Structure-Preserving Signatures and Commitments to Group Elements

Masayuki Abe[1], Georg Fuchsbauer[2], Jens Groth[3], Kristiyan Haralambiev[4] *, and Miyako Ohkubo[5] *

[1] Information Sharing Platform Laboratories, NTT Corporation, Japan
abe.masyuki@lab.ntt.co.jp
[2] École normale supérieure, CNRS - INRIA, Paris, France
http://www.di.ens.fr/∼fuchsbau
[3] University College London, UK
j.groth@ucl.ac.uk
[4] Computer Science Department, New York University, USA
kkh@cs.nyu.edu
[5] National Institute of Information and Communications Technology, Japan
m.ohkubo@nict.go.jp

Abstract. A modular approach for cryptographic protocols leads to a simple design but often inefficient constructions. On the other hand, ad hoc constructions may yield efficient protocols at the cost of losing conceptual simplicity. We suggest *structure-preserving* commitments and signatures to overcome this dilemma and provide a way to construct modular protocols with reasonable efficiency, while retaining conceptual simplicity.

We focus on schemes in bilinear groups that preserve parts of the group structure, which makes it easy to combine them with other primitives such as non-interactive zero-knowledge proofs for bilinear groups.

We say that a signature scheme is *structure-preserving* if its verification keys, signatures, and messages are elements in a bilinear group, and the verification equation is a conjunction of pairing-product equations. If moreover the verification keys lie in the message space, we call them *automorphic*. We present several efficient instantiations of automorphic and structure-preserving signatures, enjoying various other additional properties, such as *simulatability*. Among many applications, we give three examples: adaptively secure round-optimal blind signature schemes, a group signature scheme with efficient concurrent join, and an efficient instantiation of anonymous proxy signatures.

A further contribution is *homomorphic trapdoor commitments to group elements* which are also length reducing. In contrast, the messages of previous homomorphic trapdoor commitment schemes are exponents.

1 Introduction

The designer of cryptographic protocols faces a tension between choosing a modular approach using generic primitives that lead to a simple design but inefficient

* Work done while at NTT Information Sharing Platform Laboratories.

protocols or using ad hoc constructions that sometimes yield efficient protocols at the cost of losing conceptual simplicity. Cryptographic protocols often combine general building blocks such as commitments, encryption, signatures, and zero-knowledge proofs. While modular design is useful to show feasibility of cryptographic tasks and also to illustrate a comprehensible framework, efficient instantiations are sometimes left as a next challenge. Some cryptographic tasks find "cleverly crafted" efficient solutions dedicated to their specific purposes. Nevertheless, modular construction makes implementing more complex primitives easier when the building blocks have reasonable instantiations. We suggest *structure-preserving* commitments and signatures to provide a way to construct modular protocols that retain conceptual simplicity and at the same time yield reasonable efficiency.

A classical way of realizing efficient instantiations is to rely on the random-oracle heuristic [BR93] for non-interactive zero-knowledge (NIZK) proofs—or to directly use *interactive* assumptions (like the LRSW assumption [LRSW00] and its variants, or "one-more" assumptions [BNPS03]). Due to a series of criticisms starting with [CGH98] more and more practical schemes are being proposed and proved secure in the *standard model* (i.e., without random oracles) and under *falsifiable* (and thus non-interactive) assumptions [Nao03]. All schemes given in this work satisfy these criteria.

Structure-Preserving Signatures. The combination of NIZK proofs of knowledge and signatures appears frequently in privacy-protecting cryptographic protocols such as group signatures [BMW03,KY05,BSZ05,Gro07], blind signatures [Fis06,AO09], anonymous credentials [BCKL08,BCC+09], verifiably encrypted signatures [BGLS03,RS09], non-interactive group encryption [CLY09] and many more.

An efficient non-interactive proof system in the standard model, however, has been absent until recently. In [GS08], Groth and Sahai presented the first (and currently the only) efficient non-interactive proof system for a large class of statements over bilinear groups. The most interesting and widely used type is a conjunction of pairing-product equations (PPE) whose variables are elements of the bilinear group (cf. Section 2.4). A PPE consists of products of pairings applied to the variables and constants from the group. For this type of equations, the proofs are fully extractable which actually makes them proofs of knowledge. This renders GS proofs particularly interesting for modular protocol design.

Research on signature schemes that are compatible with GS proofs was initiated in [Gro06]. While the design goal is clear and simple, giving an efficient instantiation has proved hard for years. There are efficient signature schemes, e.g., [BB04,CL04,BCKL08,CKS09], whose verification predicates are pairing-product equations, but none of them have signatures *and* messages that exclusively consist of group elements. Since only group elements can be extracted from GS commitments, this entailed limited applicability of each scheme or stronger security notions such as *F-unforgeability* [BCKL08].

The desirable properties of a signature scheme enabling modular design together with GS proofs are the following:

1. the scheme is unforgeable against chosen-message attacks;
2. the verification keys, messages, and signatures are elements of a bilinear group; and
3. the verification predicate is a conjunction of pairing-product equations over the key, the message and the signature.

Note that this proscribes the use of hash functions, which usually play a central role in making signature schemes unforgeable against adaptive chosen-message attacks. We therefore call such a scheme *structure preserving*. If in addition its verification keys lie in the message space, we call it an *automorphic signature* (since it signs its *own* keys besides preserving *structure*).

Combined with GS proofs, structure-preserving signatures allow to prove knowledge of messages, signatures and/or verification keys without actually revealing them. Proving knowledge of signatures has been used in many construction of group signatures, anonymous proxy signatures, anonymous credentials, blind signatures, and others. Clearly, structure-preserving signatures combined with the GS proof system will allow to instantiate those constructions without resorting to interactive assumptions nor to the random-oracle model while maintaining a modular design.

For example, Fischlin [Fis06] presented the following framework for round-optimal blind signatures in the common reference string model. To obtain a signature from the signer, the user commits to a message and sends the commitment to the signer. Then, the signer signs the commitment and sends back the signature. The user produces a NIZK proof of knowledge of a commitment, an opening of the commitment to that message, and a signature on the commitment. This proof constitutes a blind signature for the message. Despite its simplicity, the scheme has not been instantiated efficiently in the standard model because it requires a signature scheme which signs trapdoor commitments and whose verification equations should mesh well with the GS proof system.

An application that also requires signing verification keys are *anonymous proxy signatures* [FP08]. They enable users to delegate (and redelegate) their signing rights to other users. A signature on behalf of another user (proxy signature) hides the identity of the proxy signer and possible intermediate delegators. Instantiating anonymous proxy signatures requires a signature scheme that is both GS compatible and enables users to sign other user's verification keys to delegate. Automorphic signatures can thus be used create a *delegation chain* of which the proxy signer proves knowledge using GS proofs.

Trapdoor Commitments to Group Elements. A non-interactive commitment scheme allows to create a *commitment* c to a message m. The commitment *hides* the message, but we may later disclose m and demonstrate that c was a commitment to m by revealing the randomness r used when creating it. This is called *opening* the commitment. It is essential that once a commitment is made, it is *binding*, meaning that it is infeasible to find two openings of the same commitment to two different messages.

In this paper, we consider public-key trapdoor commitments [GQ88,Ped92] which are also *homomorphic* and *length reducing*. The former means that mes-

sages and commitments belong to abelian groups and if we multiply two commitments, we get a new commitment that contains the product of the two messages, whereas the latter requires that the commitment is shorter than the message.

An example would be a generalization of Pedersen commitments whose n message components are in $\mathbb{Z}_p$. The public key consists of $n+1$ group elements $G_1, \ldots, G_n, H$ and a commitment to $(m_1, \ldots, m_n)$ is $C = H^r \prod_{i=1}^{n} G_i^{m_i}$. This scheme is length-reducing since a commitment to n messages consists of only one group element, a feature that has been found useful in contexts such as mix-nets/voting, digital credentials, blind signatures, leakage-resilient one-way functions, and zero-knowledge proofs [FS01,Nef01,Bra99,KZ06,ADW09,Lip03].

Common to all the homomorphic trapdoor commitment schemes is that they are homomorphic with respect to *addition* in a ring or a field. However, in public-key cryptography we often work over groups that are not rings or fields and it is useful to commit to elements from such groups. Of course, if we know the discrete logarithms of the group elements we want to commit to, we can commit to them using Pedersen commitments. In general, we cannot expect to know the discrete logarithms of the messages though, leaving us with the open problem of constructing homomorphic trapdoor commitments to group elements.

Furthermore, such schemes could be combined with Pedersen commitments since commitments of the latter scheme are single group element. So, if we have a homomorphic trapdoor commitment scheme whose commitments to $O(n)$ group elements are of size $O(1)$, we can commit to $m \cdot n$ elements in $\mathbb{Z}_p$ using commitment schemes with public keys of total size $O(m+n)$. In comparison, when using only Pedersen commitments the public key would be of size $O(m \cdot n)$.

Finally, note that similarly to structure-preserving signatures, "GS compatibility" of a homomorphic trapdoor commitment scheme makes it a useful component in constructing more advanced zero-knowledge arguments or giving an efficient proof of knowledge of a message and/or an opening of a commitment.

1.1 Our Contribution

The paper presents three main results, all of them based on groups with a bilinear map. We focus on constructions in asymmetric bilinear groups whereas those in the symmetric setting are given in the full versions.

Firstly, we present a *homomorphic trapdoor commitment to group elements*. The commitments are perfectly hiding, computationally binding, and *length reducing*. An advantage of our commitment scheme is that the construction is very simple. The public key consists of $n+1$ group elements $(G_R, G_1, \ldots, G_n)$ from $\mathbb{G}_1$ and we commit to $M_1, \ldots, M_n \in \mathbb{G}_2$ by choosing $R \in \mathbb{G}_2$ at random and computing the commitment

$$C = e(G_R, R) \prod_{i=1}^{n} e(G_i, M_i) \ .$$

The commitment scheme is computationally binding under the *double pairing assumption*, which we show to be implied by decisional Diffie-Hellman assumption in $\mathbb{G}_1$. We extend our construction to commit to commitments as mentioned

above and present an honest verifier zero-knowledge argument of knowledge of the contents of such commitments.

Next, we present the first instantiation of structure-preserving signatures on group elements. The messages consist of 2 group elements from an asymmetric bilinear group and signatures of 5 elements. Since the verification keys lie in the message space, the scheme is actually an *automorphic* signature. The scheme is proved secure under a variant of the *strong Diffie-Hellman* assumption [BB04], a "q-type" assumption which holds in the generic-group model. We combine the scheme with the GS proof system to construct the first efficient round-optimal blind signature scheme, which also remains automorphic. Moreover, we give a generic transformation from any automorphic signature scheme to one that signs message vectors of arbitrary length that leaves the keys unchanged.

Lastly, we present a structure-preserving signature scheme which signs vectors of general group elements. It has a *constant signature size* regardless of the message length. Our scheme does not rely on setup assumptions nor the messages having a specific structure, e.g. Diffie-Hellman pairs, like in the previous construction. While its verification key grows linearly in the maximum message length, it is possible to extend the scheme to sign unbounded-length messages at the cost of signatures growing proportionally to the length. This way, it is possible to make the signature automorphic albeit less efficient than the scheme above. The security is based on a novel strong, "q-type", assumption which is fairly complex. However, it has an *optimal quadratic security bound* in generic bilinear groups unlike the popular strong Diffie-Hellman assumption and its variations. Finally, we define the notion of *simulatable signatures* and give an efficient instantiation. It is defined in the common reference string (CRS) model and allows to create valid signatures using the trapdoor associated with the CRS.

Applications: We illustrate the advantages of structure-preserving signature schemes by presenting several useful applications. The round-optimal blind signature scheme of Fischlin described before, which is secure in the universal-composability framework [Can01], is easily instantiated with such a building block in hand. The only extra tool we need is a trapdoor commitment on messages in $\mathbb{Z}_p$ whose commitments and openings are group elements. Such scheme is easily derived from the Pedersen commitment scheme when working in bilinear groups.

We then present a practical group signature scheme in the strongest security model [BSZ05] which moreover supports concurrent join. The construction follows a commonly used approach, based on the technique of proving knowledge of a signature.

Finally, we present the first efficient instantiation of anonymous proxy signatures (APS) in the standard model. Since automorphic signatures allow certifying public keys, delegation can be done by signing the delegatee's public key. An anonymous proxy signature is a GS proof of knowledge of a certification chain that starts at the original delegator and ends at the message. We also discuss how to strengthen the anonymity guarantees of APS. Using blind auto-

morphic signatures, we give a protocol that hides the identity of the delegatee from the delegator. Moreover, using randomizability of GS proofs, we show how to maintain anonymity of the intermediate delegators w.r.t. the delegatee.

We note that since the announcement of our work, automorphic signatures have been used to construct the first fair blind signatures without random oracles [FV10] and non-interactively delegatable anonymous credentials [Fuc10]. The commitment schemes and the related assumptions have been used to construct efficient leakage-resilient signatures and one-way relations [DHLAW10]. Moreover, one can use the commitment schemes to reduce the communication complexity of Groth's [Gro09b] sub-linear size zero-knowledge argument for circuit satisfiability from $O(|C|^{\frac{1}{2}})$ group elements to $O(|C|^{\frac{1}{3}})$ group elements.

1.2 Related Work

There are many examples of homomorphic commitments. Homomorphic cryptosystems such as [ElG86,OU98,Pai99,BGN05] or Linear Encryption [BBS04] can be seen as homomorphic commitment schemes that are perfectly binding and computationally hiding. Commitments based on homomorphic encryption can be converted into computationally binding and perfectly hiding homomorphic commitments, see for instance the mixed commitments of Damgård and Nielsen [DN02] and the commitment schemes used by Groth, Ostrovsky and Sahai [GOS06], Boyen and Waters [BW06], Groth [Gro06] and Groth and Sahai [GS08]. Even in the perfectly hiding versions of these schemes the size of a commitment is larger than the size of a message though. This length increase follows from the fact that the underlying building block is a cryptosystem whose ciphertexts must be large enough to include the message.

There are also direct constructions of homomorphic trapdoor commitment schemes such as Guillou and Quisquater commitments [GQ88] and Pedersen commitments [Ped92]. The latter are one of the most used commitment schemes in the field of cryptography. They are perfectly hiding with a trapdoor and if the discrete-logarithm problem is hard they are computationally binding. There are many variants of the Pedersen commitment scheme. Fujisaki and Okamoto [FO97] and Damgård and Fujisaki [DF02] for instance suggest a variant where the messages can be arbitrary integers. However, none of the previous trapdoor commitment schemes has messages from a *group*.

Feasibility of structure-preserving signatures on group elements was first shown by Groth [Gro06], who presents a construction based on the decision linear assumption (DLIN) [BBS04]. While it is remarkable that the security can be based on a simple standard assumption, the scheme is not practical as signatures consist of hundreds of thousands of group elements. Based on the q-Hidden LRSW assumption, Green and Hohenberger [GH08] presented an efficient scheme that provides security against random-message attacks. An extension to chosen-message security is not known.

Independently of our work, Cathalo, Libert and Yung [CLY09] gave a practical scheme based on a combination of the *hidden strong Diffie-Hellman assumption*, the *flexible Diffie-Hellman assumption*, and the DLIN assumption. It was

the first structure-preserving signature scheme to sign single group elements. However, it cannot sign its own verification keys and signatures on vectors grow linearly in their length.

An instantiation, though not practical, of anonymous proxy signatures was given in [FP09]. Moreover, they are similar to the *delegatable anonymous credentials* from [BCC+09] in that they provide mechanisms enabling users to prove possession of certain rights while remaining anonymous; and they consider redelegation of received rights. The interactive delegation protocol for anonymous credentials provides even *mutual* anonymity of the delegator and the delegatee. The two instantiations rely on similar assumptions.

1.3 Merging Our Results

This paper combines the results of three different lines of research. In [Gro09a] Groth presented the first homomorphic trapdoor commitments to group elements which are moreover length-reducing (Section 3). Fuchsbauer [Fuc09] gave the first structure-preserving signatures on group elements and used it to efficiently implement round-optimal blind signatures in the standard model (Section 4). Abe, Haralambiev and Ohkubo [AHO10] gave the first constant-size signature scheme on vectors of general group elements. They also explicitly defined the notion of simulatable signatures, gave an efficient construction, and used it to implement UC-secure round-optimal blind signatures (Sections 5 and 6.1).

2 Preliminaries

2.1 Bilinear groups

We will work in bilinear groups of the form $\Lambda = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$ where

- p is a λ-bit prime, where λ is a security parameter
- $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are order p groups with efficiently computable group operations, membership tests and map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$
- G generates $\mathbb{G}_1$, H generates $\mathbb{G}_2$ and $e(G, H)$ generates $\mathbb{G}_T$
- The map e is bilinear $\forall A \in \mathbb{G}_1 \forall B \in \mathbb{G}_2 \forall x, y \in \mathbb{Z}_p : e(A^x, B^y) = e(A, B)^{xy}$

To simplify notation, we define $\mathbb{G}_1^* = \mathbb{G}_1 \setminus \{1\}, \mathbb{G}_2^* = \mathbb{G}_2 \setminus \{1\}$ and $\mathbb{G}_T^* = \mathbb{G}_T \setminus \{1\}$.

2.2 Assumptions

We will work with bilinear groups generated by a probabilistic polynomial-time algorithm $\mathcal{G}$ that takes the security parameter as input. The schemes we present will rely on one or more of the following computational assumptions about the bilinear groups generated by $\mathcal{G}$. We note right away that the assumptions imply $\mathbb{G}_1 \neq \mathbb{G}_2$ and furthermore some of them imply that we are working in so called type III bilinear groups [GPS08] where there are no efficiently computable non-trivial homomorphisms between the two base groups $\mathbb{G}_1$ and $\mathbb{G}_2$. We refer to the full papers for schemes that work in type I and type II bilinear groups.

Variants of DDH and CDH. The *decisional Diffie-Hellman (DDH) problem* in a group $\mathbb{G}$ is, given (G, G^a, G^b, G^c), to decide whether $c = ab$. The *symmetric external Diffie-Hellman (SXDH) assumption* in a bilinear group states that DDH is hard in both groups.

Assumption 1 (SXDH). *For $\Lambda = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H) \leftarrow \mathcal{G}(1^\lambda)$, the decisional Diffie-Hellman assumption holds in both $\mathbb{G}_1$ and $\mathbb{G}_2$.*

The *2-out-of-3 CDH* assumption [KP06] states that given (G, G^a, H), it is hard to output (G^r, H^{ar}) for an arbitrary $r \neq 0$. To break the *Flexible CDH* assumption [LV08,CLY09], an adversary must additionally compute G^{ar}. We further weaken the assumption by defining a solution as $(G^r, G^{ar}, H^r, H^{ar})$, and generalize it to asymmetric groups by letting $G \in \mathbb{G}_1$ and $H \in \mathbb{G}_2$. The *asymmetric weak flexible CDH* is defined as follows:

Assumption 2 (AWF-CDH). *Let $G \in \mathbb{G}_1$, $H \in \mathbb{G}_2$ and $a \in \mathbb{Z}_p$ be random. Given $(G, A = G^a, H)$, it is hard to output $(G^r, G^{ar}, H^r, H^{ar})$ with $r \neq 0$, i.e., a tuple (R, M, S, N) that satisfies*

$$e(A, S) = e(M, H) \qquad e(M, H) = e(G, N) \qquad e(R, H) = e(G, S) \quad (1)$$

Given a DDH instance (G, G^a, G^b, G^c), solving AWF-CDH for (G, G^a, H) yields $(G^r, G^{ar}, H^r, H^{ar})$; thus $G^c = G^{ab}$ can be checked by $e(G^{ab}, H^r) = e(G^b, H^{ar})$. We have thus

Lemma 1. *The AWF-CDH assumption holds if the decisional Diffie-Hellman assumption is hard in $\mathbb{G}_1$.*

The Double Pairing Assumption. The double pairing problem is given random $G_R, G_T \in \mathbb{G}_1$ to find non-trivial $R, S \in \mathbb{G}_2$ satisfying $e(G_R, R)e(G_T, T) = 1$.

Assumption 3 (DBP). *For all nonuniform polynomial-time adversaries $\mathcal{A}$*

$$\Pr\Big[\Lambda \leftarrow \mathcal{G}(1^\lambda);\ G_R, G_T \leftarrow \mathbb{G}_1;\ (R, T) \leftarrow \mathcal{A}(\Lambda, G_R, G_T) : (R, T) \in \mathbb{G}_2^* \times \mathbb{G}_2^* \quad \wedge \quad e(G_R, R)e(G_T, T) = 1\Big] = \mathrm{negl}(\lambda).$$

We show in the full papers the following lemma:

Lemma 2. *The double pairing assumption holds if the decisional Diffie-Hellman assumption is hard in $\mathbb{G}_1$.*

The reverse double pairing problem, where the base groups are interchanged and the challenge is to find a non-trivial pair $(R, S) \in \mathbb{G}_1^2$ is defined analogously.

Next, observe that given an answer to an instance of the DBP problem, one can easily yield more answers. We eliminate such possibility by multiplying random pairings to both sides of the equation. As one of those stays the same in all instances, whereas the other, $e(V, W)$, changes in each instance, the intuition

is that it would be hard to combine $e(V, W)$ and $e(V', W')$ into one equivalent pairing $e(V'', W'')$ — we call such a pairing *flexible* as it can be easily randomized and, when relations with respect to the same base is known, combined with another. Also, to make the assumption valid, we make a system of two such equations and require that their solutions share a common element, Z.

Assumption 4 (Simultaneous Flexible Pairing Assumption (q-SFP)). *Let Λ be a bilinear groups setup and let G_Z, F_Z, G_R, and F_U be random generators of $\mathbb{G}_1$. Let $(A, \tilde{A}), (B, \tilde{B})$ be random pairs in $\mathbb{G}_1 \times \mathbb{G}_2$. For $j = 1, \ldots, q$, let $R_j = (Z, R, S, T, U, V, W)$ that satisfies*

$$e(A, \tilde{A}) = e(G_Z, Z)\ e(G_R, R)\ e(S, T) \quad \textit{and} \tag{2}$$

$$e(B, \tilde{B}) = e(F_Z, Z)\ e(F_U, U)\ e(V, W). \tag{3}$$

Given $(\Lambda, G_Z, F_Z, G_R, F_U, A, \tilde{A}, B, \tilde{B})$ and uniformly chosen $R_1, \ldots, R_q$, it is hard to find $(Z^\star, R^\star, S^\star, T^\star, U^\star, V^\star, W^\star)$ that fulfill relations (2) and (3) under the restriction that $Z^\star \neq 1$ and $Z^\star \neq Z \in R_j$ for every R_j.

We also show that the SFP assumption can be justified and has an optimal bound in the generic bilinear group model.

Lemma 3. *For any generic algorithm $\mathcal{A}$, the probability that $\mathcal{A}$ breaks SFP with ℓ group operations and pairings is bound by $\mathcal{O}(q^2 + \ell^2)/p$.*

A variant of the q-strong Diffie Helmman assumption. The *q-strong Diffie-Hellman* (SDH) assumption [BB04] implies hardness of the following two problems in bilinear groups [FPV09]:

1. Given G, G^x and $q-1$ pairs $(G^{\frac{1}{x+c_i}}, c_i)$, output a new pair $(G^{\frac{1}{x+c}}, c)$.
2. Given $G, K, G^x, \left((K \cdot G^{v_i})^{\frac{1}{x+c_i}}, c_i, v_i\right)_{i=1}^{q-1}$, output a new $((K \cdot G^v)^{\frac{1}{x+c}}, c, v)$.

Boyen and Waters [BW07] define the *hidden* SDH assumption which states that the first problem is hard when the pairs are substituted with triples of the form $(G^{1/(x+c_i)}, G^{c_i}, H^{c_i})$, for a fixed H. Analogously, Fuchsbauer et al. [FPV09] define the *double hidden* SDH (DHSDH) by giving the scalars in the second problem as exponentiations of two group elements. We adapt DHSDH to asymmetric groups by giving generators $G, F, K \in \mathbb{G}_1$ and $H \in \mathbb{G}_2$; the elements c_i and v_i are given as (F^{c_i}, H^{c_i}) and (G^{v_i}, H^{v_i}). Due to the pairing, a tuple can thus be effectively verified. The assumption holds in the generic-group model [Sho97] for both asymmetric and symmetric groups [Fuc09] and falls in the generalized "Uber-Assumption" family [Boy08]

Assumption 5 (q-ADH-SDH). *Let $G, F, K \in \mathbb{G}_1$, $H \in \mathbb{G}_2$ and $x, c_i, v_i \in \mathbb{Z}_p$ be random. Given $(G, F, K, X = G^x;\ H, Y = H^x)$ and*

$$\left(A_i = (K \cdot G^{v_i})^{\frac{1}{x+c_i}},\ C_i = F^{c_i},\ D_i = H^{c_i},\ V_i = G^{v_i},\ W_i = H^{v_i}\right)\ ,$$

for $1 \leq i \leq q-1$, it is hard to output a new tuple $((K \cdot G^v)^{\frac{1}{x+c}}, F^c, H^c, G^v, H^v)$ with $(c, v) \neq (c_i, v_i)$ for all i.

Note that a tuple (A, C, D, V, W) of this form satisfies the following equations:

$$e(A, Y \cdot D) = e(K \cdot V, H) \qquad e(C, H) = e(F, D) \qquad e(V, H) = e(G, W) \qquad (4)$$

2.3 Digital Signatures

A digital signature scheme $\mathbf{Sig} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ consists of the following algorithms: Setup outputs system parameters; KeyGen outputs a pair (vk, sk) of verification and signing keys; and $\mathsf{Sign}(sk, M)$ outputs a signature σ, which is verified by $\mathsf{Verify}(vk, M, \sigma)$. Signatures are *existentially unforgeable under chosen-message attack* (EUF-CMA) [GMR88] if no adversary, given vk and a signing oracle for messages of its choice, can output a pair (M, σ) s.t. M was never queried and $\mathsf{Verify}(vk, M, \sigma) = 1$.
Signatures are *strongly* EUF-CMA (sEUF-CMA) if no adversary can output a valid pair (M, σ) such that $(M, \sigma) \neq (M_i, \sigma_i)$ for all i, with M_i being the i-th oracle query and σ_i the response.

2.4 SXDH Groth-Sahai Proofs for Pairing-Product Equations

One of the main motivations of structure-preserving signatures is to combine them with Groth-Sahai (GS) proofs [GS08], in particular witness-indistinguishable (WI) proofs of satisfiability of *pairing-product equations* (PPE). A PPE over variables $X_1, \ldots, X_m \in \mathbb{G}_1$, $Y_1, \ldots, Y_n \in \mathbb{G}_2$ is an equation of the form

$$\prod_{i=1}^{n} e(A_i, Y_i) \prod_{i=1}^{m} e(X_i, B_i) \prod_{i=1}^{m} \prod_{j=1}^{n} e(X_i, Y_j)^{\gamma_{i,j}} \;=\; \mathbf{t}_T \;, \qquad (E)$$

determined by $A_j \in \mathbb{G}_1, B_i \in \mathbb{G}_2, \gamma_{i,j} \in \mathbb{Z}_p$, and $\mathbf{t}_T \in \mathbb{G}_T$.

Groth and Sahai define an extractable commitment scheme for group elements. The setup algorithm is given a bilinear group and outputs a commitment key $ck \in \mathbb{G}_1^4 \times \mathbb{G}_2^4$. A commitment $\mathsf{Com}(ck, X, \rho)$ to $X \in \mathbb{G}_i$ using randomness $\rho \in \mathbb{Z}_p^2$ is in $\mathbb{G}_i^2$ (for $i = 1, 2$). These commitments are perfectly binding and given an *extraction key*, the committed values can be recovered.

A *proof of satisfiability* of a PPE is constructed as follows. First, make commitments to the satisfying witness $(X_1, \ldots, X_m, Y_1, \ldots, Y_n)$. Then make a *proof* ϕ that the committed values satisfy the equation, using the values and the randomness of the commitments. The proofs, which are in $\mathbb{G}_1^4 \times \mathbb{G}_2^4$, are perfectly sound: if a proof passes verification for a set of commitments then the committed (and extractable) values satisfy the equation.

There is an alternative setup that outputs keys ck^* which lead to commitments and proofs that are equally distributed for all witnesses. Under SXDH, these keys are indistinguishable from original keys; witness indistinguishability of GS proofs follows thus from SXDH.

Note that due to extractability, a proof of satisfiability is actually a non-interactive *proof of knowledge* of a witness; we will write thus

$$\mathsf{NIPK}\{(X_1, .., X_m, Y_1, .., Y_n) : \textstyle\prod e(A_i, Y_i) \prod e(X_i, B_i) \prod \prod e(X_i, Y_j)^{\gamma_{i,j}} = \mathbf{t}_T\}$$

and PKVrf for the verification algorithm.

If for a signature scheme, public keys, messages and signatures are group elements that are verified by checking PPEs, we can commit to (encrypt) keys, messages and/or signatures and prove validity of the committed values using GS proofs.

Randomization. Groth-Sahai commitments can be randomized, in particular, given $\mathbf{c} = \mathsf{Com}(ck, X, \rho)$, one can compute $\mathsf{Com}(ck, X, \rho + \rho')$ for any ρ' without knowledge of X or ρ. Moreover, given commitments and a proof ϕ that the committed values satisfy a PPE, we can randomize the commitments and adapt ϕ to the randomized commitments [BCC+09]. WI implies that a randomized proof is indistinguishable from a proof computed with a different witness.

3 Commitments

A non-interactive commitment scheme consists of three polynomial-time algorithms $(\mathcal{G}, \mathcal{G}_{\mathrm{com}}, \mathrm{com})$. $\mathcal{G}$ is a probabilistic polynomial-time setup algorithm that takes as input the security parameter λ and outputs some setup information Λ; in our commitment scheme $\mathcal{G}$ will be a bilinear group generator. $\mathcal{G}_{\mathrm{com}}$ is a probabilistic polynomial-time algorithm that takes as input the setup Λ and and generates a public commitment key ck and a trapdoor key tk. The commitment key ck specifies a message space $\mathcal{M}_{ck}$, a randomizer space $\mathcal{R}_{ck}$ and a commitment space $\mathcal{C}_{ck}$. We assume it is easy to verify membership of the message space, randomizer space and the commitment space and it is possible to sample randomizers uniformly at random from $\mathcal{R}_{ck}$. The algorithm Com takes as input the commitment key ck, a message m from the message space, a randomizer r from the randomizer space and outputs a commitment c in the commitment space. We call a message-randomizer pair an opening. Anybody with an opening and a commitment can check whether the commitment is a commitment to the message specified in the opening.

A commitment scheme should be binding, which means it is infeasible to find two openings of the same commitment to two different messages. A commitment scheme should also be hiding such that the commitment does not disclose anything about the message. Our commitment scheme is a trapdoor commitment scheme, which makes it hiding in a very strong sense. The commitment has a trapdoor opening algorithm Topen that takes the trapdoor key, an opening of a commitment and a message and outputs a randomizer such that the message and the randomizer constitute a new opening of the commitment.

We will now describe our commitment scheme. The commitment scheme will have message space $\mathcal{M}_{ck} = \mathbb{G}_2^n$, randomizer space $\mathcal{R}_{ck} = \mathbb{G}_2$ and commitment space $\mathcal{C}_{ck} = \mathbb{G}_T$. In other words, we can commit to an n-tuple of base group elements with a commitment that consists of a single target group element.

Setup: On input 1^λ return $\Lambda = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H) \leftarrow \mathcal{G}(1^\lambda)$.

Key generation: On input Λ pick $G_R \leftarrow \mathbb{G}_1^*$ and $x_1, \ldots, x_n \leftarrow \mathbb{Z}_p$ and set $G_1 = G_R^{x_1}, \cdots, G_n = G_R^{x_n}$. The commitment and trapdoor keys are

$$ck = (\Lambda, G_R, G_1, \ldots, G_n) \quad \text{and} \quad tk = (ck, x_1, \ldots, x_n).$$

Commitment: Using commitment key ck on input message $(M_1, \ldots, M_n) \in \mathbb{G}_2^n$ pick randomizer $R \leftarrow \mathbb{G}_2$. The commitment is given by

$$C = e(G_R, R) \prod_{i=1}^{n} e(G_i, M_i) \ .$$

Trapdoor opening: On a commitment $C \in \mathbb{G}_T$ with opening $(M_1, \ldots, M_n, R) \in \mathbb{G}_2^n \times \mathbb{G}_2$ and another message $(M_1', \ldots, M_n') \in \mathbb{G}_2^n$ use the trapdoor key tk to compute the trapdoor randomizer $R' = R \prod_{i=1}^{n} (M_i/M_i')^{x_i}$. This gives us a trapdoor opening $(M_1', \ldots, M_n', R')$ satisfying

$$C = e(G_R, R) \prod_{i=1}^{n} e(G_i, M_i) = e(G_R, R') \prod_{i=1}^{n} e(G_i, M_i') \ .$$

The commitment scheme has several useful properties. The commitment is length-reducing, since a commitment to a tuple of messages yields a commitment consisting of a single target group element. The commitment scheme is homomorphic since multiplying two commitments yields a commitment to the entry-wise product of the messages, i.e.,

$$e(G_R, R) \prod_{i=1}^{n} e(G_i, M_i) \cdot e(G_R, R') \prod_{i=1}^{n} e(G_i, M_i') \ = \ e(G_R, RR') \prod_{i=1}^{n} e(G_i, M_i M_i').$$

The commitment scheme is perfectly hiding since for all messages $(M_1, \ldots, M_n) \in \mathbb{G}_2^n$ the commitment procedure returns a uniformly random commitment $C \in \mathbb{G}_T$ and therefore no information is leaked about the commitment. Indeed, with the trapdoor key we can even take a commitment and its opening and create an opening to any other message. Finally, we prove in the full papers that the commitment scheme is computationally binding if the double pairing assumption holds for the bilinear group generator $\mathcal{G}$. We summarize these properties in the theorem below, which we prove in the full papers.

Theorem 1. $(\mathcal{G}, \mathcal{G}_{\text{com}}, \mathsf{Com}, \mathsf{Topen})$ *described above is a homomorphic, perfectly hiding trapdoor commitment scheme; and assuming the double pairing assumption holds for $\mathcal{G}$ the commitment scheme is computationally binding.*

It is straightforward to construct a similar type of commitment scheme for tuples in $\mathbb{G}_1^n$ using the reverse double pairing assumption.

Committing to commitments. The defining characteristic of our commitment scheme is that we commit to base group elements as opposed to field

elements. This opens up new applications for commitment schemes. As a simple example, we can for instance construct commitments to commitments. Recall that Pedersen commitments to tuples $(m_1, \ldots, m_n) \in \mathbb{Z}_p^n$ are of the form $C = H^r \prod_{j=1}^{n} H_j^{m_j}$. Each Pedersen commitment is a group element, and we can commit to many Pedersen commitments using our commitment scheme. Combining the two commitment schemes we can commit to n^2 field elements from $\mathbb{Z}_p$. Since both Pedersen commitments and our commitments are homomorphic, the combined commitment scheme is also homomorphic. It also preserves the trapdoor opening property and is perfectly hiding. A commitment consists of a single group element in $\mathbb{G}_T$ and the commitment key consists of approximately $2n$ group elements, so unlike the Pedersen commitment we have a commitment key that is much smaller than the messages.

4 Automorphic Signatures

For elaborate applications, Groth-Sahai compatibility of a signature scheme is not sufficient; in addition, the verification keys have to lie in the message space. This enables constructions of *certification chains* (sequences of public keys linked by certificates from one key on the next one), which can be anonymized by GS proofs, as required by anonymous proxy signatures (see Section 6.3) and delegatable anonymous credentials. We call such a scheme an *automorphic signature*, as it is able to sign its *own* keys and it is *structure* preserving.

Definition 1. *An* automorphic signature *over* $\Lambda = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$ *is an EUF-CMA secure signature whose verification keys lie in the message space. Moreover, the messages and signatures consist of elements from* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*, and the verification predicate is a conjunction of pairing-product equations.*

The trick that enables an efficient instantiation of automorphic signatures is to define a message (and thus a verification key) as a *pair* of group elements of the form (G^v, H^v). Hence, the message space is the set of *Diffie-Hellman pairs* $\mathcal{DH} = \{(G^v, H^v) \,|\, v \in \mathbb{Z}_p\}$. In Assumption 5, we could interpret G, F, K, H as parameters, (X, Y) as the public key, (V, W) as the message and (A, C, D) as the signature—since a signer holding the secret key x can choose c and produce (A, C, D) without knowing v. ADH-SDH states that these signatures are unforgeable when the adversary gets $q - 1$ signatures on *random* messages.

To make the scheme secure against chosen-message attacks, we interpret G^v in the definition of A as a *trapdoor commitment* to the message (M, N). The key is an element $T := G^t \in \mathbb{G}_1$, where t is the trapdoor, and a commitment to (M, N) is defined as $V := T^r \cdot M$ with opening (G^r, H^r). AWF-CDH implies that the commitments are computationally binding. Trapdoor opening requires knowledge of W such that $(V, W) \in \mathcal{DH}$: for any $(V, W), (M, N) \in \mathcal{DH}$, a valid opening is $((V \cdot M)^{-t}, (W \cdot N)^{-t})$.

The final signature will be (A, C, D) together with the opening of the commitment (R, S); a signature is thus in $\mathbb{G}_1^3 \times \mathbb{G}_2^2$.

4.1 Instantiation

Our automorphic signature scheme $\mathbf{Sig} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is defined as follows.

Setup: On input 1^λ run $\Lambda = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H) \leftarrow \mathcal{G}(1^\lambda)$, choose random elements $F, K, T \in \mathbb{G}_1$ and output the parameters $pp := (\Lambda, F, K, T)$. The message space is $\mathcal{DH} := \{(G^m, H^m) \,|\, m \in \mathbb{Z}_p\}$.

Key generation: On input pp choose $x \leftarrow \mathbb{Z}_p$ and return the verification key $vk := (G^x, H^x)$ and the signing key $sk := x$.

Signing: On input the parameters pp, a secret key x and a message $(M, N) \in \mathcal{DH}$, choose $c, r \leftarrow \mathbb{Z}_p$ and return

$$A := (K \cdot T^r \cdot M)^{\frac{1}{x+c}} \qquad C := F^c \qquad D := H^c \qquad R := G^r \qquad S := H^r$$

Verification: On input pp, a public key (X, Y) and a message (M, N), both in $\mathcal{DH}$, and a signature (A, C, D, R, S), return 1 if

$$e(A, Y \cdot D) = e(K \cdot M, H)\, e(T, S) \qquad \begin{matrix} e(C, H) = e(F, D) \\ e(R, H) = e(G, S) \end{matrix} \tag{5}$$

Theorem 2. *Under ADH-SDH and AWF-CDH,* **Sig** *is strongly unforgeable against chosen-message attacks.*

We refer to the full version [Fuc09] for a proof. Note that the scheme can also be instantiated for $\mathbb{G}_1 = \mathbb{G}_2$. Our scheme (and the blind signature scheme in the next section) can also be used to sign bit strings if we assume a collision-resistant hash function $\mathsf{Hash}\colon \{0,1\}^* \to \mathbb{Z}_p$: before signing a message or verifying a signature, we map $m \in \{0,1\}^*$ to $(M, N) := (G^{\mathsf{Hash}(m)}, H^{\mathsf{Hash}(m)}) \in \mathcal{DH}$.

4.2 Automorphic Blind Signatures

We now show how to combine automorphic signatures with the Groth-Sahai proof system to construct the first round-optimal blind signature scheme, satisfying standard security requirements as in [Oka06] (see Section 6.1 for a universally composable scheme). Similarly to Fischlin's generic construction, our blind signatures are defined as a proof of knowledge of a signature from an underlying scheme, which perfectly hides the signature. We thus only have to ensure that the signer does not learn the message while signing. In our scheme the user sends a *randomization* of the message, on which the signer makes a "pre-signature". By adapting the randomness, the user can retrieve a signature *on the message* (rather than on a commitment for which the user has to prove knowledge of the opening, as in Fischlin's construction). This increases useability of our blind signatures for applications (cf. Section 6.3) and also makes them shorter.

To obtain a blind signature on (M, N), the user randomly picks $\rho \leftarrow \mathbb{Z}_p$ and *blinds* M by the factor T^ρ. In addition to $U := T^\rho \cdot M$, she sends a GS proof

Obtain$\big((pp', ck), vk, (M, N)\big)$. Choose $\rho \leftarrow \mathbb{Z}_p$, set $P := G^\rho$, $Q := H^\rho$, and send:

- $U := T^\rho \cdot M$
- $\phi := \mathsf{NIPK}\big\{(M, N, P, Q) : e(M, H) = e(G, N) \wedge e(P, H) = e(G, Q) \wedge e(T, Q)\, e(M, H) = e(U, H)\big\}$

Issue$\big((pp', ck), x\big)$. If ϕ is valid, choose $c, r \leftarrow \mathbb{Z}_p$ and send:

$$A := (K \cdot T^r \cdot U)^{\frac{1}{x+c}} \qquad C := F^c \qquad D := H^c \qquad R' := G^r \qquad S' := H^r$$

Obtain sets $R := R' \cdot P$, $S := S' \cdot Q$. If (A, C, D, R, S) is valid on (M, N) under vk, it outputs

$$\sigma := \mathsf{NIPK}\big\{(A, C, D, R, S) : \mathsf{Verify}_{\mathbf{Sig}}\big(pp, vk, (M, N), (A, C, D, R, S)\big)\big\} \ .$$

Fig. 1. Two-move blind signing protocol.

of knowledge of (M, N, G^ρ, H^ρ). The signer now formally produces a signature[6] on U, for which we have $A = (K \cdot T^r \cdot U)^{1/(x+c)} = (K \cdot T^{r+\rho} \cdot M)^{1/(x+c)}$; thus A is the first component of a signature on (M, N) with randomness $r + \rho$. The user can complete the signature by adapting randomness r to $r + \rho$ in the other components. The blind signature is a GS proof of knowledge of this signature.

Our blind signature scheme $\mathbf{BSig} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Obtain}, \mathsf{Issue}, \mathsf{Verify})$ is defined as follows.

Setup: On input 1^λ run the setup algorithms for **Sig** and for Groth-Sahai proofs; return the respective outputs pp' and ck as parameters pp.

Key generation: The message space and key generation are defined as for **Sig**.

Signature issuing: The protocol consists of interactive algorithms Obtain, run by the user, and Issue, run by the signer. Obtain has inputs pp, the signer's verification key vk and a message $(M, N) \in \mathcal{DH}$. Issue has inputs pp and the signing key x. The protocol is given in Figure 1.

Verification: On input pp, a verification key vk, a message $(M, N) \in \mathcal{DH}$ and a signature σ, return 1 if σ is a valid Groth-Sahai proof, i.e.,

$$\mathsf{PKVrf}\{\sigma : \mathsf{Verify}_{\mathrm{A}}(vk, (M, N), \cdot)\} = 1 \ .$$

Theorem 3. *Under ADH-SDH and SXDH, scheme* **BSig** *is an unforgeable blind-signature scheme.*

Using soundness of Groth-Sahai proofs, unforgeability is shown by reduction to the unforgeability of **Sig**, which holds under ADH-SDH and SXDH (the latter

[6] Note that the user does *not* obtain a signature on U (unless $U = M$), since it is not an element of the message space; to produce $(U, H^{\log_G U}) \in \mathcal{DH}$, the user would have to break AWF-CDH.

implies AWF-CDH). Under SXDH, the user's message (U, ϕ) computationally hides (M, N) and the blind signature hides what the signer sends in the issuing; together this can be shown to imply blindness. See [Fuc09] for a formal proof of Theorem 3.

The round complexity of the scheme is optimal. A blind signature consists of commitments to (A, C, D, R, S) in $\mathbb{G}_1^6 \times \mathbb{G}_2^4$ and GS proofs, which are in $\mathbb{G}_1^4 \times \mathbb{G}_2^4$, for 3 equations. A blind signature is thus in $\mathbb{G}_1^{18} \times \mathbb{G}_2^{16}$, the two messages sent during issuing are in $\mathbb{G}_1^{17} \times \mathbb{G}_2^{16}$ and $\mathbb{G}_1^3 \times \mathbb{G}_2^2$, respectively. Note that the scheme remains automorphic since GS proofs consists of group elements and are verified by checking pairing-product equations.

4.3 Automorphic Signatures on Message Vectors

In order to sign vectors of messages of arbitrary length, we proceed as follows. We first show how to transform any signature scheme whose message space forms an algebraic group (and contains the public-key space) into one that signs 2 messages at once—if we exclude the neutral element from the message space of the transform. A signature on a message pair will contain 3 signatures (of the original scheme) on different *products* of the components. Note that $\mathcal{DH}$, the message space of **Sig**, is a group when the group operation is defined as component-wise multiplication.

We then give a straightforward generic transformation from any scheme signing 2 messages (and whose verification keys lie in the message space) to one signing message vectors of arbitrary length (Definition 3). Both transformations do not modify setup and key generation and they are invariant w.r.t. the structure of verification; in particular, if the verification predicate of the original scheme is a conjunction of PPEs then so is that of the transform.

Definition 2. *Let* $\mathbf{Sig} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *be a signature scheme whose message space* $(\mathcal{M}, \cdot)$ *is an algebraic group that contains the verification keys. The* pair transform *of* **Sig** *with message space* $\mathcal{M}^* \times \mathcal{M}^*$ *is defined as* $\mathbf{Sig}' = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}', \mathsf{Verify}')$ *with*

$\mathsf{Sign}'(sk, (M_1, M_2))$**:** *Set* $(vk_0, sk_0) \leftarrow \mathsf{KeyGen}$ *and return*

$$\sigma := \big(vk_0,\ \mathsf{Sign}(sk, vk_0),\ \mathsf{Sign}(sk_0, M_1),\ \mathsf{Sign}(sk_0, M_1 \cdot M_2),\ \mathsf{Sign}(sk_0, M_1 \cdot M_2^3)\big)\ .$$

$\mathsf{Verify}'\big(vk, (M_1, M_2), (vk_0, \sigma_0, \sigma_1, \sigma_2, \sigma_3)\big)$**:** *Return 1 if all of the following are 1:*

$$\mathsf{Verify}(vk, vk_0, \sigma_0)$$
$$\mathsf{Verify}(vk_0, M_1, \sigma_1) \quad \mathsf{Verify}(vk_0, M_1 \cdot M_2, \sigma_2) \quad \mathsf{Verify}(vk_0, M_1 \cdot M_2^3, \sigma_3)$$

Theorem 4. *If* **Sig** *is EUF-CMA secure then so is* $\mathbf{Sig}'$.

Definition 3. *Let* $\mathbf{Sig} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *be a signature scheme with message space* $\mathcal{M} \times \mathcal{M}$*, such that* $\mathcal{M}$ *contains the verification keys. Assume an efficiently computable injection* $I\colon \{1, \ldots, |\mathcal{M}|\} \to \mathcal{M}$*. The* vector transform *of* $\mathbf{Sig}$ *is defined as* $\mathbf{Sig}'' = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}'', \mathsf{Verify}'')$ *with*

$\mathsf{Sign}''(sk, (M_1, \ldots, M_n))$**:** *Set* $(vk_0, sk_0) \leftarrow \mathsf{KeyGen}$ *and return*

$$\sigma := \big(vk_0,\, \mathsf{Sign}(sk, vk_0, I(n)),\, \mathsf{Sign}(sk_0, M_1, I(1)), \ldots, \mathsf{Sign}(sk_0, M_n, I(n))\big)\ .$$

$\mathsf{Verify}''\big(vk, (M_1, \ldots, M_n), (vk_0, \sigma_0, \sigma_1, \ldots, \sigma_n)\big)$**:** *Return 1 if the following are 1:*

$$\mathsf{Verify}\big(vk, (vk_0, I(n)), \sigma_0\big) \qquad \mathsf{Verify}\big(vk_0, (M_i, I(i)), \sigma_i\big) \quad \textit{(for all } 1 \le i \le n\textit{)}$$

Theorem 5. *If* $\mathbf{Sig}$ *is EUF-CMA secure then so is* $\mathbf{Sig}''$*.*

We refer to [Fuc09] for proofs of Theorems 4 and 5 where we also discuss why the construction in Definition 2 is optimal and why it seems somehow hard to construct a generic vector transform directly.

5 Signatures on Vectors of Group Elements

In this section, we present the first *constant-size* structure-preserving signature scheme for messages of general bilinear groups elements. We start by describing useful randomization techniques, followed by the scheme description and various extensions. Full details, as well as the byproduct of several trapdoor commitment schemes, can be found in [AHO10].

5.1 Randomization Techniques

We introduce techniques that randomize elements in a pairing or a pairing product without changing their value in $\mathbb{G}_T$. Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H) \leftarrow \mathcal{G}(1^\lambda)$.

Inner Randomization $(X', Y') \leftarrow \mathsf{Rand}(X, Y)$**:** A pairing $A = e(X, Y) \neq 1$ is randomized as follows. Choose $\gamma \leftarrow \mathbb{Z}_p^*$ and let $(X', Y') = (X^\gamma, Y^{1/\gamma})$. It then holds that (X', Y') distributes uniformly over $\mathbb{G}_1 \times \mathbb{G}_2$ under the condition of $A = e(X', Y')$. If $A = 1$, then first flip a coin and pick $e(1, 1)$ with probability $1/(2p-1)$. If it is not selected, flip a coin and pick either $e(1, X)$ or $e(X, 1)$ with probability $1/2$. Then select X uniformly from the corresponding group except for 1.

Sequential Randomization $\{X_i', Y_i'\}_{i=1}^k \leftarrow \mathsf{RandSeq}(\{X_i, Y_i\}_{i=1}^k)$**:** A pairing product $A = e(X_1, Y_1)\, e(X_2, Y_2) \ldots e(X_k, Y_k)$ is randomized into $A = e(X_1', Y_1')\ e(X_2', Y_2') \ldots\ e(X_k', Y_k')$ as follows: Let $(\gamma_1, \ldots, \gamma_{k-1}) \leftarrow \mathbb{Z}_p^{k-1}$. We begin with randomizing the first pairing by using the second pairing as follows. First verify that $Y_1 \neq 1$ and $X_2 \neq 1$. If $Y_1 = 1$, replace the first pairing $e(X_1, 1)$ with $e(1, Y_1)$ with a new random $Y_1 (\neq 1)$. The case of $X_2 = 1$ is handled in the same manner.

Then multiply $1 = e(X_2^{-\gamma_1}, Y_1)\, e(X_2, Y_1^{\gamma_1})$ to both sides of the formula. We thus obtain

$$A = e(X_1 X_2^{-\gamma_1}, Y_1)\, e(X_2, Y_1^{\gamma_1} Y_2)\, e(X_3, Y_3) \cdots e(X_k, Y_k).$$

Next we randomize the second pairing by using the third one. As before, if $Y_1^{\gamma_1} Y_2 = 1$ or $X_3 = 1$, replace them to random values. Then multiply $1 = e(X_3^{-\gamma_2}, Y_1^{\gamma_1} Y_2)\, e(X_3, (Y_1^{\gamma_1} Y_2)^{\gamma_2})$. We thus have

$$A = e(X_1 X_2^{-\gamma_1}, Y_1)\, e(X_2 X_3^{-\gamma_2}, Y_1^{\gamma_1} Y_2)\, e(X_3, (Y_1^{\gamma_1} Y_2)^{\gamma_2} Y_3) \cdots e(X_k, Y_k).$$

This continues up to the $(k-1)$-st pairing. When done, the value of the i-th pairing distributes uniformly in $\mathbb{G}_T$ due to the uniform choice of γ_i. The k-th pairing follows the distribution determined by A and preceding $k-1$ pairings. To complete the randomization, every pairing is processed by the inner randomization.

The sequential randomization can be used to **extend** a product of k pairings to a product of arbitrary k' pairings, $k' \geq k$, by appending $e(1,1)$ before randomization. By $\{X'_i, Y'_i\}_{i=1}^{k'} \leftarrow \mathsf{RandExtend}(\{X_i, Y_i\}_{i=1}^{k})$ we denote the sequential randomization with extension. Parameters k and k', $k' \geq k$, should be clear from the input and the output.

Note that the algorithms yield uniform elements and thus may include pairings that evaluate to $1_{\mathbb{G}_T}$. If it is not preferable, it can be avoided by repeating that particular step once again excluding the bad randomness.

5.2 Basic Signature Scheme

We define the signature scheme $\mathbf{Sig} = (\mathcal{G}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ below. In addition to the common parameters outputted by the $\mathcal{G}$ algorithm, the key generation algorithm KeyGen also takes a parameters k which determines the message space $\mathbb{G}_2^k$; messages of shorter length are implicitly padded with $1_{\mathbb{G}_2}$-s. We do not use any trusted setup, but only the bilinear group generation.

Setup: On input 1^λ return $\Lambda = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H) \leftarrow \mathcal{G}(1^\lambda)$.

Key generation: On input Λ and k, choose random generators $G_R, F_U \leftarrow \mathbb{G}_1^*$. For $i = 1, \ldots, k$, choose $\gamma_i, \delta_i \leftarrow \mathbb{Z}_p^{*2}$ and compute $G_i = G_R^{\gamma_i}$ and $F_i = F_U^{\delta_i}$. Choose $\gamma_Z, \delta_Z \leftarrow \mathbb{Z}_p^{*2}$ and compute $G_Z = G_R^{\gamma_Z}$ and $F_Z = F_U^{\delta_Z}$. Also choose $\alpha, \beta \leftarrow \mathbb{Z}_p^{*2}$ and compute $\{A_i, \tilde{A}_i\}_{i=0}^{1} \leftarrow \mathsf{RandExtend}(G_R, H^\alpha)$ and $\{B_i, \tilde{B}_i\}_{i=0}^{1} \leftarrow \mathsf{RandExtend}(F_U, H^\beta)$. Set $sk = (vk, \alpha, \beta, \gamma_Z, \delta_Z, \{\gamma_i, \delta_i\}_{i=1}^{k})$ and $vk = (\Lambda, G_Z, F_Z, G_R, F_U, \{G_i, F_i\}_{i=1}^{k}, \{A_i, \tilde{A}_i, B_i, \tilde{B}_i\}_{i=0}^{1})$. Output (vk, sk).

Signature issuing: On input sk and $\vec{M}$, choose $\zeta, \rho, \tau, \varphi, \omega$ randomly from $\mathbb{Z}_p^*$ and set:

$$Z = H^\zeta,\ R = H^{\rho - \gamma_Z \zeta} \textstyle\prod_{i=1}^{k} M_i^{-\gamma_i},\ S = G_R^\tau,\ T = H^{(\alpha-\rho)/\tau},$$
$$U = H^{\varphi - \delta_Z \zeta} \textstyle\prod_{i=1}^{k} M_i^{-\delta_i},\ V = F_U^\omega,\ W = H^{(\beta-\varphi)/\omega}.$$

Output $\sigma = (Z, R, S, T, U, V, W)$ as a signature.

Verification: On input vk, $\vec{M}$, and σ, parse the signature σ as (Z, R, S, T, U, V, W). Output 1 if the following equations:

$$A = e(G_Z, Z)\ e(G_R, R)\ e(S, T)\ \prod_{i=1}^{k} e(G_i, M_i) \text{ and} \tag{6}$$

$$B = e(F_Z, Z)\ e(F_U, U)\ e(V, W)\ \prod_{i=1}^{k} e(F_i, M_i) \tag{7}$$

hold for $A = e(A_0, \tilde{A}_0)\ e(A_1, \tilde{A}_1)$ and $B = e(B_0, \tilde{B}_0)\ e(B_1, \tilde{B}_1)$. Output 0, otherwise.

The following theorem is proved in [AHO10]:

Theorem 6. $(\mathcal{G}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *described above provides perfect correctness. It is existentially unforgeable against adaptive chosen-message attack if the SFP assumption holds for* $\mathcal{G}$*.*

Next, we describe some notable properties of the signature scheme:

Partial Perfect Randomizability. Given a signature (Z, R, S, T, U, V, W) one can randomize every element except for Z by applying the sequential randomization technique with small tweak as follows. Define the function $(R', S', T', U', V', W') \leftarrow \mathsf{SigRand}(R, S, T, U, V, W)$, as:

- Randomize (R, S, T) into (R', S', T') as follows.
  - First, if $T = 1$, set $S = 1$ and choose $T \leftarrow \mathbb{G}_2^*$.
  - Then, choose $\varrho \leftarrow \mathbb{Z}_p$ and compute

$$R' = R\,T^{\varrho}, \quad (S', T') \leftarrow \mathsf{Rand}(S G_R^{-\varrho}, T)$$

- Randomize (U, S, T) into (U', S', T') analogously.

Lemma 4. *The above* (R', S', T', U', V', W') *distributes uniformly over* $(\mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_2)^2$ *under constraint that* $e(G_R, R)\, e(S, T) = e(G_R, R')\, e(S', T')$ *and* $e(F_U, U)\, e(V, W) = e(F_U, U')\, e(V', W')$*.*

The claim implies that (S', T', V', W') is information theoretically independent of Z, the message, and the verification key. (In general, the same is true for publishing any two elements from (R', S', T') and (U', V', W') respectively.)

Signature Binding Property. Roughly, it claims that no one but the signer can obtain two signatures which have the same S and V. In the following formal statement, the adversary is allowed to submit both $\vec{M}$ and $\vec{M}^\dagger$ to the signing oracle. That is way the property is not implied by EUF-CMA in general.

Lemma 5. *Under adaptive chosen message attacks, no adversary can output* $(\vec{M}, \sigma)$ *and* $(\vec{M}^\dagger, \sigma^\dagger)$ *such that* $1 = \mathsf{Verify}(vk, \vec{M}, \sigma) = \mathsf{Verify}(vk, \vec{M}^\dagger, \sigma^\dagger)$*,* $\vec{M} \neq \vec{M}^\dagger$*, and* (S, V) *are shared in* σ *and* $\sigma^\dagger$*.*

Hence, in a way, publishing (S, V) together with the verification key works as a commitment on the signature and the message without revealing any information (recall that (S, V) are chosen uniformly in the signing algorithm).

5.3 Variations and Extensions

In this section we describe various extensions and modifications of the above scheme. Due to the space limitations, the ideas are only described briefly and the full description is presented in the full version.

Messages $\in \mathbb{G}_1^k$. When working with asymmetric pairings, it is possible to define a "dual scheme" with a message space $\mathbb{G}_1^k$ (by essentially swapping $\mathbb{G}_1$ and $\mathbb{G}_2$ in the above description).

Messages $\in \mathbb{G}_1^{k_1} \times \mathbb{G}_2^{k_2}$. It is possible to combine the signature schemes with message spaces $\mathbb{G}_1^{k_1}$ and $\mathbb{G}_2^{k_2}$ to obtain a signature scheme whose message space is $\mathbb{G}_1^{k_1} \times \mathbb{G}_2^{k_2}$. Note that this is not trivial, as there is no efficient mappings between $\mathbb{G}_1$ and $\mathbb{G}_2$, and straightforward independent signing allows a forgery. The transformation is applicable to (or required by) the extensions below.

Short Variable-Length Messages. Let $\langle n \rangle$ denote a deterministic encoding of non-negative integer n ($< p$) to an element of $\mathbb{G}_2^*$. By limiting the maximum message length to be $k-1$, for a signature with message space $\mathbb{G}_2^k$, and appending $\langle |\vec{M}| \rangle$ to the input message $\vec{M}$, messages with length less than k can be treated.

Unbounded-Length Messages. For a signature scheme with message space $\mathbb{G}_2^k$, it is possible to sign messages from the space $\mathbb{G}_2^n$, $n > k$, by using a "chaining" technique. The basic idea is to split the message vector into (almost) equal chunks and sign each chunk along with the signature of the previous chunk (or part of it using the signature binding property described above). This is useful when the signer does not know a priori the maximum length of the messages or has to sign her own verification key (e.g. *automorphic signatures*).

Strong One-time Signatures. Dropping the flexible part $e(S,T)$ and $e(V,W)$ from the construction results in a strongly unforgeable one-time signature scheme based on a (weaker) static assumption which is implied by the DBP.

Strongly Unforgeable Signatures. We construct a structure-preserving signature scheme with constant-size signatures that is sEUF-CMA secure. The generic construction, combining a EUF-CMA and a one-time sEUF-CMA signature schemes, is optimized by sharing some parts of the verification keys.

vk **Variations.** We can replace $\{A_i, \tilde{A}_i, B_i, \tilde{B}_i\}_{i=0}^1$ with $A = e(G_R, H^\alpha)$ and $B = e(F_U, H^\beta)$ in a verification key, and use A and B directly in the verification equations (6) and (7). The reason we include a representation of A (and B) in $\mathbb{G}_1$ and $\mathbb{G}_2$ is to address the needs to put the verification key into the base groups. The GS proof system provides zero-knowledge property for statements that do not include elements from $\mathbb{G}_T$ except for $1_{\mathbb{G}_T}$. When WI is of only concern, we do such replacement.

Symmetric Pairings. The signature scheme is also secure when working with symmetric pairings ($\mathbb{G}_1 = \mathbb{G}_2$). The above extensions apply in that case as well.

5.4 Simulatable Signatures

A *simulatable signature scheme* $\mathbf{SSig} = (\mathcal{G}, \mathsf{CrsGen}, \mathsf{KeyGen}, \mathsf{Check}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Sim})$ consists of algorithms for which $\mathbf{Sig}=((\mathcal{G} + \mathsf{CrsGen}), \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ constitutes a regular signature scheme. It is defined in the common reference string (CRS) model and allows to create valid signatures using the trapdoor associated with the CRS. The three algorithms not defined for regular signatures (CrsGen, Check, Sim) are, respectively, for generating a CRS and the associated trapdoor, for checking that a verification key produced by a user is valid, and for simulating a signature on any valid message on behalf of any user using the trapdoor key rather than the corresponding signing key. A simulatable signature is a useful tool in combination with a witness indistinguishable (WI) proof system. Unlike zero-knowledge (ZK) proofs, WI proof system does not accompany a simulator. So when a signature is part of the witness and the signer is corrupt and useless, simulatable signature can provide a correct witness to the entity having the trapdoor.

The notion is introduced in [AO09] but in an informal way dedicated for their purposes. We present a formal treatment and give an efficient construction, but due to the space limitation, we can only sketch the intuition, the security definitions, and the construction details. Full details are presented in [AHO10].

The security properties we require from a simulatable signature scheme are correctness, simulatability, and unforgeability, extended to a multi-user setting where the adversary has access to a signing oracle for all correctly generated verification keys in addition to a proof oracle for simulated signatures on any valid verification key and message. Our construction shares a lot with our basic signature scheme. The main difference is that to sign messages of length k, we need k flexible pairings rather than 1, so the signature is of size $4k + 3$ group elements. The Verify algorithm is defined similarly, with the verification equations being:

$$A = e(G_Z, Z)\; e(G_R, R) \prod_{i=1}^{k} e(G_i, M_i)\; e(S_i, T_i) \quad \text{and} \tag{8}$$

$$B = e(F_Z, Z)\; e(F_U, U) \prod_{i=1}^{k} e(F_i, M_i)\; e(V_i, W_i) \;\; . \tag{9}$$

So, for $k = 1$, the two schemes have the same signature distribution and verification algorithms. The key generation algorithm of the basic scheme is divided into two parts: CrsGen generating the elements on the right side of equations (8)-(9) and KeyGen computing those on the left as well as a signature on the default message (e.g. the all-$1_{\mathbb{G}_2}$ vector). The CRS is, in fact, a commitment key for a trapdoor commitment scheme similar to the one presented in Section 3, whereas any vk is a commitment to the default message. The signing algorithm is quite intricate as it opens the commitment, the signer's vk, to any given message without using the commitment trapdoor. That is why we need k flexible pairings to achieve perfectly random distribution for a signature under the condition that the verification equations are satisfied.

Theorem 7. *The* **SSig** *described above is a perfectly correct signature scheme and signature-simulatable. It is EUF-CMA with WI-simulation in the multi-user setting for $k = 1$ if the SFP assumption holds for $\mathcal{G}$.*

The security for the case of $k > 1$ is shown under a generalization of the SFP assumption and also presented in the full version.

6 Applications of Signatures on Group Elements

This section highlights the benefits of combining structure-preserving signatures on group elements with the GS proof system when building applications. We present the first efficient round-optimal non-committing blind signature scheme which is adaptively secure in the universal-composability framework, efficient group signatures with concurrent join under the strongest security definitions, and efficient anonymous proxy signatures with enhanced anonymity properties.

6.1 UC-Secure Blind Signatures

It has been an open problem to efficiently instantiate Fischlin's [Fis06] framework for UC-secure round-optimal blind signatures. We do so using our signature scheme from Section 5 and a variant of Pedersen commitments [Ped92]. In fact, we use the modification of [HKKL07,AO09] for which the generic construction uses a NIWI proof system and a simulatable signature scheme as it achieves *adaptive* security.

We instantiate the framework as follows: a user commits to a message $m \in \mathbb{Z}_p$, with opening $D = G^r$, as $C = H^m Y^r$ and sends C to the signer. Note that the verification equation for (D, m) being a valid opening is $e(G, C)\, e(D, Y^{-1}) = e(G, H^m)$ which could be viewed as a "pairing-based variant" of Pedersen commitment. The signer signs the commitment c using the simulatable signature scheme from Section 5 and returns the signature to the user. Then, the user computes a NIWI proof of knowledge π of a commitment C to the message m, an opening D of the commitment for that message, and a valid signature on C with respect to the signer's verification key. The user outputs that proof as a blind signature on the message m.

Details of the instantiation can be found in [AHO10]. The signature size is 28 group elements when working with symmetric pairings and 38 group elements with asymmetric, while the total communication complexity is only 8 group elements in both cases.

6.2 Group Signatures

Group signatures have enjoyed much interest since they were introduced by Chaum and van Heyst [Cv91] almost twenty years ago. Most previous constructions, [CS97,ACJT00,BBS04,CL04,BSZ05,BW06,BW07,Gro06] among others, could be viewed as unsatisfactory in some aspect: relying on the random-oracle model, satisfying weaker security definitions, or not being efficient. The

scheme by Groth [Gro07] both is practical and satisfies the strengthened security definitions of [BSZ05]. However, it does not support concurrent join of new users. Using our signature schemes in combination with the GS proof system and an appropriate encryption scheme [Kil06,Sha07], we overcome this shortcoming and construct a group signature scheme under the strongest security definitions which supports concurrent join while achieving comparable efficiency.

Our construction follows a common approach used, e.g., in [CS97,Gro07]. The dynamic join protocol between a group member and the issuer simply consists in the issuer signing the member's verification key. To sign a message m, the member signs the message using her secret key and gives a NIWI proof of knowledge of a verification key, a signature on that key by the issuer, and a signature on the message under that key. For the details of our constructions and further discussions, we refer to the full versions of our papers.

6.3 Anonymous Proxy Signatures

Combined with Groth-Sahai proofs, automorphic signatures enable the first efficient instantiation of anonymous proxy signatures [FP08]. This primitive generalizes (multi-level) proxy signatures [MUO96,BPW03] and group signatures. Consider a setting where users publish signature verification keys, which they have previously registered with an authority. Proxy signatures enable users to delegate others to sign on their behalf; moreover, received rights can be redelegated. Anonymity of proxy signatures guarantees that they neither reveal who signed nor who redelegated. As for group signatures, an *opening authority* can revoke anonymity to deter from misuse. Every valid signature can be opened to registered users (*traceability*) and no coalition even comprising the authorities can produce a signature that wrongfully accuses an honest user (*non-frameability*).

Automorphic signatures allow a straightforward instantiation of the generic construction. To delegate to Bob, Alice signs his public key (and possibly some public attributes). To redelegate to Carol, Bob forwards her the received signature and signs her public key. Carol makes a proxy signature by signing the message and then making a proof of knowledge of the following: Bob's key, Alice's signature on it, her own key, Bob's signature on it, and her signature on the message.[7] Since all of them consist of elements of a bilinear group and validity is expressed as pairing-product equations, Groth-Sahai (GS) proofs apply perfectly. The extraction key is given to the opener who can thus revoke anonymity of a signature by retrieving the public keys of the intermediate delegators and the proxy signer. A signature is verified by checking validity of the GS proof with respect to Alice's public key.

Enhanced Anonymity Guarantees. In the model of [FP08], anonymity holds only w.r.t. the verifier. We show how to protect the privacy of the delegatee and

[7] To guarantee traceability, Carol additionally proves knowledge of certificates from the authority on the public keys. Moreover, to delegate, a user actually signs (a hash value of) an identifier set by the original delegator and his position in the chain in addition to the public key to achieve non-frameability.

the delegators even during delegation. The delegatee remains anonymous if we use the issuing protocol of the blind signature from Section 4.2 for delegation. In the end, the delegatee holds an actual signature on her public key, as in the original scheme, but without the delegator having learned her identity.

The previous delegators can remain anonymous w.r.t. the delegatee as well, as due to the modularity of Groth-Sahai proofs, the "anonymization" of a signature need not be delayed until the proxy signing: instead of forwarding the received delegation chain, a delegator forwards a proof of knowledge of it. The delegatee can then extend the proof by one delegation step, or make a proxy signature; before doing so, she *randomizes* the proof, which prevents linkability of delegations and signatures. By additionally proving knowledge of his public key and signature, the delegator can also hide *his own identity*. Unfortunately, this is not compatible with blind delegation, while hiding the previous delegators is. We refer to [Fuc09] for the details.

Acknowledgments

The second author is supported by EADS, the French ANR-07-TCOM-013-04 PACE Project and the European Commission through the ICT Program under Contract ICT-2007-216646 ECRYPT II.

References

[ACJT00] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 255–270. Springer, August 2000.

[ADW09] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Survey: Leakage-resilience and the bounded-retrieval model. (invited paper to International Conference on Information Theoretic Security), 2009. Available at http://cs.nyu.edu/∼dodis/surveys.html.

[AHO10] Masayuki Abe, Kristiyan Haralambiev, and Miyako Ohkubo. Signing on elements in bilinear groups for modular protocol design. Cryptology ePrint Archive, Report 2010/133, 2010. http://eprint.iacr.org/.

[AO09] Masayuki Abe and Miyako Ohkubo. A framework for universally composable non-committing blind signatures. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 435–450. Springer, December 2009.

[BB04] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer, May 2004.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, August 2004.

[BCC+09] Mira Belenkiy, Jan Camenisch, Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Hovav Shacham. Randomizable proofs and delegatable anonymous credentials. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 108–125. Springer, August 2009.

[BCKL08] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. P-signatures and noninteractive anonymous credentials. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 356–374. Springer, March 2008.

[BGLS03] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 416–432. Springer, May 2003.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 325–341. Springer, February 2005.

[BMW03] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 614–629. Springer, May 2003.

[BNPS03] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3):185–215, June 2003.

[Boy08] Xavier Boyen. The uber-assumption family (invited talk). In Steven D. Galbraith and Kenneth G. Paterson, editors, *PAIRING 2008*, volume 5209 of *LNCS*, pages 39–56. Springer, September 2008.

[BPW03] Alexandra Boldyreva, Adriana Palacio, and Bogdan Warinschi. Secure proxy signature schemes for delegation of signing rights. Cryptology ePrint Archive, Report 2003/096, 2003. http://eprint.iacr.org/.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

[Bra99] Stefan Brands. Rethinking public key infrastructure and digital certificates—building privacy. PhD thesis, Eindhoven Inst. of Tech., The Netherlands, 1999.

[BSZ05] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In Alfred Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 136–153. Springer, February 2005.

[BW06] Xavier Boyen and Brent Waters. Compact group signatures without random oracles. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 427–444. Springer, May / June 2006.

[BW07] Xavier Boyen and Brent Waters. Full-domain subgroup hiding and constant-size group signatures. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007*, volume 4450 of *LNCS*, pages 1–15. Springer, April 2007.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd FOCS*, pages 136–145. IEEE Computer Society Press, October 2001.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218. ACM Press, May 1998.

[CKS09] Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous creden-

tials. In Stanislaw Jarecki and Gene Tsudik, editors, *PKC 2009*, volume 5443 of *LNCS*, pages 481–500. Springer, March 2009.

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 56–72. Springer, August 2004.

[CLY09] Julien Cathalo, Benoît Libert, and Moti Yung. Group encryption: Non-interactive realization in the standard model. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 179–196. Springer, December 2009.

[CS97] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups (extended abstract). In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 410–424. Springer, August 1997.

[Cv91] David Chaum and Eugène van Heyst. Group signatures. In Donald W. Davies, editor, *EUROCRYPT'91*, volume 547 of *LNCS*, pages 257–265. Springer, April 1991.

[DF02] Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In Yuliang Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 125–142. Springer, December 2002.

[DHLAW10] Yevgeniy Dodis, Kristiyan Haralambiev, Adriana Lopez-Alt, and Daniel Wichs. Efficient public-key cryptography in the presence of key leakage. Cryptology ePrint Archive, Report 2010/154, 2010. http://eprint.iacr.org/.

[DN02] Ivan Damgård and Jesper Buus Nielsen. Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 581–596. Springer, August 2002.

[ElG86] Taher ElGamal. On computing logarithms over finite fields. In Hugh C. Williams, editor, *CRYPTO'85*, volume 218 of *LNCS*, pages 396–402. Springer, August 1986.

[Fis06] Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 60–77. Springer, August 2006.

[FO97] Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 16–30. Springer, August 1997.

[FP08] Georg Fuchsbauer and David Pointcheval. Anonymous proxy signatures. In Rafail Ostrovsky, Roberto De Prisco, and Ivan Visconti, editors, *SCN 08*, volume 5229 of *LNCS*, pages 201–217. Springer, September 2008.

[FP09] Georg Fuchsbauer and David Pointcheval. Proofs on encrypted values in bilinear groups and an application to anonymity of signatures. In Hovav Shacham and Brent Waters, editors, *PAIRING 2009*, volume 5671 of *LNCS*, pages 132–149. Springer, August 2009.

[FPV09] Georg Fuchsbauer, David Pointcheval, and Damien Vergnaud. Transferable constant-size fair e-cash. In Juan A. Garay, Atsuko Miyaji, and Akira Otsuka, editors, *CANS 09*, volume 5888 of *LNCS*, pages 226–247. Springer, December 2009.

[FS01] Jun Furukawa and Kazue Sako. An efficient scheme for proving a shuffle. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 368–387. Springer, August 2001.

[Fuc09] Georg Fuchsbauer. Automorphic signatures in bilinear groups and an application to round-optimal blind signatures. Cryptology ePrint Archive, Report 2009/320, 2009. http://eprint.iacr.org/.

[Fuc10] Georg Fuchsbauer. Commuting signatures and verifiable encryption and an application to non-interactively delegatable credentials. Cryptology ePrint Archive, Report 2010/233, 2010. http://eprint.iacr.org/.

[FV10] Georg Fuchsbauer and Damien Vergnaud. Fair blind signatures without random oracles. In Daniel J. Bernstein and Tanja Lange, editors, *AFRICACRYPT*, volume 6055, pages 16–33. Springer, 2010.

[GH08] Matthew Green and Susan Hohenberger. Universally composable adaptive oblivious transfer. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 179–197. Springer, December 2008.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 97–111. Springer, August 2006.

[GPS08] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[GQ88] Louis C. Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both trasmission and memory. In C. G. Günther, editor, *EUROCRYPT'88*, volume 330 of *LNCS*, pages 123–128. Springer, May 1988.

[Gro06] Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, December 2006.

[Gro07] Jens Groth. Fully anonymous group signatures without random oracles. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 164–180. Springer, December 2007.

[Gro09a] Jens Groth. Homomorphic trapdoor commitments to group elements. Cryptology ePrint Archive, Report 2009/007, 2009. http://eprint.iacr.org/.

[Gro09b] Jens Groth. Linear algebra with sub-linear zero-knowledge arguments. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 192–208. Springer, August 2009.

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, April 2008.

[HKKL07] Carmit Hazay, Jonathan Katz, Chiu-Yuen Koo, and Yehuda Lindell. Concurrently-secure blind signatures without random oracles or setup assumptions. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 323–341. Springer, February 2007.

[Kil06] Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 581–600. Springer, March 2006.

[KP06] Sébastien Kunz-Jacques and David Pointcheval. About the security of MTI/C0 and MQV. In Roberto De Prisco and Moti Yung, editors, *SCN 06*, volume 4116 of *LNCS*, pages 156–172. Springer, September 2006.

[KY05] Aggelos Kiayias and Moti Yung. Group signatures with efficient concurrent join. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 198–214. Springer, May 2005.

[KZ06] Aggelos Kiayias and Hong-Sheng Zhou. Concurrent blind signatures without random oracles. In Roberto De Prisco and Moti Yung, editors, *SCN 06*, volume 4116 of *LNCS*, pages 49–62. Springer, September 2006.

[Lip03] Helger Lipmaa. Verifiable homomorphic oblivious transfer and private equality test. In Chi-Sung Laih, editor, *ASIACRYPT 2003*, volume 2894 of *LNCS*, pages 416–433. Springer, November / December 2003.

[LRSW00] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In Howard M. Heys and Carlisle M. Adams, editors, *SAC 1999*, volume 1758 of *LNCS*, pages 184–199. Springer, August 2000.

[LV08] Benoît Libert and Damien Vergnaud. Multi-use unidirectional proxy resignatures. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM CCS 08*, pages 511–520. ACM Press, October 2008.

[MUO96] Masahiro Mambo, Keisuke Usuda, and Eiji Okamoto. Proxy signatures for delegating signing operation. In *ACM CCS 96*, pages 48–57. ACM Press, March 1996.

[Nao03] Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, August 2003.

[Nef01] C. Andrew Neff. A verifiable secret shuffle and its application to e-voting. In *ACM CCS 01*, pages 116–125. ACM Press, November 2001.

[Oka06] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 80–99. Springer, March 2006.

[OU98] Tatsuaki Okamoto and Shigenori Uchiyama. A new public-key cryptosystem as secure as factoring. In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 308–318. Springer, May / June 1998.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238. Springer, May 1999.

[Ped92] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 129–140. Springer, August 1992.

[RS09] Markus Rückert and Dominique Schröder. Security of verifiably encrypted signatures and a construction without random oracles. In Hovav Shacham and Brent Waters, editors, *PAIRING 2009*, volume 5671 of *LNCS*, pages 17–34. Springer, August 2009.

[Sha07] Hovav Shacham. A cramer-shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Report 2007/074, 2007. http://eprint.iacr.org/.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, May 1997.