m_1 := (\text{cost for } S_0) + (\text{cost for switcher}) + (\text{cost for } S_0)$. Then the overhead in a proof is $\psi_2 := 21d + 18\tau_2 + 4\tau_1 + 2k_1 + 12k_2 + 44$ for $\mathsf{SE\text{-}NIZK2}$ and $\psi_3 := 18\tau_2 + 4\tau_1 + 2k_1 + 12k_1 + 44$ for $\mathsf{SE\text{-}NIZK3}$. When those schemes are used, parameters k_1 and k_2 should be chosen to minimize the overhead. Unfortunately, the general assessment in Table 3 is not intuitive enough to see the difference of efficiency due to the several parameters involved. One can see their difference in more concrete manner in the next section.

scheme	$\lvert\sigma_{\mathsf{sig}}\rvert$	$S_1(C)$	$S_1(L)$	$S_1(NL)$	$\lvert opk_o\rvert$	$\lvert\sigma_o\rvert$	overhead
SE-NIZK0	$10d + 2$	3	3	$3d$	2	2	$(57d + 64)_{\mathbb{G}} + 9_{\mathbb{Z}_p}$
SE-NIZK1	$5d + 3$	3	$2d$	0	2	2	$(21d + 64)_{\mathbb{G}} + 9_{\mathbb{Z}_p}$
SE-NIZK2	$5d + 4\tau_2 - 1$	$2k_2 + 1$	$2(d + \tau_2)$	0	$2k_1 + 6$	$4\tau_1 - 1$	ψ_2
SE-NIZK3	$4\tau_2 - 1$	$2k_2 + 1$	$2\tau_2$	0	$2k_1 + 6$	$4\tau_1 - 1$	ψ_3
SE-NIZK4	3	3	2	0	2	2	$64_{\mathbb{G}} + 6_{\mathbb{Z}_p}$

Table 3. Parameterized costs for simulation extractable NIZKs. See the main text for the meaning of parameters.

We note that the instantiations follow the generic construction rigorously. Some hand-crafted optimization is possible in reality by carefully choosing variables and constants in GS-proofs. In particular, it is not necessary to commit the entire signature when we compute π. The tag and Z in every signature can be sent in the clear. Such optimization saves considerable number of group elements. The impact of optimization will be discussed in the next section with concrete numbers.

5.2 Tight/Structure-Preserving CCA-Secure Encryption from SE-NIZK

In [27], the SS-NIZK is used to construct a chosen-ciphertext-secure (CCA) PKE that is secure against multiple challenges retaining the tightness property. It follows the Naor-Yung paradigm that combines *two* chosen-plaintext-secure public-key encryption schemes (CPA-secure PKE) with an SS-NIZK. As we now know that their instantiation of SS-NIZK actually gives SE-NIZK, we rather follow more efficient generic construction by Dodis, et. al.,[18] that combines *one* CPA-secure PKE with SE-NIZK. This results in more efficient CCA PKE. Since slightly different components is used in [18] for their purpose of adding leakage resilience and no quantified evaluation was presented, we restate their theorem in a simplified form with a proof in the following.

Let CPA be a CPA-secure encryption scheme and $\mathsf{SE\text{-}NIZK}$ be simulation extractable NIZK. We construct CCA-secure encryption scheme $\mathsf{PKE} := \mathsf{PKE}.\{\mathsf{Key}, \mathsf{Enc}, \mathsf{Dec}\}$ by combining CPA and $\mathsf{SE\text{-}NIZK}$ as shown in Fig. 4. Let gk be a common parameter generated by $\mathsf{Setup}(1^\lambda)$. Underlying encryption scheme CPA must satisfy the following property. There exists efficiently computable function W and efficiently verifiable relation R such that

$$(R((ek_{\mathsf{cpa}}, c_{\mathsf{cpa}}), (msg, W(r)) = 1) \iff (c_{\mathsf{cpa}} = \mathsf{CPA}.\mathsf{Enc}(ek_{\mathsf{cpa}}, msg; r)). \quad (14)$$

Function W is understood as a converter that transforms random coin r into a form that is easily handled in verifying relation R. In our instantiation with Groth-Sahai proof system, W transforms $r \in \mathbb{Z}_p$ to a vector of group elements.

[Scheme PKE**]**

$\mathsf{PKE}.\mathsf{Key}(gk)$**:** Run $(crs_{\mathsf{nizk}}, \tau_{\mathsf{zk}}, \tau_{\mathsf{ex}}) \leftarrow \mathsf{SE\text{-}NIZK}.\mathsf{Crs}(gk)$, $(ek_{\mathsf{cpa}}, dk_{\mathsf{cpa}}) \leftarrow \mathsf{CPA}.\mathsf{Key}(gk)$. Set $ek := (crs_{\mathsf{nizk}}, ek_{\mathsf{cpa}})$ and $dk := dk_{\mathsf{cpa}}$.

$\mathsf{PKE}.\mathsf{Enc}(ek, msg)$**:** Run $c_{\mathsf{cpa}} \leftarrow \mathsf{CPA}.\mathsf{Enc}(ek_{\mathsf{cpa}}, msg; r)$ and $\pi \leftarrow \mathsf{SE\text{-}NIZK}.\mathsf{Prv}(crs_{\mathsf{nizk}}, c_{\mathsf{cpa}}, (msg, r))$. The proof is for relation $1 = R((ek_{\mathsf{cpa}}, c_{\mathsf{cpa}}), (msg, r))$. Output ciphertext $c := (c_{\mathsf{cpa}}, \pi)$.

$\mathsf{PKE}.\mathsf{Dec}(dk, c)$**:** Parse c into (c_{cpa}, π). If $0 \leftarrow \mathsf{SE\text{-}NIZK}.\mathsf{Vrf}(crs_{\mathsf{nizk}}, c_{\mathsf{cpa}}, \pi)$, return $\perp$. Otherwise, output $msg := \mathsf{CPA}.\mathsf{Dec}(dk_{\mathsf{cpa}}, c_{\mathsf{cpa}})$.

Fig. 4. CCA-secure PKE from SE-NIZK.

In addition to the use of only one CPA-secure encryption, the construction in Fig. 4 is different from [27] in the following sense. In [27], crs_{nizk} is included in gk and common for all users. Hence the security of resulting CCA PKE fully relies on the secrecy of the trapdoors behind crs_{nizk}. In our case, fresh crs_{nizk} is selected for every public-key. If gk includes no trapdoors (as is usually the case in the certified group model where only the group description Λ is included in gk), the security of the resulting CCA PKE is reduced to complexity assumptions defined over gk. In fact, when gk does not include trapdoors, and the underlying complexity assumption is random self reducible, it is rather trivial to preserve tightness when extending the security reduction from the single-user to the multi-user setting because no secret information is shared between users. On the contrary, it is not trivial to preserve tightness in the multi-challenge setting since every challenge is related to the same public-key which involves a trapdoor. We therefore focus on security in the multi-challenge and single-user setting in the following argument.

Theorem 7. *If* CPA *is left-or-right CPA secure encryption scheme with advantage* ϵ_{cpa} *and* $\mathsf{SE\text{-}NIZK}$ *be unbounded (or one-time, resp.) simulation-extractable NIZK with zero-knowledge error* ϵ_{zk} *and simulation-extraction error* ϵ_{se}*, then* PKE *is multi-challenge (or standard single-challenge, resp.) CCA-secure with advantage* $\epsilon_{\mathsf{cca}} \leq 2 \cdot (\epsilon_{\mathsf{zk}} + \epsilon_{\mathsf{se}}) + \epsilon_{\mathsf{cpa}}$.

Proof. The proof structure follows [18]. Games are numbered by the combination of an idealization step counter and a bit indicating whether to encrypt the left or the right side to visualize its inherent symmetry.

Game 0.0. This is the IND-CCA security experiment from [27], executed with $b = 0$. The challenger always returns encryptions of msg_0.

Game 1.0. This game is identical to Game 0.0, except that we use the zero-knowledge simulator of $\mathsf{SE\text{-}NIZK}$ to generate proofs in the challenge ciphertexts. (If $\mathsf{SE\text{-}NIZK}$ is one-time simulation extractable, this is limited to a single challenge.) We have $[Pr[Win_{1.0}] - Pr[Win_{0.0}] \leq \epsilon_{\mathsf{zk}}$.

Game 2.0 This game is identical to Game 1.0, except that decryption queries $c = (c_{\mathsf{cpa}}, \pi)$ are answered by running $\mathsf{SE\text{-}NIZK.Ext}$ on π to extract msg. (This modification accommodates with the previous one since $\mathsf{SE\text{-}NIZK.Crs}$ outputs trapdoors for simulation and extraction at the same time.) We have $\Pr[Win_{2.0}] - \Pr[Win_{1.0}] \leq \epsilon_{\mathsf{se}}$.

Game 2.1 This game is identical to Game 2.0, except that the challenger always returns encryptions of msg_1. As we do not use dk_{cpa} anywhere we can do a reduction to IND-CPA security and have $\Pr[Win_{2.1}] - \Pr[Win_{2.0}] \leq \epsilon_{\mathsf{cpa}}$.

Game 1.1. This game is identical to Game 2.1, except that decryption queries $c = (c_{\mathsf{cpa}}, \pi)$ are no longer answered by running the extractor but by decrypting c_{cpa} to obtain msg. We have $\Pr[Win_{1.1}] - \Pr[Win_{2.1}] \leq \epsilon_{\mathsf{se}}$.

Game 0.1. This game is identical to Game 1.1, except that we no longer use the zero-knowledge simulator of $\mathsf{SE\text{-}NIZK}$ to generate all proofs but generate them honestly. We have $\Pr[Win_{0.1}] - \Pr[Win_{1.1}] \leq \epsilon_{\mathsf{zk}}$. This is the IND-CCA security experiment executed with $b = 1$.

By accumulating the differences, we have $\epsilon_{\mathsf{cca}} \leq 2 \cdot (\epsilon_{\mathsf{zk}} + \epsilon_{\mathsf{se}}) + \epsilon_{\mathsf{cpa}}$ as stated. □

We instantiate CPA with the linear encryption scheme [28, 30] shown in Fig. 5. It is IND-CPA secure and tightly reducible to DLIN in the multi-challenge and multi-user setting as formally proven in [27]. Well formness of a ciphertext can be proven by providing a GS proof for relations

$$e(C_1, \underline{G}) = e(G_1, \underline{W_1}), \qquad e(C_2, \underline{G}) = e(G_2, \underline{W_2}),$$
$$e(\underline{W_1}\,\underline{W_2}, G) = e(C_3/\underline{M}, G), \qquad e(\underline{G}, G) = e(G, \underline{X_0}).$$

The underlined variables $G, W_1 := G^{r_1}, W_2 := G^{r_2}, M$ are witneses and X_0 is a switcher as explained in Section 5.1. Accordingly, the "cost for S_0" in (10) is 24 group elements (12 for four commitments and 12 for proof of four linear equations).

[Scheme CPA**]**
Let gk include $\Lambda = (p, \mathbb{G}, \mathbb{G}_T, e)$ and generator $G \in \mathbb{G}$ as global parameters.

CPA.Key(gk)**:** Uniformly select y_1, y_2 from $\mathbb{Z}_p^*$. Compute $G_1 = G^{y_1}$ and $G_2 = G^{y_2}$, And then output $ek := (\Lambda, G_1, G_2)$ and $dk := (ek, y_1, y_2)$. The message space is $\mathbb{G}$.

CPA.Enc(ek, msg)**:** Parse msg into $M \in \mathbb{G}$ and ek accordingly. Pick random $r_1, r_2 \in \mathbb{Z}_p$. Compute and output signature $c := (C_1, C_2, C_3)$ where $C_1 := G_1^{r_1}$, $C_2 := G_2^{r_2}$, and $C_3 := M\,G^{r_1+r_2}$.

CPA.Dec(dk, c)**:** Parse c into (C_1, C_2, C_3), and dk into (y_1, y_2). Then output $M := C_3\,C_1^{-1/y_1}\,C_2^{-1/y_2}$.

Fig. 5. The Linear Encryption Scheme.

The SE-NIZK in the construction of PKE can be instantiated with any SE-NIZKi in Section 5.1. The efficiency and properties of the resulting PKE is shown in Table 4 [1]. For SE-NIZK1,2,3 that uses a tree-based signature scheme, we set the depth of the tree to $d = 20$, which allows up to 2^{20} simulations. (If one demands virtually unbounded simulatability, d should equal to the security parameter as suggested in [27].) For SE-NIZK3,4 that uses TOS as OTS, we seek for optimal value for parameter k_1 and k_2 that minimizes the size of the cipehrtext. As originally stated in [18], leakage resilience can be added by using a leakage resilient CPA encryption from [18] while retaining other properties.

[1] The figures in Table 4 are slightly different from those in the conference version of this paper [2]. We use an updated version of GS-proof system that includes elements of $\mathbb{Z}_p$ in their proof (when possible), and fix some counting errors found in the previous assessment.

$\mathsf{SE\text{-}NIZK}i$	Cipheretxt Size Size	Properties			Parameter Setting
		Publicly-Verifiable	Tightly-Secure	Strucure-Preserving	
0	$1207\,\mathbb{G} + 9\,\mathbb{Z}_p$	yes	yes	no	d=20
1	$487\,\mathbb{G} + 9\,\mathbb{Z}_p$	yes	yes	no	d=20
2	$861\,\mathbb{G}$	yes	yes	yes	d=20, k_1=20, k_2=10
3	$321\,\mathbb{G}$	yes	no	yes	k_1=8, k_2=8
4	$67\,\mathbb{G} + 6\,\mathbb{Z}_p$	yes	no	no	

Table 4. Properties and ciphertext size of CCA PKE constructed with $\mathsf{SE\text{-}NIZK}i$.

We finally remark that the ciphertext size is assessed with non-optimized instantiations of $\mathsf{SE\text{-}NIZK}i$. Following the already mentioned observation that only a part of a simulated signature in NIZK must be committed, one can optimize the GS proofs and slightly reduce the size of ciphertext.

6 Conclusion

We present a new efficient tagged one-time signature scheme that features tight reduction to DLIN and optimal tag size. We then revisit several generic constructions where (tagged) one-time signatures play a central role, and build structure preserving signature and public-key encryption schemes that for the first time simultaneously achieve several desirable properties. Although many of our instantiations are not necessarily practical with hundreds of group elements, the concrete efficiency assessment should serve as a reference for comparison for more efficient non-generic construction.

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 2012*, volume 7658 of *LNCS*, pages 4–24. Springer, 2012.
2. M. Abe, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Tagged One-Time Signatures: Tight Security and Optimal Tag Size. In *PKC 2013*, volume 7778 of *LNCS*, pages 312–332. Springer, 2013.
3. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In *CRYPTO 2010*, volume 6223 of *LNCS*, pages 209–237. Springer, 2010.
4. M. Abe, J. Groth, K. Haralambiev, and M. Ohkubo. Optimal structure-prserving signatures in asymmetric bilinear groups. In *CRYPTO 2011*, volume 6841 of *LNCS*. pages 649–666. Springer, 2011.
5. M. Abe, K. Haralambiev, and M. Ohkubo. Signing on group elements for modular protocol designs. IACR ePrint Archive, 2010/133, 2010. `http://eprint.iacr.org`.
6. M. Abe, K. Haralambiev, and M. Ohkubo. Group to group commitments do not shrink. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 301–317. Springer, 2012.
7. N. Attrapadung, B. Libert, and T. Peters. Computing on authenticated data: New privacy definitions and constructions. In *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 367–385. Springer, 2012.

8. M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. Randomizable proofs and delegatable anonymous credentials. In *CRYPTO 2009*, volume 5677 of *LNCS*, pages 108–125. Springer, 2009.
9. M. Bellare and S. Shoup. Two-tier signatures, strongly unforgeable signatures, and Fiat-Shamir without random oracles. In *PKC 2007*, volume 4450 of *LNCS*, pages 201–216. Springer, 2007.
10. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.
11. J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 351–368. Springer, 2009.
12. J. Camenisch, M. Dubovitskaya, and K. Haralambiev. Efficient structure-preserving signature scheme from standard assumptions. In *SCN 2012*, volume 7485 of *LNCS*, pages 76–94. Springer, 2012.
13. J. Camenisch, K. Haralambiev, M. Kohlweiss, J. Lapon, and V. Naessens. Structure preserving CCA secure encryption and applications. In *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 89–106. Springer, 2011.
14. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer, 2004.
15. J. Cathalo, B. Libert, and M. Yung. Group encryption: Non-interactive realization in the standard model. In *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 179–196. Springer, 2009.
16. M. Chase and M. Kohlweiss. A new hash-and-sign approach and structure-preserving signatures from DLIN. In *SCN 2012* , volume 7485 of *LNCS*, pages 131–148. Springer, 2012.
17. M. Chase, M. Kohlweiss, A. Lysyanskaya, and S. Meiklejohn. Malleable proof systems and applications. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 281–300. Springer, 2012.
18. Y. Dodis, K. Haralambiev, A. López-Alt, and D. Wichs. Efficient public-key cryptography in the presence of key leakage. In *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 613–631, 2010.
19. G. Fuchsbauer and D. Pointcheval. Anonymous proxy signatures. In *SCN 2008* , volume 5229 of *LNCS*, pages 201–217. Springer, 2008.
20. G. Fuchsbauer, D. Pointcheval, and D. Vergnaud. Transferable constant-size fair e-cash. In *CANS 2009*, volume 5888 of *LNCS*, pages 226–247. Springer, 2009.
21. S. D. Galbraith, K. G. Peterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.
22. M. Green and S. Hohenberger. Universally composable adaptive oblivious transfer. In *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 179–197, Springer, 2008.
23. M. Green and S. Hohenberger. Practical adaptive oblivious transfer from simple assumptions. In *TCC 2011*, volume 6597 of *LNCS*, pages 347–363. Springer, 2011.
24. J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, 2006.
25. J. Groth. Fully anonymous group signatures without random oracles. In *Asiacrypt 2007*, volume 4833 of *LNCS*, pages 164–180. Springer, 2007.
26. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, 2008.
27. D. Hofheinz and T. Jager. Tightly secure signatures and public-key encryption. In *CRYPTO 2012*, volume 7417 of *LNCS*, pages 590–607. Springer, 2012.
28. D. Hofheinz and E. Kiltz. Secure hybrid encryption from weakened key encapsulation. In *CRYPTO 2007*, volume 4622 of *LNCS*, pages 553–571. Springer, 2007.
29. A. D. Santis, G. D. Crescenzo, R. Ostrovsky, G. Persiano, and A. Sahai. Robust non-interactive zero knowledge. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 566–598. Springer, 2001.
30. H. Shacham. A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants. IACR ePrint Archive, 2007/074, 2007. `http://eprint.iacr.org`.