J Cryptol (2016) 29:363–421
DOI: 10.1007/s00145-014-9196-7

Journal of
CRYPTOLOGY

Structure-Preserving Signatures and Commitments to Group Elements

Masayuki Abe

NTT Secure Platform Laboratories, 3-9-11 Midori-cho, Musashino-shi, Tokyo 180-8585, Japan
abe.masayuki@lab.ntt.co.jp

Georg Fuchsbauer*

Institute of Science and Technology Austria, Am Campus 1, 3400 Klosterneuburg, Austria
georg.fuchsbauer@ist.ac.at

Jens Groth

Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK
j.groth@ucl.ac.uk

Kristiyan Haralambiev†

IBM Research - Zurich, Zurich, Switzerland
kha@zurich.ibm.com

Miyako Ohkubo‡

Security Fundamentals Laboratory, NSR, NICT, 4-2-1 Nukui-Kitamachi, Koganei, Tokyo184-8795, Japan
m.ohkubo@nict.go.jp

Communicated by Nigel Smart.

Received 20 August 2013
Online publication 27 January 2015

Abstract. A modular approach to constructing cryptographic protocols leads to simple designs but often inefficient instantiations. On the other hand, ad hoc constructions may yield efficient protocols at the cost of losing conceptual simplicity. We suggest a new design paradigm, *structure-preserving cryptography*, that provides a way to construct modular protocols with reasonable efficiency while retaining conceptual simplicity. A cryptographic scheme over a bilinear group is called structure-preserving if its public inputs and outputs consist of elements from the bilinear groups and their consistency can be verified by evaluating pairing-product equations. As structure-preserving schemes smoothly interoperate with each other, they are useful as building blocks in modular design of cryptographic applications. This paper introduces structure-preserving commitment and signature schemes over bilinear groups with several desirable properties.

* Work done while at École normale supérieure, Paris, France.

† Work done while at NYU and visiting NTT Information Sharing Platform Laboratories.

‡ Work done while working in NTT Information Sharing Platform Laboratories.

The commitment schemes include homomorphic, trapdoor and length-reducing commitments to group elements, and the structure-preserving signature schemes are the first ones that yield constant-size signatures on multiple group elements. A structure-preserving signature scheme is called automorphic if the public keys lie in the message space, which cannot be achieved by compressing inputs via a cryptographic hash function, as this would destroy the mathematical structure we are trying to preserve. Automorphic signatures can be used for building certification chains underlying privacy-preserving protocols. Among a vast number of applications of structure-preserving protocols, we present an efficient round-optimal blind-signature scheme and a group signature scheme with an efficient and concurrently secure protocol for enrolling new members.

Keywords. Structure-preserving cryptography, Structure-preserving signatures, Automorphic signatures, Homomorphic commitments, Groth–Sahai proofs, Group signatures, Blind signatures.

1. Introduction

1.1. *Structure-Preserving Cryptography*

Cryptographic protocols often use modular constructions that combine general building blocks such as commitments, encryption, signatures, and zero-knowledge proofs. Modular design is useful to show the feasibility of realizing a particular security goal and may lead to simpler security proofs that are less error-prone than seen in ad hoc constructions. On the other hand, modular design can incur a significant overhead, and in feasibility proofs efficient instantiations are often left as the next challenge. This challenge is often solved by finding a "cleverly crafted" efficient solution for the specific security goal. However, modular constructions make it easier to design and understand protocols. It is therefore desirable to have a framework of interoperable building blocks with efficient instantiations such that we can design protocols that are both modular and efficient at the same time.

Since the seminal works in [24,63,86], bilinear groups have been widely used for constructing efficient cryptographic protocols. However, protocols defined over bilinear group are not necessarily compatible with each other as they may involve both group and field elements in different places as well as additional cryptographic primitives such as collision-resistant hash functions for instance.

We propose *structure-preserving cryptography* as an approach for efficiently instantiating modular constructions over bilinear groups $\Lambda := (p, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, e, G, \tilde{G})$ where $\mathbb{G}$, $\tilde{\mathbb{G}}$ and $\mathbb{G}_T$ are groups of prime order p generated by G, $\tilde{G}$ and $e(G, \tilde{G})$, respectively, with a bilinear map $e\colon \mathbb{G} \times \tilde{\mathbb{G}} \to \mathbb{G}_T$. In structure-preserving cryptography, building blocks are designed over common bilinear groups Λ such that

- all public objects such as public keys, messages, signatures, commitments and ciphertexts are elements of the groups $\mathbb{G}$ and $\tilde{\mathbb{G}}$, and
- verifying relations of interest, such as signature verification or opening of a commitment, can be done by performing group operations and evaluating pairing-product equations of the form

$$\prod_i \prod_j e(X_i, \tilde{Y}_j)^{c_{i,j}} = 1,$$

where $c_{i,j} \in \mathbb{Z}$ are constants specified by the scheme.

We call cryptographic schemes satisfying these conditions structure-preserving. It will sometimes be useful to consider also *relaxed* structure preservation where public elements in the target group $\mathbb{G}_T$ are permitted as well and where pairing-product equations may be of the form $\prod_i \prod_j e(X_i, \tilde{Y}_j)^{c_{i,j}} = T$ with $T \in \mathbb{G}_T$.

The properties defining structure preservation are preserved by modular constructions. Namely, a scheme built by modularly combining (relaxed) structure-preserving building blocks is (relaxed) structure-preserving as well. They therefore offer strong compatibility and modularity. On the other hand, the restrictive properties make it more challenging to design structure-preserving schemes. In particular, structure-preserving cryptography inherently prohibits the use of collision-resistant hash functions that break the underlying mathematical structure of the bilinear groups.

Combining non-interactive proofs with other primitives is a typical approach in modular construction of secure cryptographic protocols. A classical way of realizing efficient instantiations is to rely on the random-oracle heuristic [18] for non-interactive proofs—or to directly use *interactive* assumptions like (variations of) the LRSW assumption [78] and "one-more" assumptions [17]. Due to a series of criticisms starting with [37], 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) intractability assumptions [80]. In [59], Groth and Sahai presented the first (and currently the only) efficient non-interactive proof system based on standard assumptions in bilinear groups. Their proof system, called GS for short, exerts its full power as a non-interactive proof-of-knowledge system when the proof statement is a set of relations described by pairing-product equations, for which the witnesses are group elements in the source groups, that is $\mathbb{G}$ and $\tilde{\mathbb{G}}$. Due to these limitations, however, many existing cryptographic schemes cannot be modularly combined with GS proofs. In contrast, structure-preserving schemes are defined so that they are compatible with the GS proof system. Accordingly, by using GS proofs, one can efficiently prove one's knowledge about the witness for relations of interest in structure-preserving schemes.

We address two major building blocks in cryptographic protocol design, commitment schemes and signature schemes, and present their structure-preserving instantiations with several useful properties as explained in the following.

1.2. *Homomorphic Trapdoor Commitments*

A non-interactive commitment scheme allows a sender to create a commitment to a message. The commitment *hides* the message but the sender may later choose to *open* the commitment to the message. A commitment is *bound* to a message in the sense that a commitment cannot be opened to two different messages. On top of the fundamental properties of hiding and binding, a commitment may have other desirable features. In a *trapdoor* commitment scheme [60,83], a certain piece of trapdoor information makes it possible to circumvent the binding property and open a commitment to an arbitrary message. In a *homomorphic* commitment scheme, messages and commitments belong to abelian groups, and by multiplying two commitments, we obtain a commitment to the product of the committed messages. Finally, a commitment scheme is often required to be *length-reducing* such that the commitment is shorter than the message.

An example that provides all those properties is a generalization of Pedersen commitments [83] where a message is a vector of values in $\mathbb{Z}_p$ and a commitment consists of only one group element. Such commitments have been found useful in contexts such as mix-nets, voting, digital credentials, blind signatures, leakage-resilient one-way functions and zero-knowledge proofs [11,30,50,66,77,81].

We present structure-preserving commitment schemes whose public keys, messages, commitments and openings are elements of bilinear groups, and whose opening is verified by evaluating pairing-product equations. Our commitments are trapdoor and homomorphic, and some of them are length-reducing as well. The attributes that discriminate our constructions are the types of bilinear groups used and the groups that messages and commitments belong to. (See Table 1 on page 21 for a summary.)

- The first and the second homomorphic trapdoor commitment schemes are length-reducing by mapping vectors of source-group elements to a constant number of target-group elements and thus relaxed structure-preserving.
- The third homomorphic trapdoor commitment scheme is *strictly* structure-preserving, which means both messages and commitments consists of source-group elements. They are, however, not length-reducing.
- The last commitment scheme takes messages from $\mathbb{Z}_p$ and maps them to a single source-group element. This scheme is by definition not structure-preserving though all other properties are provided. We include it here for its usefulness as a building block for applications.

Our commitment schemes can be used to build structure-preserving one-time signatures, as we demonstrate in Sect. 4. The first two length-reducing schemes are useful in reducing the size of zero-knowledge arguments. Groth [58] showed that the square-root-size zero-knowledge arguments in [57] can be reduced to cubic-root-size zero-knowledge arguments by using our homomorphic trapdoor commitments. We explore this topic in Sect. 3.1.

There are follow-up works about structure-preserving commitments. In [9], it is proved that strictly structure-preserving commitment schemes cannot yield commitments that are shorter than messages; a commitment to a k-element message must have more than k elements itself. This should be contrasted with the relaxed structure-preserving commitment schemes we give, where a commitment to a k-element message consists of a small constant number of target-group elements.

1.3. *Signatures*

A signature scheme is called structure-preserving if the public verification keys, messages, and signatures are source-group elements of bilinear groups, and the verification of a signature consists of evaluating pairing-product equations. It is called automorphic if in addition its verification keys lie in its message space. There are many applications using combinations of digital signatures and non-interactive zero-knowledge proofs of knowledge such as blind signatures [10,43], group signatures [16,19,68], anonymous credential systems [15], verifiably encrypted signatures [26,85], non-interactive group encryption [38] and so on, and structure-preserving signatures are ideally suited for these applications.

Research on structure-preserving signature schemes was initiated by Groth [54], who gave the first feasibility result based on the decision linear assumption (DLIN) [23]. His structure-preserving signature scheme yields a signature of size $\mathcal{O}(k)$ when the message consists of k group elements. While it is remarkable that the security can be based on a simple standard assumption, the scheme is not practical due to its large constant factor. Based on the q-Hidden LRSW assumption for asymmetric bilinear groups, Green and Hohenberger [53] presented a structure-preserving signature scheme that only provides security against random-message attacks. Unfortunately, an extension to chosen-message attack security is not known. In [38], Cathalo et al. gave a scheme with relaxed structure preservation based on a combination of the Hidden Strong Diffie–Hellman Assumption (HSDH), the Flexible Diffie–Hellman Assumption, and the DLIN assumption. Their signature consists of $9k + 4$ group elements for a k-element message, and it was left as an open problem to construct constant-size signatures. There are also several signature schemes, such as [15,22,31,35], where validity is defined via pairing-product equations, but whose signatures do not only contain group elements.

We present several constructions of structure-preserving signatures with high efficiency and useful properties.

- *Structure-PreservingOne-Time Signatures.* We construct two signature schemes that are existentially unforgeable when the adversary is only allowed one signing query. The schemes are currently the most efficient in the literature, and their security follows from the decision Diffie–Hellman and the decision linear assumptions, respectively.
- *Constant-Size Structure-Preserving Signatures.* We construct the first constant-size structure-preserving signature scheme. A signature consists of 7 group elements independently of the message length and the verifier needs to check two pairing-product equations. Existential unforgeability against adaptive chosen-message attacks is proven under a new non-interactive assumption called the Simultaneous Flexible Pairing Assumption (SFP).
- *Automorphic Signatures.* We construct the first automorphic signature scheme, whose signatures consist of 5 group elements. We prove the scheme existentially unforgeable against adaptive chosen-message attacks under a variant of the Strong Diffie–Hellman assumption [22].

After the publication of [5], structure-preserving signatures have been intensively studied. Several constructions over asymmetric bilinear groups, i.e., where $\mathbb{G} \neq \tilde{\mathbb{G}}$, are presented in [6]. The latter shows that there is a scheme whose signature consists of only 3 group elements when the security is directly proven in the generic bilinear-group model. It also presents a scheme with 4-element signatures, which is 3 group elements fewer compared to our scheme in Sect. 5.1. Its security is based on a non-interactive assumption that is, however, not known to be as tight as the discrete-logarithm assumption when assessed in the generic bilinear-group model. This is the case for the assumption (q-SFP in Sect. 2.5) that implies security of our scheme.

Significant theoretical advances are made in [2,3], which present structure-preserving signature schemes based on compact and static, i.e., not q-type, assumptions such as the decision linear assumption. The schemes are over symmetric and asymmetric bilinear groups and yield signatures consisting of 11 to 14 group elements.

1.4. *Applications*

The usefulness of structure-preserving cryptography as a design paradigm is demonstrated by a growing list of applications including round-optimal blind signatures [5,49], group signatures with concurrent join [5,49,72,73], homomorphic signatures [14,71], anonymous proxy signatures [47], delegatable anonymous credentials [46], direct anonymous attestation [20], transferable e-cash [21,48], conditional e-cash [89], compact verifiable shuffles [39], network coding [13], oblivious transfer [1,33,53], chosen-ciphertext-secure encryption [3,34,62] and many more. Among this vast number of applications, we present group signatures and blind signatures in this paper.

1.4.1. *Group Signatures with Concurrent Join*

We give a modular structure-preserving construction of group signatures [41] supporting a concurrent join procedure [68] for enrolling new members.

A group signature scheme is a classical primitive ensuring user anonymity. It allows members that were enrolled by a group manager to sign on behalf of a group without revealing their identity. To prevent misuse, anonymity can be revoked by an authority. There are numerous constructions of group signature schemes in the literature providing different properties. However, previous constructions are not in the standard model, do not provide a concurrent join procedure, lack some important property like non-frameability or are inefficient. The scheme in [68] is the first one that allows efficient concurrent join, but its security relies on the random-oracle model [18]. The scheme in [12] is non-frameable but only allows new members to join sequentially and is based on strong interactive assumptions. Both [28,29] provide efficiency with reasonable assumptions but the group manager enrolling members knows their secret keys and can thus frame them by creating signatures using their keys. The scheme in [55] is non-frameable but does not allow concurrent join. More recent papers, such as [42,75,76], focus on advanced properties leaving one or more of the above issues unaddressed.

1.4.2. *Round-Optimal Blind Signatures*

Blind signatures, introduced by Chaum [40], allow a user to obtain a signature on a message such that the signer cannot relate the resulting signature to the execution of the signing protocol. They were formalized by [64,84] and practical schemes without random oracles have been constructed in e.g., [36,66,67,82]. However, all these schemes require more than one round (i.e., two moves) of communication between the user and the signer to issue a blind signature. This is even the case for most instantiations in the random-oracle model, an exception being Chaum's scheme proved secure in [17] under an interactive assumption.

In [43], Fischlin gives a generic construction of *round-optimal* blind signatures in the common reference string (CRS) model; the signing protocol consists of one message from the user to the signer and one response by the signer. This immediately implies *concurrent* security, an explicit goal in other works such as [61]. Before our work, a practical instantiation of round-optimal blind signatures in the standard model was an open problem. Using our automorphic signature scheme, we provide the first efficient instantiation, which is the basis for commuting signatures and verifiable encryption in [46].

1.5. *Correspondence to Preliminary Papers and Organization*

This paper is based on three papers [7,45,56] submitted separately to CRYPTO 2010 and presented as a merged paper [5], in which the term "structure-preserving signatures" is introduced. In [56], Groth presented the first homomorphic trapdoor commitments to group elements which are, moreover, length-reducing. Fuchsbauer [45] gave the first efficient structure-preserving signatures and used them to efficiently implement round-optimal blind signatures in the standard model. Abe et al. [7] then gave the first constant-size signature scheme on vectors of general group elements and constructed a group signature scheme with concurrently secure enrollment of new members.

Section 2 introduces notations, security notions and building blocks used in this paper. It includes, in Sect. 2.7, a useful technique that appeared in [7]. Section 3 features several homomorphic trapdoor commitment schemes, which originate from [7,56]. Section 4 presents one-time signature schemes from [7], which can be a warm-up to the fully fledged structure-preserving signature scheme in Sect. 5. In Sect. 6, we present the automorphic signature scheme from [45] with a technique from [8] to extend the message space. Finally, Sect. 7 contains applications discussed above: the group signature scheme and the blind-signature scheme which originally appeared in [7] and [45], respectively.

2. Preliminaries

2.1. *Notation*

For a set S, $x \leftarrow S$ denotes assigning to x a uniformly random value in S. Similarly, $x_1, \ldots, x_k \leftarrow S$ means independent uniformly random selection of k elements from S. For a probabilistic algorithm A, we write $y := A(x; r)$ for assigning y the value of the output of the algorithm when running it on input x using randomness r. We write $y \leftarrow A(x)$ for the process of picking uniformly random r and setting $y := A(x; r)$.

By $\Pr[y \leftarrow \mathrm{Exp}(x) : \mathrm{Cond}(y)]$, we denote the probability that the condition Cond holds for the output y of running the experiment Exp on some input x. The probability is taken over all coin flips used in the experiment Exp. Typically, the input x will be of the form 1^λ, where $\lambda \in \mathbb{N}$ is a security parameter. We say $f : \mathbb{N} \rightarrow [0, 1]$ is negligible if $f(\lambda) = \lambda^{-\omega(1)}$ and $g : \mathbb{N} \rightarrow [0, 1]$ is overwhelming if $g(\lambda) = 1 - f(\lambda)$ for some negligible function f. When defining security, we require the attacker's success probability to be negligible as a function of the security parameter.

We will work over groups $\mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T$ of prime order p, which we denote multiplicatively. We will in general denote group elements by capital letters, i.e., $R \in \mathbb{G}$, $\tilde{S} \in \tilde{\mathbb{G}}$, $T \in \mathbb{G}_T$. Integers modulo p will be denoted by lower case or Greek letters. We define $\mathbb{G}^* = \mathbb{G}\backslash\{1\}$, $\tilde{\mathbb{G}}^* = \tilde{\mathbb{G}}\backslash\{1\}$ and $\mathbb{G}_T^* = \mathbb{G}_T\backslash\{1\}$. When $\vec{X}$ is a tuple of group elements, $|\vec{X}|$ denotes the number of elements in $\vec{X}$.

When a group element is given as input to a function, its group membership must be tested. If the test fails, the function should output a special symbol that means rejection of the input. For conciseness of the description, we treat this procedure as implicit throughout the paper.

2.2. *Commitment Schemes*

A commitment scheme allows a sender to commit to a secret message *msg*. Later the sender may open the commitment and reveal the value *msg* to the receiver. We focus on non-interactive commitment schemes where the sender and receiver do not need to interact to commit or to open commitments; both the commitment and the opening are bitstrings generated by the sender without interacting with the receiver.

We rely on a trusted setup phase where joint system parameters are generated and a commitment key is produced. We deliberately separate the setup phase in two parts Setup and Key to distinguish joint system parameters (which in our schemes will contain a description of bilinear groups that may be shared with other protocols such as signature schemes, encryption schemes) and the commitment key, which is specific to the commitment scheme.

Definition 1. (*Trapdoor Commitment Scheme*) We define a non-interactive trapdoor commitment scheme C as a tuple of polynomial-time algorithms C = (Setup, Key, Com, Vrf, Sim, Equiv) in which:

- $gk \leftarrow \mathsf{Setup}(1^\lambda)$ is a common-parameter generator that takes security parameter λ and outputs a set of common parameters, gk.
- $(ck, tk) \leftarrow \mathsf{Key}(gk)$ is a key generator that takes gk as input and outputs a commitment key ck and a trapdoor key tk. The commitment key ck determines the message space $\mathcal{M}_{ck}$, the commitment space $\mathcal{C}_{ck}$ and the opening space $\mathcal{O}_{ck}$.
- $(com, open) \leftarrow \mathsf{Com}(ck, msg)$ is a commitment algorithm that takes ck and message $msg \in \mathcal{M}_{ck}$ and outputs a commitment $com \in \mathcal{C}_{ck}$ and an opening $open \in \mathcal{O}_{ck}$.
- $1/0 \leftarrow \mathsf{Vrf}(ck, com, msg, open)$ is a verification algorithm that takes ck, $com \in \mathcal{C}_{ck}$, $msg \in \mathcal{M}_{ck}$ and $open \in \mathcal{O}_{ck}$ as input and outputs 1 or 0 representing acceptance or rejection, respectively.
- $(com, ek) \leftarrow \mathsf{Sim}(ck)$ takes commitment key ck and outputs commitment $com \in \mathcal{C}_{ck}$ and equivocation key ek.
- $open \leftarrow \mathsf{Equiv}(ck, msg, ek, tk)$ takes ck, ek, tk and $msg \in \mathcal{M}_{ck}$ as input and returns an opening $open$.

For correctness, it must hold that for all $\lambda \in \mathbb{N}$:

$$\Pr\left[\begin{array}{l} gk \leftarrow \mathsf{Setup}(1^\lambda);\ (ck, tk) \leftarrow \mathsf{Key}(gk) \\ msg \leftarrow \mathcal{M}_{ck};\ (com, open) \leftarrow \mathsf{Com}(ck, msg) \end{array} : 1 \leftarrow \mathsf{Vrf}(ck, com, msg, open)\right] = 1.$$

For perfect trapdoor commitment schemes, the experiment $\{gk \leftarrow \mathsf{Setup}(1^\lambda); (ck, tk) \leftarrow \mathsf{Key}(gk); msg \leftarrow \mathcal{M}_{ck}; (com, open) \leftarrow \mathsf{Com}(ck, msg); (com', ek) \leftarrow \mathsf{Sim}(ck);$

$open' \leftarrow \mathsf{Equiv}(ck, msg, ek, tk)\}$ must yield identical probability distributions for $(ck, msg, com, open)$ and $(ck, msg, com', open')$.

In this paper, we consider Vrf as deterministic and say that $(msg, com, open)$ is *valid* with respect to ck if $1 \leftarrow \mathsf{Vrf}(ck, com, msg, open)$. It follows from the correctness of the commitment scheme that in a perfect trapdoor commitment scheme, the equivocation of a trapdoor commitment will be accepted by the verifier.

Commitment schemes must be hiding and binding. Since commitments generated by Sim do not contain any information about the message, a perfect trapdoor commitment scheme is perfectly hiding in the sense that legitimate commitments do not reveal anything about the message. For the binding property, we use the following standard definition.

Definition 2. (*Binding*) A commitment scheme is computationally binding if for any probabilistic polynomial-time adversary $\mathcal{A}$

$$\Pr\left[\begin{array}{l} gk \leftarrow \mathsf{Setup}(1^\lambda);\, ck \leftarrow \mathsf{Key}(gk) \\ (com, msg, open, msg', open') \leftarrow \mathcal{A}(ck) \end{array} : \begin{array}{l} msg \neq msg' \\ 1 \leftarrow \mathsf{Vrf}(ck, com, msg, open) \\ 1 \leftarrow \mathsf{Vrf}(ck, com, msg', open') \end{array}\right]$$

is negligible.

Definition 1 follows a typical definition of a trapdoor commitment scheme with a simulation algorithm Sim that generates fake commitments. A slightly stronger definition referred to as a chameleon hash [69] demands Equiv to equivocate legitimate commitments generated by Com.

Definition 3. (*Homomorphic*) A commitment scheme is homomorphic if for any correctly generated ck, the message, commitment and opening spaces are abelian groups with binary operations "$\cdot$," "$\odot$" and "$\otimes$," respectively, and for all $(msg, com, open)$ and $(msg', com', open')$ valid with respect to ck, it holds that $1 \leftarrow \mathsf{Vrf}(ck, com \cdot com', msg \odot msg', open \otimes open')$.

2.3. *Digital Signatures*

Definition 4. (*Digital Signature Scheme*) A digital signature scheme SIG is a quadruple of efficient algorithms $(\mathsf{Setup}, \mathsf{Key}, \mathsf{Sign}, \mathsf{Vrf})$ such that

- $gk \leftarrow \mathsf{Setup}(1^\lambda)$ is a common-parameter generator that takes security parameter λ and outputs a set of common parameters gk.
- $(vk, sk) \leftarrow \mathsf{Key}(gk)$ is a key generation algorithm that takes common parameters gk and generates a verification key vk and a signing key sk. The verification key vk defines the message space $\mathcal{M}$.
- $\sigma \leftarrow \mathsf{Sign}(sk, msg)$ is a signature-generation algorithm that computes a signature σ for input message $msg \in \mathcal{M}$ by using signing key sk.
- $0/1 \leftarrow \mathsf{Vrf}(vk, msg, \sigma)$ is a signature-verification algorithm that outputs 1 for acceptance or 0 for rejection.

We require the signature scheme to be correct, i.e., for any (vk, sk) generated by Key, any message $msg \in \mathcal{M}$ and any signature σ output by $\mathsf{Sign}(sk, msg)$ the verification $\mathsf{Vrf}(vk, msg, \sigma)$ outputs 1.

We use the standard notion of existential unforgeability against adaptive chosen-message attacks [52] (EUF-CMA for short), which says it is not possible to forge a signature on a previously unsigned message.

Definition 5. (*Existential Unforgeability against Adaptive Chosen-Message Attacks*) A signature scheme is existentially unforgeable against adaptive chosen-message attacks if for any probabilistic polynomial-time adversary $\mathcal{A}$

$$\Pr\left[\begin{array}{l} gk \leftarrow \mathsf{Setup}(1^\lambda);\ (vk, sk) \leftarrow \mathsf{Key}(gk) \\ (m^\star, \sigma^\star) \leftarrow \mathcal{A}^{\mathsf{Sign}(sk,\cdot)}(vk) \end{array} : \begin{array}{l} m^\star \notin Q_m \\ 1 \leftarrow \mathsf{Vrf}(vk, m^\star, \sigma^\star) \end{array}\right]$$

is negligible, where Q_m is the set of messages that were queried to the signing oracle $\mathsf{Sign}(sk, \cdot)$.

By requiring $(m^\star, \sigma^\star) \notin Q_{m,\sigma}$, where $Q_{m,\sigma}$ are the pairs of messages and signatures observed by $\mathsf{Sign}(sk, \cdot)$, we obtain the notion of Strong EUF-CMA (denoted by sEUF-CMA for short). This ensures that even if a message has been signed before, it is not possible to forge a new different signature on the message.

2.4. *Bilinear Groups*

We say $\mathcal{G}$ is a bilinear-group generator if on input security parameter λ returns the description of a bilinear group $\Lambda = (p, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, e, G, \tilde{G}) \leftarrow \mathcal{G}(1^\lambda)$ with the following properties:

- $\mathbb{G}$, $\tilde{\mathbb{G}}$ and $\mathbb{G}_T$ are groups of prime order p, whose bit-length is λ.
- $e\colon \mathbb{G} \times \tilde{\mathbb{G}} \to \mathbb{G}_T$ is a bilinear map, that is $e(U^a, \tilde{V}^b) = e(U, \tilde{V})^{ab}$ for all $U \in \mathbb{G}, \tilde{V} \in \tilde{\mathbb{G}}, a, b \in \mathbb{Z}_p$.
- G and $\tilde{G}$ are uniformly chosen generators of $\mathbb{G}$ and $\tilde{\mathbb{G}}$, and $e(G, \tilde{G})$ generates $\mathbb{G}_T$.
- There are efficient algorithms for computing group operations, evaluating the bilinear map, comparing group elements and deciding membership of the groups. We refer to these as *generic* operations.

Galbraith et al. [51] distinguish between 3 types of bilinear-group generators. Type I groups have $\mathbb{G} = \tilde{\mathbb{G}}$ and are called *symmetric* bilinear groups. By $\mathcal{G}_{\mathsf{sym}}$, we denote a group generator that takes security parameter λ and outputs a description of a symmetric bilinear group $\Lambda = (p, \mathbb{G}, \mathbb{G}_T, e, G)$. Type II and Type III groups, which are referred to as *asymmetric* bilinear groups, have $\mathbb{G} \neq \tilde{\mathbb{G}}$, and we may sometimes write $\Lambda \leftarrow \mathcal{G}_{\mathsf{asym}}(1^\lambda)$ to emphasize when we are generating asymmetric bilinear groups. Type II groups have an efficiently computable homomorphism $\psi : \tilde{\mathbb{G}} \to \mathbb{G}$, while Type III groups do not have efficiently computable homomorphisms in either direction. When we need to explicitly discriminate Λ in Type I, II and III, we write Λ_{sym}, Λ_{xdh} and Λ_{sxdh}, respectively.

The commitment and signature schemes in this work will have a common setup phase which generates common parameters gk. These will always consist of a description of a bilinear group $\Lambda \leftarrow \mathcal{G}(1^\lambda)$ and in some cases contain additional generators. This definitional approach means that many different structure-preserving protocols may use the same setup gk and thus work over the same bilinear group, which is what makes them interoperable and easy to combine.

Note that in the symmetric case Λ includes a generator G, and in the asymmetric case, it includes two generators G and $\tilde{G}$. Some constructions and assumptions described for asymmetric bilinear groups can accommodate the symmetric case by considering $\mathbb{G} = \tilde{\mathbb{G}}$ and $G = \tilde{G}$. Some applications require $\tilde{G} \neq G$ in symmetric bilinear groups. In these cases, we pick a random generator, say H, and set $\tilde{G} = H$. A note will be given when such a treatment is necessary.

2.5. *Assumptions*

Our commitment and signature schemes rely on different assumptions regarding the bilinear groups we use. Most of the assumptions have reductions from the well-known decisional assumptions DDH and DLIN, but we will also rely on two new computational assumptions q-SFP and q-ADH-SDH, which we will describe later in the section. All assumptions are defined relative to a group generator. Therefore, every construction in the succeeding sections assumes that there exists a group generator for which relevant assumptions hold.

Assumption 1. (*Decision Diffie–Hellman Assumption* (DDH)) The decision Diffie–Hellman assumption holds in $\mathbb{G}$ relative to $\mathcal{G}_{\mathsf{asym}}$ if for all probabilistic polynomial-time $\mathcal{A}$

$$\left|\Pr\left[\begin{array}{r}\Lambda \leftarrow \mathcal{G}_{\mathsf{asym}}(1^\lambda);\, a, b \leftarrow \mathbb{Z}_p :\\ 1 \leftarrow \mathcal{A}(\Lambda, G^a, G^b, G^{ab})\end{array}\right] - \Pr\left[\begin{array}{r}\Lambda \leftarrow \mathcal{G}_{\mathsf{asym}}(1^\lambda);\, a, b, c \leftarrow \mathbb{Z}_p :\\ 1 \leftarrow \mathcal{A}(\Lambda, G^a, G^b, G^c)\end{array}\right]\right|$$

is negligible.

The decision Diffie–Hellman assumption in $\tilde{\mathbb{G}}$ is defined analogously. In the bilinear-group setting the decision Diffie–Hellman assumption in $\mathbb{G}$ (which we denote $\mathrm{DDH}_{\mathbb{G}}$) or in $\mathrm{DDH}_{\tilde{\mathbb{G}}}$ is sometimes referred to as the *external Diffie–Hellman* (XDH) assumption. The assumption that DDH holds in both $\mathbb{G}$ and $\tilde{\mathbb{G}}$ is sometimes referred to as the *symmetric external Diffie–Hellman* (SXDH) assumption. The DDH assumption cannot hold in symmetric bilinear groups, in which the decision linear assumption may be made instead.

Assumption 2. (*Decision Linear Assumption* (DLIN)) The decision linear assumption [23] holds in $\mathbb{G}$ relative to $\mathcal{G}$ if for all probabilistic polynomial-time $\mathcal{A}$

$$\left|\Pr\left[\begin{array}{l}\Lambda \leftarrow \mathcal{G}(1^\lambda);\, a, b, r, s \leftarrow \mathbb{Z}_p :\\ 1 \leftarrow \mathcal{A}(\Lambda, G^a, G^b, G^r, G^s, G^{ar+bs})\end{array}\right] - \Pr\left[\begin{array}{l}\Lambda \leftarrow \mathcal{G}(1^\lambda);\, a, b, r, s, t \leftarrow \mathbb{Z}_p :\\ 1 \leftarrow \mathcal{A}(\Lambda, G^a, G^b, G^r, G^s, G^t)\end{array}\right]\right|$$

is negligible.

The 2-out-of-3 CDH assumption [70] states that given a tuple of group elements (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 [74], 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 $G \in \mathbb{G}$ and $H = \tilde{G} \in \tilde{\mathbb{G}}$ (whereas in symmetric groups, we let H be an additional generator of $\mathbb{G}$ playing the role of $\tilde{G}$). The asymmetric weak flexible CDH is formalized as follows:

Assumption 3. (*Asymmetric Weak Flexible CDH Assumption* (AWF-CDH)) We say that the asymmetric weak flexible computational Diffie–Hellman assumption holds relative to $\mathcal{G}$ if for all probabilistic polynomial-time adversaries $\mathcal{A}$

$$\Pr\left[\begin{array}{l} \Lambda \leftarrow \mathcal{G}(1^\lambda); \\ A \leftarrow \mathbb{G}^*; \\ (R, M, S, N) \leftarrow \mathcal{A}(\Lambda, A) \end{array} : \begin{array}{l} (R, M, S, N) \in (\mathbb{G}^*)^2 \times (\tilde{\mathbb{G}}^*)^2 \\ e(A, S) = e(M, \tilde{G}) \\ e(M, \tilde{G}) = e(G, N) \\ e(R, \tilde{G}) = e(G, S) \end{array}\right] \tag{1}$$

is negligible.

Lemma 1. *If* $\mathrm{DDH}_{\mathbb{G}}$ *holds relative to* $\mathcal{G}$*, then AWF-CDH holds relative to* $\mathcal{G}$*.*

Proof. Let (Λ, G^a, G^b, G^c) be an instance of $\mathrm{DDH}_{\mathbb{G}}$. We have to decide whether $c = ab$. On input (Λ, G^a), a successful AWF-CDH adversary outputs $(G^r, G^{ra}, \tilde{G}^r, \tilde{G}^{ra})$. We can thus check whether $e(G^c, \tilde{G}^r) = e(G^{ab}, \tilde{G}^r) = e(G^b, \tilde{G}^{ra})$. □

Assumption 4. (*Double Pairing Assumption* (DBP)) We say the double pairing assumption holds relative to $\mathcal{G}_{\mathsf{asym}}$ if for any probabilistic polynomial-time algorithm $\mathcal{A}$

$$\Pr\big[\Lambda \leftarrow \mathcal{G}_{\mathsf{asym}}(1^\lambda);\, G_z \leftarrow \mathbb{G}^*;\, (\tilde{Z}, \tilde{R}) \leftarrow \mathcal{A}(\Lambda, G_z) : \\ \tilde{Z}, \tilde{R} \in \tilde{\mathbb{G}}^* \text{ and } 1 = e(G_z, \tilde{Z})\, e(G, \tilde{R})\big]$$

is negligible.

Lemma 2. *If* $\mathrm{DDH}_{\mathbb{G}}$ *holds relative to* $\mathcal{G}_{\mathsf{asym}}$*, then so does* $\mathrm{DBP}_{\mathbb{G}}$ *relative to* $\mathcal{G}_{\mathsf{asym}}$*.*

Proof. Suppose that there is an adversary $\mathcal{A}$ that breaks the DBP assumption. Namely, $\mathcal{A}$ finds a pair $(\tilde{Z}, \tilde{R}) \neq (1, 1)$ satisfying the equation $e(G_z, \tilde{Z})\, e(G, \tilde{R}) = 1$ for randomly chosen G_z with more than negligible probability. We will construct an adversary $\mathcal{B}$ which breaks $\mathrm{DDH}_{\mathbb{G}}$ by using $\mathcal{A}$ as a black-box.

Given a $\mathrm{DDH}_{\mathbb{G}}$ challenge tuple $(\Lambda, A, B, C) = (\Lambda, G^a, G^b, G^c)$, algorithm $\mathcal{B}$ gives $\mathcal{A}$ an input (Λ, A). If $\mathcal{A}$ outputs $(\tilde{Z}, \tilde{R}) \neq (1, 1)$ satisfying $e(G, \tilde{Z})\, e(A, \tilde{R}) = 1$, it is true that $\tilde{Z} = \tilde{R}^{-a}$. $\mathcal{B}$ outputs 1 if $e(B, \tilde{Z})\, e(C, \tilde{R}) = 1$ and outputs 0, otherwise. This strategy is correct since $e(B, \tilde{Z})\, e(C, \tilde{R}) = e(B, \tilde{R}^{-a})\, e(C, \tilde{R}) = e(G, \tilde{R})^{c-ab}$ equals to 1 if and only if $c = ab \mod p$. Thus, $\mathcal{B}$ breaks $\mathrm{DDH}_{\mathbb{G}}$ if $\mathcal{A}$ breaks $\mathrm{DBP}_{\mathbb{G}}$. □

We could consider a dual assumption of DBP by swapping $\mathbb{G}$ and $\tilde{\mathbb{G}}$. When we need to discriminate these assumptions, we call this assumption DBP in $\mathbb{G}$ (as it is implied by DDH in $\mathbb{G}$) and the dual assumption DBP in $\tilde{\mathbb{G}}$, and denote them as $\mathsf{DBP}_{\mathbb{G}}$ and $\mathsf{DBP}_{\tilde{\mathbb{G}}}$, respectively. Analogously to Lemma 2, $\mathsf{DBP}_{\tilde{\mathbb{G}}}$ holds if $\mathsf{DDH}_{\tilde{\mathbb{G}}}$ holds. Therefore, if DDH holds in both $\mathbb{G}$ and $\tilde{\mathbb{G}}$, then DBP holds in both $\mathbb{G}$ and $\tilde{\mathbb{G}}$.

Corollary 1. *If SXDH holds relative to $\mathcal{G}$, then DBP holds in both $\mathbb{G}$ and $\tilde{\mathbb{G}}$ relative to $\mathcal{G}$.*

DBP does not hold in symmetric bilinear groups and we will therefore also rely on the Simultaneous Double Pairing Assumption (SDP), which is plausible in both symmetric and asymmetric bilinear groups.

Assumption 5. (*Simultaneous Double Pairing Assumption* (SDP)) The simultaneous double pairing assumption SDP holds in $\mathbb{G}$ relative to $\mathcal{G}$ if for all probabilistic polynomial-time adversaries $\mathcal{A}$

$$\Pr\left[\begin{array}{ll} \Lambda \leftarrow \mathcal{G}(1^\lambda); & (\tilde{Z}, \tilde{R}, \tilde{U}) \in (\tilde{\mathbb{G}}^*)^3 \\ G_z, H_z, H_u \leftarrow \mathbb{G}^*; & :\ 1 = e(G_z, \tilde{Z})\, e(G, \tilde{R}) \\ (\tilde{Z}, \tilde{R}, \tilde{U}) \leftarrow \mathcal{A}(\Lambda, G_z, H_z, H_u) & \ \ 1 = e(H_z, \tilde{Z})\, e(H_u, \tilde{U}) \end{array}\right]$$

is negligible.

The following lemma is proved in [38].

Lemma 3. *If the DLIN assumption holds relative to a symmetric bilinear groups generator $\mathcal{G}_{\mathsf{sym}}$, then the SDP assumption holds relative to $\mathcal{G}_{\mathsf{sym}}$.*

Assumption 6. (*External Diffie–Hellman Inversion Assumption* (XDHI)) The XDHI assumption holds relative to $\mathcal{G}$ if for all probabilistic polynomial-time adversaries $\mathcal{A}$

$$\Pr\left[\Lambda \leftarrow \mathcal{G}(1^\lambda);\ a \leftarrow \mathbb{Z}_p^*;\ \tilde{H} \leftarrow \tilde{\mathbb{G}}^*;\ A \leftarrow \mathcal{A}(\Lambda, \tilde{H}, \tilde{H}^a)\ :\ A = G^{1/a}\right]$$

is negligible.

Assumption 7. (*Co-Computational Diffie–Hellman Assumption* (co-CDH)) The co-CDH assumption [25] holds relative to $\mathcal{G}$ if for all probabilistic polynomial-time adversaries $\mathcal{A}$

$$\Pr\left[\Lambda \leftarrow \mathcal{G}(1^\lambda);\ a, b \leftarrow \mathbb{Z}_p^*;\ A \leftarrow \mathcal{A}(\Lambda, G^a, \tilde{G}^b)\ :\ A = G^{ab}\right]$$

is negligible.

Depending on the type of the bilinear group Λ, the XDHI assumption is implied by standard assumptions, such as the computational Diffie–Hellman assumption (CDH), the co-Diffie–Hellman assumption (co-CDH) and the decisional Diffie–Hellman assumption in $\tilde{\mathbb{G}}$ ($\mathrm{DDH}_{\tilde{\mathbb{G}}}$), as follows. Note that, CDH is implied by DLIN in Λ_{sym} and $\mathrm{DDH}_{\tilde{\mathbb{G}}}$ is implied by SXDH in Λ_{sxdh}.

Lemma 4. *CDH $\Rightarrow$ XDHI for $\mathcal{G}_{sym}$. co-CDH $\Rightarrow$ XDHI for $\mathcal{G}_{asym}$. $DDH_{\tilde{\mathbb{G}}}$ $\Rightarrow$ XDHI for $\mathcal{G}_{asym}$.*

Proof. Let $\mathcal{A}$ be an XDHI adversary with respect to $\mathcal{G}_{\mathsf{sym}}$ and let $\Lambda_{\mathsf{sym}} = (p, \mathbb{G}, \mathbb{G}_T, G, e) \leftarrow \mathcal{G}_{\mathsf{sym}}(1^\lambda)$. Note that G is a uniformly chosen generator. Given a CDH instance $(\Lambda_{\mathsf{sym}}, G^\alpha, G^\beta)$, set $\Lambda'_{\mathsf{sym}} = (p, \mathbb{G}, \mathbb{G}_T, G^\alpha, e)$ and run $\mathcal{A}$ on the XDHI instance $(\Lambda'_{\mathsf{sym}}, G^\beta, G)$. It outputs $G^{\alpha\beta}$, which is the solution to the CDH instance.

For the second implication, given an co-CDH instance $(\Lambda_{\mathsf{xdh}}, G^\alpha, \tilde{G}^\beta) \in \mathbb{G}^* \times \tilde{\mathbb{G}}^*$, choose $\tilde{G}' \leftarrow \tilde{\mathbb{G}}$, set $\Lambda'_{\mathsf{xdh}} = (p, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, G^\alpha, \tilde{G}', e)$ and run $\mathcal{A}$ on input $(\Lambda'_{\mathsf{xdh}}, \tilde{G}^\beta, \tilde{G})$. It outputs $G^{\alpha\beta}$, which is the solution to the co-CDH instance.

For the third implication, given an instance $(\Lambda_{\mathsf{sxdh}}, \tilde{G}^\alpha, \tilde{G}^\beta, \tilde{G}^\gamma)$ of $\mathrm{DDH}_{\tilde{\mathbb{G}}}$, choose $\tilde{G}' \leftarrow \tilde{\mathbb{G}}$, set $\Lambda'_{\mathsf{sxdh}} = (p, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, G, \tilde{G}', e)$ and run $\mathcal{A}$ on input $(\Lambda_{\mathsf{sxdh}}, \tilde{G}^\alpha, \tilde{G})$. If successful, $\mathcal{A}$ outputs G^α. Then, $\gamma = \alpha\beta$ can be tested by checking if $e(G^\alpha, \tilde{G}^\beta) = e(G, \tilde{G}^\gamma)$ holds or not. □

We wish to extend SDP in such a way that even if we are given some solutions, it should be hard to find another solution. Observe that given an answer to an instance of SDP, one can easily get more answers by exploiting the linearity of the relation to be satisfied. We eliminate this linearity by multiplying random pairings to both sides of the SDP equations. We call the added random pairing a *flexible* pairing since; on the one hand, it provides non-malleability in that solutions cannot be merged, and on the other hand, it can be easily randomized or combined with other solutions if their secret exponents are known.

Assumption 8. (*Simultaneous Flexible Pairing Assumption* (q-SFP)) Let $\mathcal{G}_{\mathrm{SFP}}$ denote an algorithm that takes a group description Λ as input and generates parameters $P_{\mathrm{SFP}} := (G_z, H_z, H_u, A, \tilde{A}, B, \tilde{B})$ where G_z, H_z, H_u are random generators of $\mathbb{G}$, and $(A, \tilde{A})$, $(B, \tilde{B})$ are random elements in $\mathbb{G} \times \tilde{\mathbb{G}}$. For Λ and P_{SFP}, let $\mathcal{I}_{\mathrm{SFP}}$ denote the set of tuples $I_j = (\tilde{Z}_j, \tilde{R}_j, \tilde{U}_j, S_{ij}, \tilde{T}_{ij}, V_{ij}, \tilde{W}_{ij}) \in \tilde{\mathbb{G}}^* \times \tilde{\mathbb{G}} \times \tilde{\mathbb{G}} \times \mathbb{G} \times \tilde{\mathbb{G}} \times \mathbb{G} \times \tilde{\mathbb{G}}$ that satisfy

$$e(A, \tilde{A}) = e(G_z, \tilde{Z}_j)\; e(G, \tilde{R}_j)\; e(S_{ij}, \tilde{T}_{ij}) \quad \text{and} \tag{2}$$

$$e(B, \tilde{B}) = e(H_z, \tilde{Z}_j)\; e(H_u, \tilde{U}_j)\; e(V_{ij}, \tilde{W}_{ij}). \tag{3}$$

For $I_1, \ldots, I_q$, let $Z(I_1, \ldots, I_q)$ denote collection of $\tilde{Z}_j$ in I_j. We say the q-simultaneous flexible pairing assumption holds relative to $\mathcal{G}$ if for any probabilistic polynomial-time adversary $\mathcal{A}$

$$\Pr\left[\begin{array}{l} \Lambda \leftarrow \mathcal{G}(1^\lambda);\ P_{\mathrm{SFP}} \leftarrow \mathcal{G}_{\mathrm{SFP}}(\Lambda); \\ I_1, \ldots, I_q \leftarrow \mathcal{I}_{\mathrm{SFP}}; \\ I^\star \leftarrow \mathcal{A}(\Lambda, P_{\mathrm{SFP}}, I_1, \ldots, I_q) \end{array} : \begin{array}{l} I^\star \in \mathcal{I}_{\mathrm{SFP}} \\ Z(I^\star) \notin Z(I_1, \ldots, I_q) \end{array}\right]$$

is negligible.

Note that the definition deliberately makes the restriction $\tilde{Z} \in \mathbb{G}^*$, since $\tilde{Z} = 1$ would make the problem easily solvable.

To show that the q-SFP assumption is plausible, we will now prove that it holds the generic group model where the adversary only uses the generic bilinear-group operations.

Theorem 1. *For any generic algorithm $\mathcal{A}$, the probability that $\mathcal{A}$ breaks the q-SFP assumption with ℓ group operations and pairings is bounded by $\mathcal{O}(q^2 + \ell^2)/p$.*

Before proving the theorem, we discuss implications of the above bound. In the real computation, the group operation over two elements corresponds to addition of their indices and the pairing operation corresponds to their multiplication. In the generic group model, these indices are simulated by addition and multiplication over variables and formulas. Among the elements initially given to algorithm $\mathcal{A}$ there are independent random group elements whose indices are unknown. In the simulation, these unknown indices are treated as independent variables. A group element related to those elements is indexed by a formula that describes the relation. Executions of group operations and pairings are simulated by adding or multiplying the formulas associated with the elements given as inputs to the operations. Since different formulas are supposed to represent different group elements, simulation becomes inconsistent to the real computation if any two indices represented by different formulas evaluate to the same value when concrete random values are assigned to the variables. The probability of inconsistent simulation is therefore an upper bound to any generic algorithm.

If all the formulas are polynomials, the index of a new group element yielded by a generic operation is a polynomial in the variables. For polynomials of total degree less than d, the bound after ℓ group operations with k initial group elements is given as $\mathcal{O}(d \cdot (\ell^2 + k^2))/p$ by Schwartz's lemma [87,88]. Consider the case of DL where the initially given group elements are G and G^x. The above argument tells that the index formula is a polynomial of degree 1 and the bound is $\mathcal{O}(\ell^2)/p$. In the case of SDH, the initial elements are $G, G^x, G^{x^2}, \ldots, G^{x^q}$. Thus, the index formula will be a polynomial of degree q, and the bound is $\mathcal{O}(q \cdot (\ell^2 + q^2))/p$. The loss factor of q can be as huge as the number of signature issuing, and hence, the security of SDH is far from DL. On the other hand, as we show in the proof, the indices of the initial input to SFP can be represented by Laurent polynomials of form $\frac{y}{x}$ with a common variable x of a small constant degree in the denominator. The formula in the numerator varies for elements, but they remain of degree 1. Accordingly, the index of a new group element is a Laurent polynomial of the same form. By offsetting the common denominator x and applying Schwartz's lemma to the resulting regular polynomial of degree 2, we have the bound of $\mathcal{O}(\ell^2 + q^2)/p$, which is close to the optimal bound in DL.

Proof. Let us without loss of generality assume we are in the symmetric setting where $\mathbb{G} = \tilde{\mathbb{G}}$ and $G = \tilde{G}$. The symmetric setting gives the adversary more freedom, since it is not restricted to treat elements in $\mathbb{G}$ and elements in $\tilde{\mathbb{G}}$ separately, so proving the lemma for the symmetric case automatically yields a proof in the asymmetric case too.

Using the generic group operations, the adversary can compute $\tilde{Z}, \tilde{R}, S, \tilde{T}, \tilde{U}, V, \tilde{W}$ as linear combinations of the $8 + 7q$ elements G, G_z, H_z, H_u,A, $\tilde{A}$, B, $\tilde{B}$, $\{\tilde{Z}_j, \tilde{R}_j, S_{ij}, \tilde{T}_{ij}, \tilde{U}_j, V_{ij}, \tilde{W}_{ij}\}_{j=1}^{q}$. Taking discrete logarithms of all the group elements with respect to base G, this means the adversary can compute z, r, s, t, u, v, w as linear combination of $1, g_z, h_z, h_u, a, \tilde{a}, b, \tilde{b}, \{z_j, r_j, s_{ij}, t_{ij}, u_j, v_{ij}, w_{ij}\}_{j=1}^{q}$, where

$$r_j = a \cdot \tilde{a} - g_z \cdot z_j - s_j \cdot t_j, \quad \text{and} \quad u_j = \frac{b \cdot \tilde{b} - h_z \cdot z_j - v_j \cdot w_j}{h_u}.$$

We will first consider z, r, s, t, u, v, w as formal Laurent polynomials in $\mathbb{Z}_p[g_z, h_z, h_u, a, \tilde{a}, b, \tilde{b}, z_1, \dots, w_{iq}]$. This means the adversary picks known coefficients $\zeta_* \in \mathbb{Z}_p$ and computes

$$z = \zeta_{g_z} g_z + \zeta_{h_z} h_z + \zeta_{g_r} + \zeta_{h_u} h_u + \zeta_a a + \zeta_{\tilde{a}} \tilde{a} + \zeta_b b + \zeta_{\tilde{b}} \tilde{b} + \sum_{j=1}^{q} \zeta_{z_j} z_j + \sum_{j=1}^{q} \zeta_{s_{ij}} s_{ij} + \sum_{j=1}^{q} \zeta_{t_{ij}} t_{ij}$$
$$+ \sum_{j=1}^{q} \zeta_{v_{ij}} v_{ij} + \sum_{j=1}^{q} \zeta_{w_{ij}} w_{ij} + \sum_{j=1}^{q} \zeta_{r_j} (a \cdot \tilde{a} - g_z \cdot z_j - s_j \cdot t_j) + \sum_{j=1}^{q} \zeta_{u_j} \frac{b \cdot \tilde{b} - h_z \cdot z_j - v_j \cdot w_j}{h_u}.$$

and constructs the Laurent polynomials r, s, t, u, v, w in a similar way with coefficients labeled $\rho_*, \sigma_*, \tau_*, \mu_*$, etc., respectively.

Suppose the adversary's Laurent polynomials satisfy the two verification equations

$$a \cdot \tilde{a} = g_z \cdot z + r + s \cdot t \tag{4}$$
$$b \cdot \tilde{b} = h_z \cdot z + h_u \cdot u + v \cdot w. \tag{5}$$

Let EQ3 be the equations obtained by substituting z, r, s, t in (4) with corresponding Laurent polynomials. We will first show that EQ3 implies $\sigma_{r_j} = 0$ for all j. Suppose for contradiction that $\sigma_{r_i} \neq 0$ for some i. The term $s_i^2 t_i^2$ only appears in the product $s \cdot t$, so its coefficient $\sigma_{r_i} \tau_{r_i}$ must be 0. This means $\tau_{r_i} = 0$. Looking at other terms, we see that most of the coefficients in t are 0, the term $s_i t_i a$ for instance has coefficient $\sigma_{r_i} \tau_a$ and gives us $\tau_a = 0$, and we get that $t = \tau_{g_z} g_z + \tau_{g_r}$. The coefficients of the $s_j t_j$ terms now give us $\rho_{r_j} + \sigma_{r_j} \tau_{g_r} = 0$ for all j. Putting all this together, we see the right-hand side of EQ3 has coefficient 0 for the $a\tilde{a}$ term. But the left hand side of EQ3 is $a\tilde{a}$ yielding a contradiction. We conclude $\sigma_{r_j} = 0$ for all j and by symmetry of s and t, we also get that $\tau_{r_j} = 0$ for all j.

If $\rho_{r_i} \neq 0$ for some i, we get without loss of generality from EQ3 that $\sigma_{s_i} \neq 0$ and $\tau_{t_i} \neq 0$. The coefficients of s_i^2 and t_i^2 in $s \cdot t$ have to be 0 since these terms do not appear in $g_z \cdot z$ or r, so we get $\sigma_{t_i} = 0$ and $\tau_{s_i} = 0$. Looking at coefficients for terms involving s_i in EQ3, we see that most of them must have 0 coefficients. We therefore get $t = \tau_{g_z} g_z + \tau_{g_r} + \tau_{t_i} t_i$. Symmetrically, we get $s = \sigma_{g_z} g_z + \sigma_{g_r} + \sigma_{s_i} s_i$. The terms of $s_j t_j$ for $j \neq i$ in EQ3 now give us $\rho_{r_j} = 0$ for $j \neq i$. Since the left-hand side of EQ3 has the term $a\tilde{a}$, we see $\rho_{r_i} = 1$. The $g_z z_i$ terms now shows us that $\zeta_{z_i} = 1$. Additional inspection of the different terms gives us $z = \zeta_{g_z} g_z + \zeta_{g_r} + z_i + \zeta_{s_i} s_i + \zeta_{t_i} t_i$.

We now consider the other possibility that ρ_{r_j} is 0 for all j. The term $a\tilde{a}$ gives us without loss of generality that $\sigma_a \neq 0$ and $\tau_{\tilde{a}} \neq 0$. The coefficients of a^2 and $\tilde{a}^2$ show us $\tau_a = 0$ and $\sigma_{\tilde{a}} = 0$. Inspecting the other terms, we get $t = \tau_{g_z} g_z + \tau_{g_r} + \tau_{\tilde{a}} \tilde{a}$ and $s = \sigma_{g_z} g_z + \sigma_{g_r} + \sigma_a a$. It then follows by looking at different terms that $z = \zeta_{g_z} g_z + \zeta_{g_r} + \zeta_a a + \zeta_{\tilde{a}} \tilde{a}$.

We have now deduced from (4) that $z = \zeta_{g_z} g_z + \zeta_{g_r} + \zeta_a a + \zeta_{\tilde{a}} \tilde{a}$ or $z = \zeta_{g_z} g_z + \zeta_{g_r} + z_i + \zeta_{s_i} s_i + \zeta_{t_i} t_i$ for some i. By symmetry, we get from (5) that $z = \zeta_{h_z} h_z + \zeta_{h_u} h_u + \zeta_b b + \zeta_{\tilde{b}} \tilde{b}$ or $z = \zeta_{h_z} h_z + \zeta_{h_u} h_u + z_i + \zeta_{v_i} v_i + \zeta_{w_i} w_i$ for some i. The equations can be consistent only if $z \in \{0, z_1, \ldots, z_q\}$. Neither of those choices of z would constitute a successful breach of the assumption, so we conclude that there are no formal Laurent polynomials the adversary can use to violate the assumption.

When instantiating the bilinear groups, we pick $g_z, h_z, h_u, z_j \leftarrow \mathbb{Z}_p^*$ and $a, \tilde{a}, b, \tilde{b}, s_j, t_j, v_j, w_j \leftarrow \mathbb{Z}_p$. In a typical run, the adversary would expect group elements corresponding to different Laurent polynomials to be different but there is some probability that this fails; when it fails, the adversary may be able to exploit it to break the assumption. The generic adversary's probability of success can therefore be bounded by the chance that two different Laurent polynomials collide on random inputs in $\mathbb{Z}_p$.

Let F and F' be two different Laurent polynomials associated with elements in $\mathbb{G}$ computed by the adversary. By multiplying F and F' by h_u, we get two different degree 3 polynomials. The probability of having a collision is therefore bounded by $\frac{3}{p-1}$ according to Schwartz's lemma [87]. Having initially $8 + 7q$ elements, we get after ℓ_1 group operations in $\mathbb{G}$ an upper bound of

$$\binom{8+7q+\ell_1}{2} \cdot \frac{3}{p-1} \leq \frac{\mathcal{O}((q+\ell_1)^2)}{p} \tag{6}$$

for Laurent polynomials for elements in $\mathbb{G}$ evaluating to the same value.

Using the pairing operation, the adversary gets Laurent polynomials for elements in $\mathbb{G}_T$ corresponding to products of Laurent polynomials for elements in $\mathbb{G}$. Multiplying by h_u^2, we get polynomials of degree at most 6. The risk of having a collision after ℓ_T pairing operations and group operations in $\mathbb{G}_T$ is bounded by $\binom{\ell_T}{2} \cdot \frac{6}{p-1} = \frac{\mathcal{O}(\ell_T^2)}{p}$. By setting $\ell = \ell_1 + \ell_T$, we simplify the sum of the upper bounds to $\frac{\mathcal{O}(q^2+\ell^2)}{p}$ as stated in Theorem 1. □

Given an answer $(\tilde{Z}, \tilde{R}, \tilde{U})$ to the SDP problem (Assumption 5) then setting $(S, \tilde{T}, V, \tilde{W}) := (A, \tilde{A}, B, \tilde{B})$ results in a correct solution $(\tilde{Z}, \tilde{R}, \tilde{U}, S, \tilde{T}, V, \tilde{W})$ to the SFP problem. We thus obtain the following:

Lemma 5. *If the q-SFP assumption (for arbitrary q) holds relative to $\mathcal{G}$, then the SDP assumption holds relative to $\mathcal{G}$.*

Proof. Suppose that there exists an algorithm $\mathcal{A}$ that successfully finds a valid answer $(\tilde{Z}, \tilde{R}, \tilde{U})$ to SDP. We construct an algorithm that breaks SFP as follows. Given an SFP instance $(\Lambda, G_z, H_z, H_u, A, \tilde{A}, B, \tilde{B}, I_1, \ldots, I_q)$, input (Λ, G_z, H_z, H_u) to $\mathcal{A}$. If $\mathcal{A}$ outputs $(\tilde{Z}^\star, \tilde{R}^\star, \tilde{U}^\star)$ breaking the SDP instance, set $(S^\star, \tilde{T}^\star, V^\star, \tilde{W}^\star) := (A, \tilde{A}, B, \tilde{B})$ and output $I^\star = (\tilde{Z}^\star, \tilde{R}^\star, \tilde{U}^\star, S^\star, \tilde{T}^\star, V^\star, \tilde{W}^\star)$.

Multiplying $1 = e(G_z, \tilde{Z}^\star), e(G_r, \tilde{R}^\star)$ to both sides of $e(A, \tilde{A}) = e(S^\star, \tilde{T}^\star)$ results in Eq. (2). Similarly, multiplying $1 = e(H_z, \tilde{Z}^\star)\, e(H_u, \tilde{U}^\star)$ to both sides of $e(B, \tilde{B}) = e(V^\star, \tilde{W}^\star)$ results in Eq. (3). Thus, $I^\star$ satisfies the SFP equations. Since $(\tilde{Z}^\star, \tilde{R}^\star, \tilde{U}^\star)$ is a valid answer to SDP, $\tilde{Z}^\star \neq 1$ holds. Since every $\tilde{Z}_j$ in I_j is uniformly chosen and independent of $(G_z, H_z, H_u, A, \tilde{A}, B, \tilde{B})$, it is independent of the view of the adversary.

Thus, $\tilde{Z}^\star = \tilde{Z}_j$ happens only with negligible probability for every $j \in \{1, \dots, q\}$. Thus, $I^\star$ is a correct and valid answer to the q-SFP instance. □

Our last assumption is a variant of the q-strong Diffie–Hellman (SDH) assumption [22]. In [48], it is shown that SDH implies hardness of the following two problems in bilinear groups:

1. Given G, G^x and $q-1$ random 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 and, for random c_i, v_i: $\big((K \cdot G^{v_i})^{\frac{1}{x+c_i}}, c_i, v_i\big)_{i=1}^{q-1}$, output a new $((K \cdot G^v)^{\frac{1}{x+c}}, c, v)$.

Boyen and Waters [29] 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 generator H; the scalar c_i is thus "hidden." Analogously, we define a variant of the second problem by "hiding" the scalars c_i and v_i, stating that given $F, G, H, K, G^x, H^x, \big((K \cdot G^{v_i})^{\frac{1}{x+c_i}}, F^{c_i}, H^{c_i}, G^{v_i}, H^{v_i}\big)_{i=1}^{q-1}$, it is hard to output a tuple $(A = (K \cdot G^v)^{\frac{1}{x+c}}, B = F^c, D = H^c, V = G^v, W = H^v)$ with $(c, v) \neq (c_i, v_i)$ for all i. Due to the pairing, a tuple can still be effectively verified. The assumption holds in the generic group model for both asymmetric and symmetric groups, and we state it for the former.

Assumption 9. (*Asymmetric Double Hidden Strong DH Assumption* (q-ADH-SDH)) Let $\mathcal{G}_{\text{ADH-SDH}}$ denote an algorithm that takes a group description Λ as input and generates parameters $P_{\text{ADH-SDH}} := (F, K, X, \tilde{Y}) \in \mathbb{G}^3 \times \tilde{\mathbb{G}}$ where F, K and $X = G^x$ are random generators of $\mathbb{G}$, and $\tilde{Y} = \tilde{G}^x$.

For Λ and $P_{\text{ADH-SDH}}$, let $\mathcal{I}_{\text{ADH-SDH}}$ denote the set of tuples $I_i = (A_i, B_i, \tilde{D}_i, V_i, \tilde{W}_i) \in \mathbb{G}^2 \times \tilde{\mathbb{G}} \times \mathbb{G} \times \tilde{\mathbb{G}}$ that satisfy $\tilde{Y} \cdot \tilde{D}_i \neq 1$ and

$$e(A_i, \tilde{Y} \cdot \tilde{D}_i) = e(K \cdot V_i, \tilde{G}) \quad e(B_i, \tilde{G}) = e(F, \tilde{D}_i) \quad e(V_i, \tilde{G}) = e(G, \tilde{W}_i) \tag{7}$$

We say the q-asymmetric double hidden strong Diffie–Hellman assumption holds relative to $\mathcal{G}$ if for any probabilistic polynomial-time adversary $\mathcal{A}$

$$\Pr\left[\begin{array}{ll} \Lambda \leftarrow \mathcal{G}(1^\lambda);\ P_{\text{ADH-SDH}} \leftarrow \mathcal{G}_{\text{ADH-SDH}}(\Lambda); & \\ I_1, \dots, I_{q-1} \leftarrow \mathcal{I}_{\text{ADH-SDH}}; & :\ I^\star \in \mathcal{I}_{\text{ADH-SDH}} \backslash \{I_1, \dots, I_q\} \\ I^\star \leftarrow \mathcal{A}(\Lambda, P_{\text{ADH-SDH}}, I_1, \dots, I_q) & \end{array}\right]$$

is negligible.

In symmetric bilinear groups (where $\mathbb{G} = \tilde{\mathbb{G}}$), we replace $\tilde{G} \in \tilde{\mathbb{G}}$ by a random generator $H \in \mathbb{G}$. We prove generic security of the assumption in symmetric groups, which yields a stronger result, as it implies security in asymmetric groups.

Theorem 2. *The q-ADH-SDH assumption holds in generic bilinear groups when q is a polynomial.*

Proof. We prove the symmetric case, therefore every $\tilde{G}$ in the statement of Assumption 9 is replaced by a random generator H of $\mathbb{G}$; that is, we prove that given $(p, \mathbb{G}, \mathbb{G}_T, e, G)$ and $(F, K, H, X) \overset{\$}{\leftarrow} \mathbb{G}^4$ and $Y := H^{\log_G X}$, as well as $q - 1$ tuples $(A_, B_i, D_i, V_i, W_i)$ satisfying

$$e(A_i, Y \cdot D_i) = e(K \cdot V_i, H) \quad e(B_i, H) = e(F, D_i) \quad e(V_i, H) = e(G, W_i) \tag{8}$$

it is hard to generate a new tuple (A, B, D, V, W) satisfying the above.

We follow the approach of [22] in proving the generic security. Every element from $\mathbb{G}$ and $\mathbb{G}_T$ is represented by a random string, and the adversary has access to an oracle for group operations in $\mathbb{G}$ and $\mathbb{G}_T$ and pairings: Given the representation of two elements $A, A' \in \mathbb{G}$, the oracle responds with the representation of $A \cdot A' \in \mathbb{G}$, or $e(A, A')$ respectively, and analogously for $T, T' \in \mathbb{G}_T$. Internally, the simulator represents the elements as their logarithms relative to the group generator G (and $\mathbb{G}_T$ elements relative to $e(G, G)$). When answering queries, it stores the symbolic representation of the returned element as an addition or multiplication of the polynomials for the queried elements. At the end, the simulator chooses random secret values and instantiates all stored polynomials. The simulation was perfect if no nonidentical polynomials yield the same value, which (due to Schwartz's lemma [87], since the initial polynomials are of constant degrees and the adversary can only make polynomially many queries) is negligible in λ.

It remains to show that from a challenge the adversary cannot symbolically compute a new tuple satisfying (8). We represent every group element by its discrete logarithm (index) with respect to G. An ADH-SDH tuple I_i that satisfies the equations in (8) can be written as

$$\left(A_i = (K \cdot G^{v_i})^{\frac{1}{x+c_i}},\ B_i = F^{c_i},\ D_i = H^{c_i},\ V_i = G^{v_i},\ W_i = H^{v_i}\right) \tag{9}$$

for some $c_i \in \mathbb{Z}_p \backslash \{-x\}$ and $v_i \in \mathbb{Z}_p$. Let a lower-case letter denote the index of the group elements denoted by the corresponding upper-case letter. A q-ADH-SDH instance is thus represented by the following rational fractions:

$$1,\ f,\ h,\ k,\ x,\ y = xh,\ \left\{a_i = \tfrac{k+v_i}{x+c_i},\ b_i = c_i f,\ d_i = c_i h,\ v_i,\ w_i = v_i h\right\}_{i=1}^{q-1} \tag{10}$$

Let $(A^*, B^*, D^*, V^*, W^*)$ be a solution to this instance, that is, a tuple satisfying the equations in (8). Considering the logarithms of the $\mathbb{G}_T$-elements in these equations w.r.t. basis $e(G, G)$ yields

$$a^*(xh + d^*) = (k + v^*)h \qquad b^*h = d^*f \qquad v^*h = w^* \tag{11}$$

In a generic group, all the adversary can do is apply the group operation to the elements of its input. We will show that the only linear combinations $(a^*, b^*, d^*, v^*, w^*)$ of elements in (10) satisfying (11) are $(a^* = a_i = \frac{k+v_i}{x+c_i}, b^* = b_i = c_i f, d^* = d_i = c_i h, v^* = v_i, w^* = w_i = v_i h)$ for some i. A quintuple from the instance is, however, not a valid solution, meaning a generic adversary cannot break the assumption. We make the following ansatz for a^*:

$$\begin{aligned} a^* = \alpha + \alpha_f\, f + \alpha_h\, h + \alpha_k\, k + \alpha_x\, x + \alpha_y\, xh \\ + \sum \alpha_{a,i} \tfrac{k+v_i}{x+c_i} + \sum \alpha_{b,i}\, c_i f + \sum \alpha_{d,i}\, c_i h + \sum \alpha_{v,i}\, v_i + \sum \alpha_{w,i}\, v_i h. \end{aligned}$$

Analogously, we write b^*, d^*, v^* and w^*, whose coefficients we denote by β, δ, μ and ω, respectively.

By the last equation of (11) we have that for any v^* the adversary forms, it has to provide $w^* = v^*h$ as well. We can therefore limit the elements used for v^* to those of which their product with h is also given: 1, x and v_i (for all i). This yields

$$v^* = \mu + \mu_x x + \sum \mu_{v,i} v_i \qquad\qquad w^* = \mu h + \mu_x xh + \sum \mu_{v,i} v_i h$$

Similarly, plugging in the ansätze for b^* and d^* in the second equation of (11) and equating coefficients eliminates all of the coefficients except those for fh (which yields $\beta_f = \delta_h =: \gamma$) and those for $c_i fh$ (which yields $\beta_{b,i} = \delta_{d,i} =: \gamma_i$), for all i. We have thus

$$b^* = \gamma f + \sum \gamma_i\, c_i f \qquad\qquad d^* = \gamma\, h + \sum \gamma_i\, c_i h$$

We substitute a^*, d^*, v^* by their ansätze in the first equation of (11); that is, $a^*(xh + d^*) - v^*h = kh$. Since every term contains h, for convenience we omit one h per term (i.e., we symbolically "divide" the equation by h). The first equation of (11) can thus be written as

$$\begin{aligned} &\alpha x + \alpha_f\, fx + \alpha_h\, hx + \alpha_k\, kx + \alpha_x\, xx + \alpha_y\, xhx \\ &\quad + \sum \alpha_{a,i} \tfrac{(k+v_i)x}{x+c_i} + \sum \alpha_{b,i}\, c_i fx + \sum \alpha_{d,i}\, c_i hx + \sum \alpha_{v,i}\, v_i x + \sum \alpha_{w,i}\, v_i hx \\ &\quad + \alpha\gamma + \alpha_f \gamma\, f + \alpha_h \gamma\, h + \alpha_k \gamma\, k + \alpha_x \gamma\, x + \alpha_y \gamma\, xh \\ &\quad + \sum \alpha_{a,i}\gamma\, \tfrac{k+v_i}{x+c_i} + \sum \alpha_{b,i}\gamma\, c_i f + \sum \alpha_{d,i}\gamma\, c_i h + \sum \alpha_{v,i}\gamma\, v_i + \sum \alpha_{w,i}\gamma\, v_i h \\ &\quad + \sum \alpha\gamma_i\, c_i + \sum \alpha_f \gamma_i\, fc_i + \sum \alpha_h \gamma_i\, hc_i + \sum \alpha_k \gamma_i\, kc_i + \sum \alpha_x \gamma_i\, xc_i + \sum \alpha_y \gamma_i\, xhc_i \\ &\quad + \sum\sum \alpha_{a,i}\gamma_j \tfrac{(k+v_i)c_j}{x+c_i} + \sum\sum \alpha_{b,i}\gamma_j\, c_i fc_j \\ &\quad + \sum\sum \alpha_{d,i}\gamma_j\, c_i hc_j + \sum\sum \alpha_{v,i}\gamma_j\, v_i c_j + \sum\sum \alpha_{w,i}\gamma_j\, v_i hc_j \\ &\quad - \mu + \mu_x x + \sum \mu_{v,i} v_i = k, \end{aligned}$$

which after regrouping yields

$$(\alpha\gamma - \mu)\, 1 \;+\; (\alpha_f \gamma)\, f \;+\; (\alpha_h \gamma)\, h \;+\; (\alpha + \alpha_x \gamma - \mu_x)\, x + (\alpha_h + \alpha_y \gamma)\, xh \tag{12a}$$

$$+ \sum (\alpha_{a,i}\gamma)\, \frac{k + v_i}{x + c_i} \;+\; \sum (\alpha_{b,i}\gamma + \alpha_f \gamma_i)\, c_i f \;+\; \sum (\alpha_{d,i}\gamma + \alpha_h \gamma_i)\, c_i h \;+ \sum (\alpha_{w,i}\gamma)\, v_i h \tag{12b}$$

$$+ (\alpha_f)\, xf \;+\; (\alpha_k)\, xk \;+\; (\alpha_x)\, x^2 + (\alpha_y)\, x^2 h + \sum (\alpha_{d,i} + \alpha_y \gamma_i)\, c_i xh \;+\; \sum (\alpha_{b,i})\, c_i xf \tag{12c}$$

$$+ \sum (\alpha_{v,i})\, v_i x \;+\; \sum (\alpha_{w,i})\, v_i xh \;+ \sum (\alpha\gamma_i)\, c_i \;+ \sum (\alpha_k \gamma_i)\, c_i k \;+\; \sum (\alpha_x \gamma_i)\, xc_i \tag{12d}$$

$$+\sum\sum(\alpha_{b,i}\gamma_j)\,c_ic_jf\;+\;\sum\sum(\alpha_{d,i}\gamma_j)\,c_ic_jh+\;\sum\sum(\alpha_{v,i}\gamma_j)\,v_ic_j\;+\;\sum\sum(\alpha_{w,i}\gamma_j)\,v_ic_jh \tag{12e}$$

$$+\underbrace{(\alpha_k\gamma)}_{=:\lambda_k}\,k\;+\;\sum\underbrace{(\alpha_{v,i}\gamma-\mu_{v,i})}_{=:\lambda_{v,i}}\,v_i\;+\;\sum\underbrace{(\alpha_{a,i})}_{=:\lambda_{xa,i}}\,\frac{x(k+v_i)}{x+c_i}+\sum\sum\underbrace{(\alpha_{a,i}\gamma_j)}_{=:\lambda_{ca,i,j}}\,\frac{c_j(k+v_i)}{x+c_i}\;=\;k. \tag{12f}$$

To do straightforward comparison of coefficients, we would have to multiply the equation by $\prod_{i=1}^{q-1}(x+c_i)$ first. For the sake of presentation, we keep the fractions and instead introduce new equations for the cases where a linear combination leads to a fraction that cancels down.

Now, comparison of coefficients of the two sides of the above equation shows that all coefficients in lines (12a)–(12e) must be 0, whereas for the last line we have a different situation: Adding $\frac{x(k+v_i)}{x+c_i}$ and $\frac{c_i(k+v_i)}{x+c_i}$ reduces to $k+v_i$ (but this is the only combination that reduces); we have thus

$$\text{for all } i:\lambda_{xa,i}=\lambda_{ca,i,i} \qquad \text{for all } i\neq j:\lambda_{ca,i,j}=0 \tag{13}$$

$$\text{coefficient of} k:\quad \sum\lambda_{xa,i}+\lambda_k=1 \qquad \text{coefficient of } v_i:\quad \lambda_{xa,i}+\lambda_{v,i}=0 \tag{14}$$

We now solve the equations "all coefficients in Lines (12a) to (12e) equal 0," and Eqs. (13) and (14) for the values $\big(\alpha,\alpha_f,\alpha_h,\alpha_k,\alpha_x,\alpha_y,\gamma,\mu,\mu_x,\{\alpha_{a,i},\alpha_{b,i},\alpha_{d,i},\alpha_{v,i},\alpha_{w,i},\gamma_i,\mu_{v,i}\}\big)$.

The first four terms and the last term in Line (12c) and the first two terms in Line (12d) immediately yield: $\alpha_f=\alpha_k=\alpha_x=\alpha_y=\alpha_{b,i}=\alpha_{v,i}=\alpha_{w,i}=0$ for all i. Now $\alpha_y=0$ implies $\alpha_h=0$ by the last term in (12a), and moreover, $\alpha_{d,i}=0$ for all i by the fifth term in (12c). Plugging in these values, the only equations different from "$0=0$" are the following:

$$\alpha\,\gamma-\mu=0 \qquad \alpha-\mu_x=0 \tag{15}$$

$$\alpha_{a,i}\,\gamma=0\;\;(\forall i) \qquad \alpha\,\gamma_i=0\;\;(\forall i) \tag{16}$$

$$\alpha_{a,i}(1-\gamma_i)=0\;\;(\forall i) \qquad \alpha_{a,i}\,\gamma_j=0\;\;(\forall i\neq j) \tag{17}$$

$$\sum_{i=1}^{q-1}\alpha_{a,i}=1 \qquad \alpha_{a,i}-\mu_{v,i}=0\;\;(\forall i) \tag{18}$$

where the second equation in (15), denoted by "(15.2)", follows from the fourth term in (12a) and $\alpha_x=0$. (16.1) and (16.2) follow from the first term in (12b) and the third term in (12d), respectively. Equations (17) are the equations in (13); those in (18) are the ones from (14) taking into account that $\alpha_k=0$ and $\alpha_{v,i}=0$ for all i. The variables not yet proved to be 0 are α, γ, μ, μ_x, $\alpha_{a,i}$, γ_i and $\mu_{v,i}$ for $1\leq i\leq q-1$.

We first show that there exists $i^*\in\{1,\ldots,q-1\}$ such that $\alpha_{a,j}=0$ for all $j\neq i^*$: Assume there exist $i\neq j$ such that $\alpha_{a,i}\neq 0$ and $\alpha_{a,j}\neq 0$; then by (17.1), we have $\gamma_i=\gamma_j=1$, which contradicts (17.2).

This result implies the following: by (18.1), we have $\alpha_{a,i^*}=1$, and by (17.1), we have $\gamma_{i^*}=1$, whereas for all $j\neq i^*$: $\gamma_j=0$ by (17.2). We have thus shown that $\alpha_{a,i^*}=\gamma_{i^*}=1$ and $\alpha_{a,j}=\gamma_j=0$ for all $j\neq i^*$.

This now implies $\alpha = 0$ (by 16.2), and thus, $\mu = \mu_x = 0$ by [(15.1) and (15.2), respectively]. Moreover, $\gamma = 0$ (by 16.1) and for all i: $\alpha_{a,i} = \mu_{v,i}$ (by 18.2). The only nonzero variables are thus $\alpha_{a,i^*} = \gamma_{i^*} = \mu_{v,i^*} = 1$.

Plugging in our results in the ansätze for a^*, b^*, d^*, v^* and w^*, we proved that there exists $i^* \in \{1, \ldots, q-1\}$ such that $a^* = \frac{k+v_{i^*}}{x+c_{i^*}}$, $b^* = c_{i^*} f$, $d^* = c_{i^*} h$, $v^* = v_{i^*}$ and $w^* = v_{i^*} h$. This means that the only tuples $(A^*, B^*, D^*, V^*, W^*)$ satisfying (8) and being generically constructible from a ADH-SDH instance are the tuples from that instance, which concludes our proof of generic security of ADH-SDH. □

2.6. *The Groth–Sahai Proof System for Pairing-Product Equations*

The Groth–Sahai (GS) proof system [59] gives efficient non-interactive witness-indistinguishable (NIWI) proofs and non-interactive zero-knowledge (NIZK) proofs for languages that can be described as sets of satisfiable equations, each of which falls in one of the following categories: pairing-product equations, multi-exponentiation equations and general arithmetic gates. The GS proof system can be instantiated under different assumption: for instance in the asymmetric setting under the SXDH assumption, which says DDH holds in both $\mathbb{G}$ and $\tilde{\mathbb{G}}$ and in the symmetric setting under the DLIN assumption.

In GS proofs, there are two types of common reference string (CRS), which are computationally indistinguishable. One type is called "real" and gives perfect soundness and allows extraction of the group elements of a witness given a secret extraction key that is set up together with the CRS. The second type is called "simulated" and yields perfectly witness-indistinguishable proofs, which are also zero knowledge for some types of equations. When proving a statement, described as a set of equations, one first commits to the witness components and then produces elements for each equation that prove the corresponding committed values satisfy the equation.

Of the types of equations the GS proof system supports, we are mainly interested in pairing-product equations over variables $X_1, \ldots, X_m \in \mathbb{G}$ and $\tilde{Y}_1, \ldots, \tilde{Y}_n \in \tilde{\mathbb{G}}$:

$$\prod_{i=1}^{n} e(A_i, \tilde{Y}_i) \prod_{i=1}^{m} e(X_i, \tilde{B}_i) \prod_{i=1}^{m} \prod_{j=1}^{n} e(X_i, \tilde{Y}_j)^{c_{i,j}} = T,$$

where $\{A_i\}_{i=1}^{n} \in \mathbb{G}^n$, $\{\tilde{B}_i\}_{i=1}^{m} \in \tilde{\mathbb{G}}^m$, $\{c_{i,j}\}_{i=1,j=1}^{m,\ n} \in \mathbb{Z}_p$, and $T \in \mathbb{G}_T$ are public constants. When the equations involve variables only in one of the groups, we get simpler, one-sided equations $\prod_{i=1}^{n} e(A_i, \tilde{Y}_i) = T$ or $\prod_{i=1}^{m} e(X_i, \tilde{B}_i) = T$, which also yield more efficient proofs.

In the SXDH instantiation, a commitment to a group element consists of two group elements and a proof for a pairing-product equation costs 8 group elements. In the symmetric DLIN instantiation, a commitment consists of 3 group elements and a proof consists of 9 group elements. For linear pairing-product equations, the size of a proof drops to 2 and 3 group elements in the SXDH and the DLIN settings, respectively.

The GS proof system is witness indistinguishable when $T \in \mathbb{G}_T$ is an element in the target group without some particular structure. If for each equation $T = 1$, possibly after some rewriting of equations, the GS proof system becomes zero knowledge.

Note that in this context the word *proof* can denominate either “proof of satisfiability” or language membership (which thus includes the commitments) or mean a proof *that the content of some given commitments satisfies a given equation*. We adopt the latter diction and say *proof of knowledge* when we include the commitments. Please refer to [59] for further details.

2.7. *Pairing-Randomization Techniques*

We introduce techniques that randomize elements in a pairing or a pairing product without changing their value in $\mathbb{G}_T$. Not all of them are used in this paper, but they do have applications, e.g., [7].

- **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 $\mathbb{G} \times \tilde{\mathbb{G}}$ under the condition of $A = e(X', Y')$. If $A = 1$, then first flip a coin and pick $e(1, 1)$ with probability $1/(2p - 1)$. Otherwise, select $X' \neq 1$ uniformly, and output either $e(1, X')$ or $e(X', 1)$ with probability $1/2$.
- **Sequential Randomization** $\{X_i', Y_i'\}_{i=1}^k \leftarrow \mathsf{RandSeq}(\{X_i, Y_i\}_{i=1}^k)$: A pairing product $A = e(X_1, Y_1) \cdots e(X_k, Y_k)$ is randomized into $A = e(X_1', Y_1') \cdots e(X_k', Y_k')$ as follows:

  Pick $(\gamma_1, \ldots, \gamma_{k-1}) \leftarrow \mathbb{Z}_p^{k-1}$. We begin with randomizing the first pairing 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 the same way. 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

  $$A = e(X_1 X_2^{-\gamma_1}, Y_1)\, e(X_2, Y_1^{\gamma_1} Y_2)\, e(X_3, Y_3) \cdots e(X_k, Y_k). \tag{19}$$

  Next, we randomize the second pairing using the third one. As before, if $Y_1^{\gamma_1} Y_2 = 1$ or $X_3 = 1$, replace them with 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

  $$A = e(X_1 X_2^{-\gamma_1}, Y_1)\, e(X_2 X_3^{-\gamma_2}, Y_1^{\gamma_1} Y_2)\, e(X_3, (Y_1^{\gamma_1} Y_2)^{\gamma_2} Y_3) \cdots e(X_k, Y_k). \tag{20}$$

  This continues up to the $(k-1)$-th 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 the preceding $k-1$ pairings. Finally, process every pairing by the inner randomization.
- **Extension** $\{X_i', Y_i'\}_{i=1}^{k'} \leftarrow \mathsf{Extend}(\{X_i, Y_i\}_{i=1}^k)$: For $k' \geq k$, a pairing product $A = e(X_1, Y_1) \cdots e(X_k, Y_k)$ is randomized to $A = e(X_1', Y_1') \cdots e(X_{k'}', Y_{k'}')$ as follows: For $i \in [k, k']$, let $X_i = 1$ and $Y_i = 1$. Then, execute $\{X_i', Y_i'\}_{i=1}^{k'} \leftarrow \mathsf{RandSeq}(\{X_i, Y_i\}_{i=1}^{k'})$ and output $\{X_i', Y_i'\}_{i=1}^{k'}$.
- **One-side Randomization** $\{X_i'\}_{i=1}^k \leftarrow \mathsf{RandOneSide}(\{G_i, X_i\}_{i=1}^k)$: This algorithm works only in the symmetric setting Λ_{sym}. Let G_i be an element in $\mathbb{G}^*$. A pairing product $A = e(G_1, X_1) \cdots e(G_k, X_k)$ is randomized into $A =$

$e(G_1, X'_1) \cdots e(G_k, X'_k)$ as follows. Pick $(\gamma_1, \ldots, \gamma_{k-1}) \leftarrow \mathbb{Z}_p^{k-1}$ and multiply $1 = e(G_1, G_2^{\gamma_1})\, e(G_2, G_1^{-\gamma_1})$ to both sides of the formula. We thus obtain

$$A = e(G_1, X_1 G_2^{\gamma_1})\, e(G_2, X_2 G_1^{-\gamma_1})\, e(G_3, X_3) \cdots e(G_k, X_k). \tag{21}$$

Next, multiply $1 = e(G_2, G_3^{\gamma_2})\, e(G_3, G_2^{-\gamma_2})$, which yields

$$A = e(G_1, X_1 G_2^{\gamma_1})\, e(G_2, X_2 G_1^{-\gamma_1} G_3^{\gamma_2})\, e(G_3, X_3 G_2^{-\gamma_2}) \cdots e(G_k, X_k). \tag{22}$$

Continue until γ_{k-1}, so we eventually have $A = e(G_1, X'_1) \cdots e(G_k, X'_k)$. Observe that every X'_i for $i = 1, \ldots, k-1$ is distributed uniformly in $\mathbb{G}$ due to the uniform multiplicative factor $G_{i+1}^{\gamma_i}$. In the k-th pairing, X'_k follows the distribution determined by A and the preceding $k-1$ pairings. Thus, $(X'_1, \ldots, X'_k)$ is uniform over $\mathbb{G}^k$ conditioned on being evaluated to A.

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

3. Homomorphic Trapdoor Commitments

Homomorphic trapdoor commitments are used in a number of contexts, in particular as a building block in zero-knowledge proofs. An example of a frequently used scheme is that of Pedersen [83] that can be used to commit to elements from the field $\mathbb{Z}_p$.

Our goal in this section is to construct homomorphic trapdoor commitment schemes for *group elements*. We will construct both strict structure-preserving commitments where both messages, commitments and openings are elements of the source groups $\mathbb{G}$, $\tilde{\mathbb{G}}$ and relaxed structure-preserving commitments where we allow the commitments to contain elements from the target group $\mathbb{G}_T$.

In [9], it is shown that when committing with source-group elements, the size of the commitment cannot be smaller than the size of the input message. The strict structure-preserving commitments therefore grow linearly with the number of group elements in the messages. To contrast this result, we also show that in the relaxed structure-preserving setting it is possible to get constant-size commitments for messages containing many group elements.

Table 1 summarizes the performance of structure-preserving homomorphic trapdoor commitment schemes from this section and the existing ones from [9,38].

3.1. *Commitments Using Target-Group Elements*

This section includes two homomorphic trapdoor commitment schemes whose commitments consist of elements in $\mathbb{G}_T$. The first scheme, HTC1, works in both symmetric and asymmetric groups and can be seen as an optimization of the scheme we first presented in [56] with a simpler assumption. The second scheme, HTC2, only works in the asymmetric setting in exchange of gaining efficiency.

Table 1. Summary of homomorphic trapdoor commitments.

Category	Scheme	Group type	$\lvert ck\rvert$	$\lvert msg\rvert$	$\lvert com\rvert$	$\lvert open\rvert$	#(pairings)	#(PPE)	Assm.
Commit with $\mathbb{G}_T$ elements	HTC1	Any	$2k+1^{[1]}$	$k^{[2]}$	$2^{[T]}$	$2^{[2]}$	$2k+2$	2	SDP
	HTC2	Asym.	$k^{[1]}$	$k^{[2]}$	$1^{[T]}$	$1^{[2]}$	$k+1$	1	DBP
Commit with source-group elements	CLY09 [38]	Symm.	$5^{[1]}$	$1^{[1]}$	$3^{[1]}$	$3^{[1]}$	9	3	DLIN
	HTC3	Symm.	$2k+1^{[1]}$	$k^{[1]}$	$2k+2^{[1]}$	$2^{[1]}$	2k+2	2	SDP
	AHO12 [9]	Symm.	$2k+3^{[1]}$	$k^{[1]}$	$k+2^{[1]}$	$2^{[1]}$	$2k+4$	2	SDP
	AHO12 [9]	Asym.	$k+1^{[1]}, 1^{[2]}$	$k^{[2]}$	$1^{[1]}, k^{[2]}$	$1^{[2]}$	$k+2$	1	DBP
Commit to $\mathbb{Z}_p$	HTC4	Any	$1^{[2]}$	$1^{[p]}$	$1^{[2]}$	$1^{[1]}$	2	1	XDHI

By [1], [2], $[T]$, and $[p]$, we denote $\mathbb{G}$, $\tilde{\mathbb{G}}$, $\mathbb{G}_T$ and $\mathbb{Z}_p$ where the data belong to. #(pairings) and #(PPE) count the number of pairings and pairing-product equations, respectively, in verifying a correct opening

Both schemes can be used to commit to k group elements in $\tilde{\mathbb{G}}$ at once. It is inspired by the Pedersen commitment scheme, but uses pairings instead of exponentiations. The use of the pairing means that we commit to source-group elements, but the final commitment is a group element in the target group.

[Commitment Scheme HTC1]

Setup(1^λ)**:** Run $\Lambda := (p, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, e, G, \tilde{G}) \leftarrow \mathcal{G}(1^\lambda)$. Output $gk := \Lambda$.

Key(gk)**:** Choose random generator H_u from $\mathbb{G}^*$. For $i = 1, \ldots, k$, choose γ_i and δ_i from $\mathbb{Z}_p^*$ and compute $G_i := G^{\gamma_i}$ and $H_i := H_u^{\delta_i}$. Output commitment key $ck := (\Lambda, H_u, G_1, H_1, \ldots, G_k, H_k)$ and trapdoor $tk := (\gamma_1, \delta_1 \ldots, \gamma_k, \delta_k)$.

Com(ck, msg)**:** Parse msg as $(\tilde{M}_1, \ldots, \tilde{M}_k) \in \tilde{\mathbb{G}}^k$. Choose $\tilde{R}$ and $\tilde{U}$ randomly from $\tilde{\mathbb{G}}$, and compute

$$C_1 := e(G, \tilde{R}) \prod_{i=1}^k e(G_i, \tilde{M}_i) \quad \text{and} \quad C_2 := e(H_u, \tilde{U}) \prod_{i=1}^k e(H_i, \tilde{M}_i). \quad (23)$$

Output $com := (C_1, C_2)$ and $open := (\tilde{R}, \tilde{U})$.

Vrf$(ck, com, msg, open)$**:** Parse com as $(C_1, C_2) \in \mathbb{G}_T^2$, msg as $(\tilde{M}_1, \ldots, \tilde{M}_k) \in \tilde{\mathbb{G}}^k$, and $open$ as $(\tilde{R}, \tilde{U}) \in \tilde{\mathbb{G}}^2$. Output 1 if (23) holds. Output 0, otherwise.

Sim(ck)**:** Choose $\tilde{R}$ and $\tilde{U}$ randomly from $\tilde{\mathbb{G}}$ and compute $C_1 := e(G, \tilde{R})$ and $C_2 := e(H_u, \tilde{U})$. Output $com := (C_1, C_2)$ and $ek := (\tilde{R}, \tilde{U})$.

Equiv(ck, msg, ek, tk)**:** Parse msg as $(\tilde{M}_1, \ldots, \tilde{M}_k) \in \tilde{\mathbb{G}}^k$, ek as $(\tilde{R}, \tilde{U}) \in \tilde{\mathbb{G}}^2$ and tk as $(\gamma_1, \delta_1, \ldots, \gamma_k, \delta_k)$. Compute $\tilde{R}' := \tilde{R} \prod_{i=1}^k \tilde{M}_i^{-\gamma_i}$, and $\tilde{U}' := \tilde{U} \prod_{i=1}^k \tilde{M}_i^{-\delta_i}$. Output $open := (\tilde{R}', \tilde{U}')$.

¶

Theorem 3. *HTC1 is a homomorphic trapdoor commitment scheme. It is perfectly trapdoor and computationally binding if the SDP assumption holds in $\mathbb{G}$ relative to $\mathcal{G}$.*

Proof. Correctness trivially holds by construction.

To see that the scheme is homomorphic, observe that

$$\Big(e(G, \tilde{R}) \prod_{i=1}^{k} e(G_i, \tilde{M}_i)\Big)\Big(e(G, \tilde{R}') \prod_{i=1}^{k} e(G_i, \tilde{M}_i')\Big) = e(G, \tilde{R} \cdot \tilde{R}') \prod_{i=1}^{k} e(G_i, \tilde{M}_i \cdot \tilde{M}_i')$$

$$\Big(e(H_u, \tilde{U}) \prod_{i=1}^{k} e(H_i, \tilde{M}_i)\Big)\ \Big(e(H_u, \tilde{U}') \prod_{i=1}^{k} e(H_i, \tilde{M}_i')\Big) = e(H_u, \tilde{U} \cdot \tilde{U}') \textstyle\prod_{i=1}^{k} e(H_i, \tilde{M}_i \cdot \tilde{M}_i')$$

This means (C_1C_1', C_2C_2') is a valid commitment to $(\tilde{M}_1\tilde{M}_1', \cdots, \tilde{M}_k\tilde{M}_k')$ that can be opened with $(\tilde{R}\tilde{R}', \tilde{U}\tilde{U}')$.

The commitment scheme is perfectly hiding because the components $e(G_r, \tilde{R})$ and $e(H_u, \tilde{U})$ make C_1 and C_2 uniformly random regardless of the message. The scheme is also perfectly trapdoor. To see this, observe that for any $(\tilde{M}_1, \ldots, \tilde{M}_k)$ and any $(C_1, C_2, \tilde{R}, \tilde{U})$ and $(\tilde{R}', \tilde{U}')$ legitimately generated by HTC1.Sim and HTC1.Equiv, respectively, and it holds that

$$e(G, \tilde{R}') \prod_{i=1}^{k} e(G_i, \tilde{M}_i) = e(G, \tilde{R} \prod_{i=1}^{k} \tilde{M}_i^{-\gamma_i}) \prod_{i=1}^{k} e(G_i, \tilde{M}_i) = e(G, \tilde{R}) = C_1, \text{ and}$$

$$e(H_u, \tilde{U}') \prod_{i=1}^{k} e(H_i, \tilde{M}_i) = e(H_u, \tilde{U} \prod_{i=1}^{k} \tilde{M}_i^{-\delta_i}) \prod_{i=1}^{k} e(H_i, \tilde{M}_i) = e(H_u, \tilde{U}) = C_2.$$

Thus, HTC1.Vrf accepts $(\tilde{M}_1, \ldots, \tilde{M}_k)$ and $(\tilde{R}', \tilde{U}')$ as a correct opening for (C_1, C_2). Moreover, since C_1, C_2 are uniformly random and the commitments and messages uniquely determine the openings, real commitments and openings have the same probability distribution as do simulated commitments and trapdoor openings.

Finally, we will show that HTC1 is computationally binding. Suppose that there exists an adversary that successfully opens a commitment to two distinct messages. We show that one can break SDP by using such an adversary. Given an SDP challenge (Λ, H_u, G_z, H_z), do as follows.

- Set $G_i := G_z^{\chi_i} G^{\gamma_i}$ and $H_i := H_z^{\chi_i} H_u^{\delta_i}$ for $i = 1, \ldots, k$. Abort if $G_i = 1$ or $H_i = 1$ for any i. Otherwise, run the adversary on $ck = (\Lambda, H_u, G_1, H_1, \ldots, G_k, H_k)$.
- Given two openings $(\tilde{M}_1, \ldots, \tilde{M}_k, \tilde{R}, \tilde{U})$ and $(\tilde{M}_1', \ldots, \tilde{M}_k', \tilde{R}', \tilde{U}')$ from the adversary that yield the same commitment (C_1, C_2), compute

$$\tilde{Z}^\star := \prod_{i=1}^{k} \left(\frac{\tilde{M}_i}{\tilde{M}_i'}\right)^{\chi_i}, \quad \tilde{R}^\star := \frac{\tilde{R}}{\tilde{R}'} \prod_{i=1}^{k} \left(\frac{\tilde{M}_i}{\tilde{M}_i'}\right)^{\gamma_i}, \quad \tilde{U}^\star := \frac{\tilde{U}}{\tilde{U}'} \prod_{i=1}^{k} \left(\frac{\tilde{M}_i}{\tilde{M}_i'}\right)^{\delta_i}. \qquad (24)$$

- Output $(\tilde{Z}^\star, \tilde{R}^\star, \tilde{U}^\star)$.

The key generation step aborts with probability at most $2k/p$ that is negligible. Then, since the openings fulfills (23), we have

$$\begin{aligned}
1 &= e\Big(G, \tfrac{\tilde{R}}{\tilde{R}'}\Big) \prod e\Big(G_i, \tfrac{\tilde{M}_i}{\tilde{M}_i'}\Big) = e\Big(G_z, \prod_{i=1}^{k} \Big(\tfrac{\tilde{M}_i}{\tilde{M}_i'}\Big)^{\chi_i}\Big)\, e\Big(G, \tfrac{\tilde{R}}{\tilde{R}'} \prod_{i=1}^{k} \Big(\tfrac{\tilde{M}_i}{\tilde{M}_i'}\Big)^{\gamma_i}\Big) \\
&= e(G_z, \tilde{Z}^\star)\, e(G, \tilde{R}^\star), \text{ and} \\
1 &= e\Big(H_u, \tfrac{\tilde{U}}{\tilde{U}'}\Big) \prod e\Big(H_i, \tfrac{\tilde{M}_i}{\tilde{M}_i'}\Big) = e\Big(H_z, \prod_{i=1}^{k} \Big(\tfrac{\tilde{M}_i}{\tilde{M}_i'}\Big)^{\chi_i}\Big)\, e\Big(H_u, \tfrac{\tilde{U}}{\tilde{U}'} \prod_{i=1}^{k} \Big(\tfrac{\tilde{M}_i}{\tilde{M}_i'}\Big)^{\delta_i}\Big) \\
&= e(H_z, \tilde{Z}^\star)\, e(H_u, \tilde{U}^\star).
\end{aligned}$$

But $\vec{\tilde{M}} \neq \vec{\tilde{M}}'$, so there exists i such that $\tilde{M}_i/\tilde{M}_i' \neq 1$. Also, χ_i is independent from the view of the adversary. That is, for every choice of χ_i, there exist corresponding γ_i and δ_i that give the same G_i and H_i. Therefore, $\tilde{Z}^\star = \prod_i (\tilde{M}_i/\tilde{M}_i')^{\chi_i} \neq 1$ with overwhelming probability and $\tilde{R}^\star, \tilde{U}^\star \neq 1$ holds automatically. Thus, $(\tilde{Z}^\star, \tilde{R}^\star, \tilde{U}^\star)$ is a valid answer to the instance of SDP. □

In asymmetric bilinear groups, we can deploy keys and messages in different groups that cannot map each other. As a result, we can construct more efficient scheme HTC2 that works only in asymmetric bilinear groups.

[Commitment Scheme HTC2]

Setup(1^λ)**:** Run $\Lambda := (p, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, e, G, \tilde{G}) \leftarrow \mathcal{G}_{\mathsf{asym}}(1^\lambda)$. Output $gk := \Lambda$.

Key(gk)**:** For $i = 1, \ldots, k$ choose $\gamma_i \leftarrow \mathbb{Z}_p^*$ and compute $G_i := G^{\gamma_i}$. Output commitment key $ck := (\Lambda, G_1, \ldots, G_k)$ and trapdoor $tk := (\gamma_1, \ldots, \gamma_k)$.

Com(ck, msg)**:** Parse msg as $(\tilde{M}_1, \ldots, \tilde{M}_k) \in \tilde{\mathbb{G}}^k$. Choose $\tilde{R} \leftarrow \tilde{\mathbb{G}}$, and compute

$$C := e(G, \tilde{R}) \textstyle\prod_{i=1}^{k} e(G_i, \tilde{M}_i). \tag{25}$$

Output commitment $com := C$ and opening $open := \tilde{R}$.

Vrf$(ck, com, msg, open)$**:** Parse the inputs as $com = C \in \mathbb{G}_T$, $msg = (\tilde{M}_1, \ldots, \tilde{M}_k) \in \tilde{\mathbb{G}}^k$ and $open = \tilde{R} \in \mathbb{G}_T$. Output 1 if (25) holds and 0 otherwise.

Sim(ck)**:** Choose $\tilde{R} \leftarrow \tilde{\mathbb{G}}$ and compute $C := e(G, \tilde{R})$. Output $com := C$ and $ek := \tilde{R}$.

Equiv(ck, msg, ek, tk)**:** Parse msg as $(\tilde{M}_1, \ldots, \tilde{M}_k) \in \tilde{\mathbb{G}}^k$. Get $\tilde{R}$ and $(\gamma_1, \ldots, \gamma_k)$ from ek and tk, respectively. Compute $\tilde{R}' := \tilde{R} \cdot \prod_{i=1}^{k} \tilde{M}_i^{-\gamma_i}$. Output the opening $open := \tilde{R}'$.

¶

To commit to messages in $\mathbb{G}$, one can construct a "dual" scheme by interchanging the role of $\mathbb{G}$ and $\tilde{\mathbb{G}}$ in the above construction.

Theorem 4. *HTC2 is a homomorphic trapdoor commitment scheme. It is perfectly trapdoor and computationally binding if the DBP assumption holds in $\mathbb{G}$ relative to $\mathcal{G}_{asym}$.*

Proof. Correctness can be verified by inspection.

To see the commitment scheme is homomorphic, observe

$$\left(e(G,\tilde{R})\prod_{i=1}^{k}e(G_i,\tilde{M}_i)\right)\left(e(G,\tilde{R}')\prod_{i=1}^{k}e(G_i,\tilde{M}_i')\right)=e(G,\tilde{R}\cdot\tilde{R}')\prod_{i=1}^{k}e(G_i,\tilde{M}_i\cdot\tilde{M}_i').$$

The commitment scheme is perfectly hiding because the randomization, $e(G, R)$, makes the commitment a uniformly random target-group element irrespective of the message. Furthermore, it is perfectly trapdoor because a simulated commitment is also a uniformly random target-group element and conditioned on a message and commitment (25) uniquely determines the opening.

To prove the commitment scheme is computationally binding, consider an adversary $\mathcal{A}$ that has not negligible probability of opening a commitment to two different messages. We will use it to construct an adversary $\mathcal{B}$ that breaks the DBP assumption.

$\mathcal{B}$ gets as input a DBP challenge (Λ, G_z) and wants to find $\tilde{Z}^*, \tilde{R}^* \in \tilde{\mathbb{G}}^*$ such that $1 = e(G_z, \tilde{Z}^*)\,e(G, \tilde{R}^*)$. $\mathcal{B}$ sets $G_i := G_z^{\chi_i} G^{\gamma_i}$ with random $\chi_i, \gamma_i \leftarrow \mathbb{Z}_p$ for $i = 1, \ldots, k$. $\mathcal{B}$ aborts if $G_i = 1$ happens for any i, but it happens only with negligible probability. It then runs $\mathcal{A}$ on $ck = (\Lambda, G_1, \ldots, G_k)$. If $\mathcal{A}$ returns two different messages and openings $(\tilde{M}_1, \ldots, \tilde{M}_k, \tilde{R})$ and $(\tilde{M}_1', \ldots, \tilde{M}_k', \tilde{R}')$ of the same commitment C then $\mathcal{B}$ computes

$$\tilde{Z}^* := \prod_{i=1}^{k}\left(\frac{\tilde{M}_i}{\tilde{M}_i'}\right)^{\chi_i}, \quad \tilde{R}^* := \frac{\tilde{R}}{\tilde{R}'}\prod_{i=1}^{k}\left(\frac{\tilde{M}_i}{\tilde{M}_i'}\right)^{\gamma_i}. \tag{26}$$

Since $C = e(G, \tilde{R})\prod_{i=1}^{k} e(G_i, \tilde{M}_i) = e(G, \tilde{R}')\prod_{i=1}^{k} e(G_i, \tilde{M}_i')$, the output satisfies $1 = e(G_z, \tilde{Z}^*)\,e(G, \tilde{R}^*)$. Each χ_i value in $G_i = G_z^{\chi_i} G^{\gamma_i}$ is perfectly hidden by the random G^{γ_i} component, so there is only a $\frac{1}{p}$ chance of $\tilde{Z}^* = 1$. This gives $\mathcal{B}$ a solution to its DBP challenge. □

As mentioned in the introduction, the length-reducing schemes in this section are useful in reducing the size of zero-knowledge arguments. In [58], HTC2 is used to make commitments to group elements that themselves are Pedersen commitments. Pedersen commitments allow the commitment to multiple values $m_1, \ldots, m_n \in \mathbb{Z}_p$ as $\tilde{G}^t \prod_{i=1}^{n} \tilde{G}_i^{m_i}$. We can use our length-reducing commitment schemes to commit to k Pedersen commitments at once. Since our commitment schemes are homomorphic and the Pedersen commitment scheme is homomorphic, their combination is also homomorphic. We therefore get a homomorphic trapdoor commitment scheme to nk elements from $\mathbb{Z}_p$. In contrast to the Pedersen commitment scheme, however, the public key of our scheme is only $\mathcal{O}(n + k)$ group elements. This means that both commitments and keys are much smaller than the number of messages nk.

3.2. *Commitments Using Source-Group Elements*

We will now show that it is also possible to make strict structure-preserving commitment to tuples of group elements. This means the commitments themselves should also consist of source-group elements. We will do this by modifying HTC1 in the symmetric setting such that instead of publishing a commitment in $\mathbb{G}_T$, we publish the message and opening in randomized form. The randomization is done by applying the one-sided randomization RandOneSide from Sect. 2.7.

[Commitment Scheme HTC3]

Setup(1^λ)**:** Run $\Lambda := (p, \mathbb{G}, \mathbb{G}_T, e, G) \leftarrow \mathcal{G}_{\mathsf{sym}}(1^\lambda)$. Output $gk := \Lambda$.

Key(gk)**:** Choose random generators $H_u \leftarrow \mathbb{G}^*$. For $i = 1, \ldots, k$, choose $\gamma_i, \delta_i \leftarrow \mathbb{Z}_p^*$ and compute $G_i := G^{\gamma_i}$ and $H_i := H_u^{\delta_i}$. Output commitment key $ck := (\Lambda, H_u, G_1, H_1, \ldots, G_k, H_k)$ and trapdoor $tk := (\gamma_1, \delta_1 \ldots, \gamma_k, \delta_k)$.

Com(ck, msg)**:** Parse msg as $(\tilde{M}_1, \ldots, \tilde{M}_k) \in \mathbb{G}^k$. Choose $\tilde{R}, \tilde{U} \leftarrow \mathbb{G}$, and compute

$$\{C_{ai}\}_{i=0}^k \leftarrow \mathsf{RandOneSide}((G, \tilde{R}), (G_1, \tilde{M}_1), \ldots, (G_k, \tilde{M}_k)), \quad \text{and} \tag{27}$$

$$\{C_{bi}\}_{i=0}^k \leftarrow \mathsf{RandOneSide}((H_u, \tilde{U}), (H_1, \tilde{M}_1), \ldots, (H_k, \tilde{M}_k)). \tag{28}$$

Output commitment $com := (\{C_{ai}\}_{i=0}^k, \{C_{bi}\}_{i=0}^k)$ and opening $open := (\tilde{R}, \tilde{U})$.

Vrf$(ck, com, msg, open)$**:** Parse com as $(\{C_{ai}\}_{i=0}^k, \{C_{bi}\}_{i=0}^k)$, msg as $(\tilde{M}_1, \ldots, \tilde{M}_k)$, and $open$ as $(\tilde{R}, \tilde{U})$. Output 1 if the following equations hold.

$$1 = e(G, \tilde{R}/C_{a0}) \prod_{i=1}^k e(G_i, \tilde{M}_i/C_{ai}) \quad \text{and}$$

$$1 = e(H_u, \tilde{U}/C_{b0}) \prod_{i=1}^k e(H_i, \tilde{M}_i/C_{bi}). \tag{29}$$

Otherwise output 0.

Sim(ck)**:** Run $(com, open) \leftarrow \mathsf{HTC3.Com}(ck, (1, \ldots, 1))$. Output com and $ek := open$.

Equiv(ck, msg, ek, tk)**:** Parse msg as $(\tilde{M}_1, \ldots, \tilde{M}_k)$ and ek as $(\tilde{R}, \tilde{U})$. Compute $\tilde{R}' := \tilde{R} \cdot \prod_{i=1}^k \tilde{M}_i^{-\gamma_i}$, and $\tilde{U}' := \tilde{U} \cdot \prod_{i=1}^k \tilde{M}_i^{-\delta_i}$. Output the opening $dk := (\tilde{R}', \tilde{U}')$.

¶

Theorem 5. *HTC3 is a homomorphic trapdoor commitment scheme. It is perfectly trapdoor and computationally binding if the SDP assumption holds relative to $\mathcal{G}_{\mathsf{sym}}$.*

Proof. Since RandOneSide preserves the value of the pairing product, we have

$$e(G, C_{a0}) \prod_{i=1}^k e(G_i, C_{ai}) = e(G, \tilde{R}) \prod_{i=1}^k e(G_i, \tilde{M}_i) \tag{30}$$

$$e(H_u, C_{a0}) \prod_{i=1}^k e(H_i, C_{ai}) = e(H_u, \tilde{U}) \prod_{i=1}^k e(H_i, \tilde{M}_i), \tag{31}$$

which means the commitment scheme is correct.

We will now show the commitment scheme is homomorphic. Observe that

$$\begin{aligned}
&e(G, C_{a0} \cdot C'_{a0}) \prod_{i=1}^{k} e(G_i, C_{ai} \cdot C'_{ai}) \\
&= \Bigl(e(G, C_{a0}) \prod_{i=1}^{k} e(G_i, C_{ai})\Bigr)\Bigl(e(G, C'_{a0}) \prod_{i=1}^{k} e(G_i, C'_{ai})\Bigr) \\
&= \Bigl(e(G, \tilde{R}) \prod_{i=1}^{k} e(G_i, \tilde{M}_i)\Bigr)\Bigl(e(G, \tilde{R}') \prod_{i=1}^{k} e(G_i, \tilde{M}'_i)\Bigr) \\
&= e(G, \tilde{R} \cdot \tilde{R}') \prod_{i=1}^{k} e(G_i, \tilde{M}_i \cdot \tilde{M}'_i),
\end{aligned}$$

and a similar statement holds for the C_{bi} values.

This means $com := (\{C_{ai}C'_{ai}\}_{i=0}^{k}, \{C_{bi}C'_{bi}\}_{i=0}^{k})$ is a commitment to $msg := (\tilde{M}_1\tilde{M}'_1, \cdots, \tilde{M}_k\tilde{M}'_k)$ with opening $open := (\tilde{R}\tilde{R}', \tilde{U}\tilde{U}')$.

To see the commitment scheme is perfectly hiding, observe that regardless of the message, the commitment is a set of $2k + 2$ uniformly random elements in $\mathbb{G}$.

The perfect trapdoor property and the computational binding under the SDP assumption carry over directly from the proof of Theorem 3 since the messages and openings are identical to those of HTC1. □

3.3. *Committing to Messages in* $\mathbb{Z}_p$

Though not structure-preserving, schemes to commit to elements in $\mathbb{Z}_p$ are often useful and needed in applications. The following scheme, HTC4, resembles the commitment schemes in the literature studying NIZK over bilinear groups, e.g., [4,59]. We present the scheme with a formal treatment.

[Commitment Scheme HTC4]

Setup(1^λ)**:** Run $\Lambda := (p, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, e, G, \tilde{G}) \leftarrow \mathcal{G}(1^\lambda)$. Output $gk := \Lambda$.

Key(gk)**:** Select $\gamma \leftarrow \mathbb{Z}_p^*$ and set $\tilde{F} := \tilde{G}^\gamma$. Output $ck := (\Lambda, \tilde{F})$ and trapdoor $tk := \gamma$.

Com(ck, m)**:** Choose random $\delta \in \mathbb{Z}_p$ and compute commitment $C := \tilde{G}^m \tilde{F}^\delta \in \tilde{\mathbb{G}}$ and decommit key $D := G^\delta \in \mathbb{G}$. Output $com := C$ and $open := D$.

Vrf$(ck, com, msg, open)$**:** Parse the inputs accordingly. Output 1 if

$$e(G, C/\tilde{G}^m) = e(D, \tilde{F}).$$

Output 0 otherwise.

Sim(ck)**:** Choose $\delta \leftarrow \mathbb{Z}_p$ and compute $C := \tilde{F}^\delta$. Output $com := C$ and $ek := \delta$.

Equiv(ck, msg, ek, tk)**:** Take δ, γ, and m from the input, and compute $D := G^{\delta - m/\gamma}$. Output $open := D$

¶

Theorem 6. *Commitment scheme* HTC4 *is a homomorphic trapdoor commitment scheme. It is perfectly hiding and computationally binding if the XDHI assumption holds for $\mathcal{G}$.*

Proof. Correctness follows since $e(G, C/\tilde{G}^m) = e(G, \tilde{F}^\delta) = e(G^\delta, \tilde{F}) = e(D, \tilde{F})$. It is additively homomorphic since the following holds

$$\begin{aligned} e(G, C \cdot C'/\tilde{G}^{m+m'}) &= e\big(G, (\tilde{G}^m \tilde{F}^\delta) \cdot (\tilde{G}^{m'} \tilde{F}^{\delta'})/\tilde{G}^{m+m'}\big) \\ &= e(G^\delta \cdot G^{\delta'}, \tilde{F}) = e(D \cdot D', \tilde{F}). \end{aligned}$$

The perfect hiding property holds from the fact that, for any $C \in \tilde{\mathbb{G}}$, for every $m \in \mathbb{Z}_p$ there exists a single consistent $\delta \in \mathbb{Z}_p$.

The binding property is proven by showing a reduction to XDHI. Given an instance of XDHI, $(\Lambda, \tilde{H}, \tilde{H}^a)$, compose Λ' from Λ by replacing $\tilde{G}$ in Λ with $\tilde{H}$ and set $\tilde{F} = \tilde{H}^a$. Suppose that an adversary is given $ck = (\Lambda', \tilde{F})$ and outputs a commitment c correctly opened to (m, D) and (m', D') for $m \neq m'$. Then, $e(G, C/\tilde{G}^m) = e(D, \tilde{F})$ and $e(G, C/\tilde{G}^{m'}) = e(D', \tilde{F})$ hold. By dividing both sides of the equations, we have $e(G, \tilde{G}^{m-m'}) = e(D'/D, \tilde{F}) = e(D'/D, \tilde{H}^a)$. Thus by computing $(D'/D)^{1/m-m'}(= G^{1/a})$, we have a correct answer to the XDHI instance.

The equivocation is correct because

$$e(D, \tilde{F}) = e(G^{\delta - m/\gamma}, \tilde{F}) = e(G, \tilde{G}^{-m} \tilde{F}^\delta) = e(G, C/\tilde{G}^m).$$

And it is perfect because commitment C generated by Sim distributes uniformly over $\tilde{\mathbb{G}}$ as well as those by Com. Thus, the scheme has the trapdoor property. □

4. One-Time Signature Schemes

This section presents two structure-preserving one-time signature schemes. The constructions are based on the trapdoor commitment schemes in Sect. 3. Mohassel [79] presented generic conversions from chameleon hash schemes to one-time signature schemes. Their conversion can start from trapdoor commitment schemes. In this section, we present one-time signature schemes achieving better security and efficiency than the generic ones.

4.1. *Construction in General Setting*

The first scheme, OTS1, is based on the trapdoor commitment scheme HTC1 from Sect. 3.1. Compared to HTC1, it uses one extra random element to handle the one-time signing query and avoids target-group elements by representing them with pairings of random source-group elements.

The scheme is mainly designed for symmetric groups (as we show more efficient scheme in asymmetric groups), but it works also for asymmetric groups with unilateral messages belonging to $\tilde{\mathbb{G}}$.

Let $msg := (M_1, \ldots, M_k) \in \tilde{\mathbb{G}}^k$ be a message to be signed. Parameter k determines the length of a message.

[Scheme OTS1]

Setup(1^λ)**:** Run $\Lambda := (p, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, e, G, \tilde{G}) \leftarrow \mathcal{G}(1^\lambda)$. Output $gk := \Lambda$.

Key(gk)**:** Select random generator $H_u \leftarrow \mathbb{G}^*$. For $i = 0, \ldots, k$, choose γ_i, $\delta_i \leftarrow \mathbb{Z}_p^*$ and compute $G_i := G^{\gamma_i}$ and $H_i := H_u^{\delta_i}$. Let $G_z := G_0$ and $H_z := H_0$. Also choose $\rho, \varphi \leftarrow \mathbb{Z}_p^*$, and set $A := G^\rho$ and $B := H_u^\varphi$. Set $vk := (\Lambda, H_u, G_z, H_z, \{G_i, H_i\}_{i=1}^k, A, B)$ and $sk := (vk, \rho, \varphi, \{\gamma_i, \delta_i\}_{i=0}^k)$. Output (vk, sk).

Sign(sk, msg)**:** Pick $\zeta \leftarrow \mathbb{Z}_p$ and compute

$$\tilde{Z} := \tilde{G}^\zeta, \quad \tilde{R} := \tilde{G}^{\rho - \gamma_0 \zeta} \prod_{i=1}^k M_i^{-\gamma_i}, \quad \tilde{U} := \tilde{G}^{\varphi - \delta_0 \zeta} \prod_{i=1}^k M_i^{-\delta_i}.$$

Output $\sigma := (\tilde{Z}, \tilde{R}, \tilde{U})$ as a signature.

Vrf(vk, msg, σ)**:** Parse σ into $(\tilde{Z}, \tilde{R}, \tilde{U})$. Output 1 if the following equations hold; output 0 otherwise.

$$e(A, \tilde{G}) = e(G_z, \tilde{Z})\, e(G, \tilde{R}) \prod_{i=1}^k e(G_i, M_i) \tag{32}$$

$$e(B, \tilde{G}) = e(H_z, \tilde{Z})\, e(H_u, \tilde{U}) \prod_{i=1}^k e(H_i, M_i) \tag{33}$$

¶

Theorem 7. *OTS1 is a one-time signature scheme that is strongly unforgeable against one-time chosen-message attacks if SDP holds for $\mathcal{G}$.*

Proof. Syntactical consistency and correctness are verified by inspection. We focus on showing strong unforgeability. Suppose that there is a successful adversary, $\mathcal{A}$. We construct a reduction algorithm to SDP. Let $\Lambda := (p, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, e, G, \tilde{G})$ and (G_z, H_z, H_u) be an instance of SDP.

Adversary $\mathcal{A}$ is given public key vk and then a one-time signature σ for message msg of its choice. It eventually outputs a valid forgery $(\sigma^\dagger, msg^\dagger)$. To run $\mathcal{A}$ properly, the reduction algorithm simulates the public key and a one-time signature as follows.

- Simulating vk: Use given Λ for gk and (G_z, H_z, H_u) for the generators of the same name. For $i = 1, \ldots, k$, choose $\chi_i, \gamma_i, \delta_i \leftarrow \mathbb{Z}_p$ and compute $G_i := G_z^{\chi_i} G^{\gamma_i}$ and $H_i := H_z^{\chi_i} H_u^{\delta_i}$. Also choose $\zeta, \rho, \varphi \leftarrow \mathbb{Z}_p$, and set $A := G_z^\zeta G^\rho$ and $B := H_z^\zeta H_u^\varphi$. Abort if any of G_i, H_i, A and B equals 1. Send $vk := (\Lambda, H_u, G_z, H_z, \{G_i, H_i\}_{i=1}^k, A, B)$ to $\mathcal{A}$.
- Simulating σ: On receiving $msg = (M_0, \ldots, M_k)$, compute

$$\tilde{Z} := \tilde{G}^\zeta \prod_{i=1}^k M_i^{-\chi_i}, \quad \tilde{R} := \tilde{G}^\rho \prod_{i=1}^k M_i^{-\gamma_i}, \quad \tilde{U} := \tilde{G}^\varphi \prod_{i=1}^k M_i^{-\delta_i}.$$

Return $\sigma := (\tilde{Z}, \tilde{R}, \tilde{U})$ to $\mathcal{A}$.

Simulation is perfect except for the case it aborts in the key generation, which happens only with negligible probability. Given output $(\tilde{Z}^\dagger, \tilde{R}^\dagger, \tilde{U}^\dagger)$ and $(M_1^\dagger, \ldots, M_k^\dagger)$ from $\mathcal{A}$, the reduction algorithm computes

$$\tilde{Z}^\star := \left(\frac{\tilde{Z}^\dagger}{\tilde{Z}}\right) \prod_{i=1}^{k} \left(\frac{M_i^\dagger}{M_i}\right)^{\chi_i}, \quad \tilde{R}^\star := \left(\frac{\tilde{R}^\dagger}{\tilde{R}}\right) \prod_{i=1}^{k} \left(\frac{M_i^\dagger}{M_i}\right)^{\gamma_i}, \quad \tilde{U}^\star := \left(\frac{\tilde{U}^\dagger}{\tilde{U}}\right) \prod_{i=1}^{k} \left(\frac{M_i^\dagger}{M_i}\right)^{\delta_i}. \tag{34}$$

Then output $(\tilde{Z}^\star, \tilde{R}^\star, \tilde{U}^\star)$ as an answer to the instance of SDP. This completes the description of the reduction algorithm.

The simulation yields the view that distributes statistically close to the one in the real execution. The negligible error occurs when random coins are chosen so that one or more elements in vk happen to be 1. Thus, we can expect that $\mathcal{A}$ remains successful in the simulation.

Next, we show the correctness of the output of the reduction. By dividing both sides of (32) with respect to $(\tilde{Z}^\star, \tilde{R}^\star, \tilde{U}^\star)$ and $(\tilde{Z}, \tilde{R}, \tilde{U})$, we have

$$\begin{aligned} 1 &= e\Big(G_z, \frac{\tilde{Z}^\dagger}{\tilde{Z}}\Big)\, e\Big(G, \frac{\tilde{R}^\dagger}{\tilde{R}}\Big) \prod_{i=1}^{k} e\Big(G_i, \frac{M_i^\dagger}{M_i}\Big) \\ &= e\Big(G_z, \frac{\tilde{Z}^\dagger}{\tilde{Z}} \prod_{i=1}^{k} \Big(\frac{M_i^\dagger}{M_i}\Big)^{\chi_i}\Big)\, e\Big(G, \frac{\tilde{R}^\dagger}{\tilde{R}} \prod_{i=1}^{k} \Big(\frac{M_i^\dagger}{M_i}\Big)^{\gamma_i}\Big) \;=\; e(G_z, \tilde{Z}^\star)\, e(G, \tilde{R}^\star). \end{aligned}$$

Similarly, with respect to (33), we have

$$\begin{aligned} 1 &= e\Big(H_z, \frac{\tilde{Z}^\dagger}{\tilde{Z}}\Big)\, e\Big(H_u, \frac{\tilde{U}^\dagger}{\tilde{U}}\Big) \prod_{i=1}^{k} e\Big(h_i, \frac{M_i^\dagger}{M_i}\Big) \\ &= e\Big(H_z, \frac{\tilde{Z}^\dagger}{\tilde{Z}} \prod_{i=1}^{k} \Big(\frac{M_i^\dagger}{M_i}\Big)^{\chi_i}\Big)\, e\Big(H_u, \frac{\tilde{U}^\dagger}{\tilde{U}} \prod_{i=1}^{k} \Big(\frac{M_i^\dagger}{M_i}\Big)^{\delta_i}\Big) \;=\; e(H_z, \tilde{Z}^\star)\, e(H_u, \tilde{U}^\star). \end{aligned}$$

Hence, $(\tilde{Z}^\star, \tilde{R}^\star, \tilde{U}^\star)$ is a correct answer to the SDP instance.

It remains to show that $\tilde{Z}^\star \neq 1$.

We first consider the case of $(M_1, \ldots, M_k) = (M_1^\dagger, \ldots, M_k^\dagger)$. In this case, $(\tilde{Z}^\dagger, \tilde{R}^\dagger, \tilde{U}^\dagger) \neq (\tilde{Z}, \tilde{R}, \tilde{U})$ must hold. Observe that $\tilde{Z}^\dagger = \tilde{Z}$ cannot be the case since it implies $\tilde{R}^\dagger = \tilde{R}$ and $\tilde{U}^\dagger = \tilde{U}$ to fulfill (32) and (33). Thus, we have $\tilde{Z}^\dagger \neq \tilde{Z}$ and $\tilde{Z}^\star = \tilde{Z}^\dagger / \tilde{Z} \neq 1$.

Next, we consider the case of $(M_1, \ldots, M_k) \neq (M_1^\dagger, \ldots, M_k^\dagger)$. In this case, there exists $i^\star$ for which $M_{i^\star} \neq M_{i^\star}^\dagger$ holds. We claim that randomness $\chi_{i^\star}$ is information-theoretically hidden from the view of the adversary. Namely, for any view of the adversary and for any $\chi_{i^\star}$, there exists a consistent coin toss which yields the view.

From now, we consider a group element by its index with respect to base G and $\tilde{G}$ and denote the index with the corresponding lower-case letter. For instance, we represent G_z by $g_z := \log_G G_z$. The independent group elements $(G_z, H_z, H_u, G_1, H_1, \ldots, G_k, H_k, A, B, Z)$ given to $\mathcal{A}$ are translated to $(g_z, h_z, h_u, g_1, h_1, \ldots, g_k, h_k, a, b, z)$. Note that

R and U are uniquely determined from other group elements. Message $(m_1, \ldots, m_k)$ is in the view of $\mathcal{A}$ as well. Then, for the simulated public key and one-time signature, the following relations hold among the variables in the view and the coins chosen by the reduction algorithm:

$$g_i = g_z \chi_i + \gamma_i, \quad h_i = h_z \chi_i + h_u \delta_i \quad (\text{for } i = 1, \ldots, k) \tag{35}$$

$$a = g_z \zeta + \rho, \quad b = h_z \zeta + h_u \varphi, \tag{36}$$

$$z = \zeta + \sum_{i=1}^{k} -m_i \chi_i. \tag{37}$$

Let χ_i for all $i \neq i^\star$ be fixed to any value. Then, for every assignment to $\chi_{i^\star}$, there exist consistent $\gamma_{i^\star}$ and $\delta_{i^\star}$ that fulfill (35). Also, there exist ρ, φ and ζ that satisfy (37) and (36). Accordingly, any choice of $\chi_{i^\star}$ is consistent with the adversary's view. Due to the factor $(M_{i^\star}^\dagger / M_{i^\star})^{\chi_{i^\star}}$, with $M_{i^\star}^\dagger / M_{i^\star} \neq 1$, variable $\tilde{Z}^\star$ distributes uniformly over $\tilde{\mathbb{G}}$ depending on $\chi_{i^\star}$. Thus, $\tilde{Z}^\star = 1$ happens only with negligible probability. □

4.2. *Construction in Asymmetric Setting*

In the case of $\Lambda \in \{\Lambda_{\mathsf{xdh}}, \Lambda_{\mathsf{sxdh}}\}$, we can construct a more efficient scheme, OTS2, that halves OTS1 just like HTC2 does for HTC1.

[Scheme OTS2]

Setup(1^λ)**:** Run $\Lambda := (p, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, e, G, \tilde{G}) \leftarrow \mathcal{G}(1^\lambda)$. Output $gk := \Lambda$.

Key(gk)**:** For $i = 0, \ldots, k$, choose $\gamma_i \leftarrow \mathbb{Z}_p^*$ and compute $G_i := G^{\gamma_i}$. Let $G_z := G_0$. Also choose $\rho \leftarrow \mathbb{Z}_p^*$, and set $A := G^\rho$. Set $vk := (\Lambda, G_z, \{G_i\}_{i=1}^k, A)$ and $sk := (vk, \rho, \{\gamma_i\}_{i=0}^k)$. Output (vk, sk).

Sign(sk, msg)**:** Pick $\zeta \leftarrow \mathbb{Z}_p$ and compute

$$\tilde{Z} := \tilde{G}^\zeta, \quad \tilde{R} := \tilde{G}^{\rho - \gamma_0 \zeta} \textstyle\prod_{i=1}^k M_i^{-\gamma_i}.$$

Output $\sigma := (\tilde{Z}, \tilde{R})$ as a signature.

Vrf(vk, msg, σ)**:** Output 1 if the following equations hold; output 0 otherwise.

$$e(A, \tilde{G}) = e(G_z, \tilde{Z})\, e(G, \tilde{R}) \textstyle\prod_{i=1}^k e(G_i, M_i) \qquad \P$$

Theorem 8. *OTS2 is a one-time signature scheme that is strongly unforgeable against one-time chosen-message attacks if DBP holds for $\mathcal{G}$.*

Theorem 8 can be proven in the same manner as done for Theorem 7 only by dropping non-existing variables.

Remark on Signing Messages from Both Groups. By swapping the roles of $\mathbb{G}$ and $\tilde{\mathbb{G}}$, we obtain a dual scheme OTS2' whose messages are in $\mathbb{G}$. By using these two schemes in a fixed order, we construct a scheme that signs a bilateral message $msg := (M_1, \dots, M_{k_m}, N_1, \dots, N_{k_n}) \in \tilde{\mathbb{G}}^{k_m} \times \mathbb{G}^{k_n}$: Sign $(M_1, \dots, M_{k_m})$ with OTS2 to obtain σ_m and then sign $(N_1, \dots, N_{k_n})$ with OTS2' to obtain σ_n; define $\sigma := (\sigma_m, \sigma_n)$ as the signature for *msg*. The signature is accepted if σ_m is a valid signature for OTS2 and σ_n is a valid signature for OTS2'. With $\Lambda = \Lambda_{\mathsf{sxdh}}$, the resulting scheme is strongly unforgeable against one-time adaptive chosen-message attacks. More precisely, the security is proven based on DBP and its dual assumption implied by DDH in $\mathbb{G}$ and $\tilde{\mathbb{G}}$, respectively.

Remark on Reusing Public Keys. In [2], it is shown that OTS2 can be used as *partially one-time signature scheme* where only element A is updated for every signing and all remaining elements in *vk* can be reused. We refer to [2] for the formal security notion and a proof.

Though it is not covered by [2], the same is true for OTS1 where only A and B must be refreshed before signing. The intuition is that, for the information-theoretical argument in the security proof, elements A and B provide random one-time pads ρ and φ, respectively, that perfectly hide the sensitive information $\chi_{i^\star}$ for signing. Hence by refreshing A and B for every signing, new randomness hides $\chi_{i^\star}$ in the view of the signature. Thus, the information-theoretical argument is preserved.

5. Constant-Size Signatures

This section presents a constant-size structure-preserving signature scheme that yields signatures consisting of 7 group elements independently of the message length, which is a priori fixed by the public key. We then argue how to extend the basic scheme to sign messages with unbounded length allowing linear growth of the signature size with an exact estimation of the growth factor.

5.1. *Construction*

Our construction combines a trapdoor commitment scheme based on SDP and a strong q-type assumption, SFP. A technical obstacle is the "exception," that besides satisfying Equations (2) and (3), a SFP solution must also satisfy $\tilde{Z}^\star \neq 1$. The signature scheme should not explicitly handle exceptions because the condition $\tilde{Z} \neq 1$ is not trivial to prove and affects the efficiency when proving a knowledge of a signature. We address this problem by involving another set of elements $(A_0, \tilde{A}_0)$ and $(B_0, \tilde{B}_0)$ in the verification predicate. In the proof of unforgeability, these elements hold a secret random offset $\tilde{g}^\zeta$ that will be multiplied to $\tilde{Z}$ in a forged signature so that the answer to SFP, $\tilde{Z}^\star = \tilde{Z}\tilde{G}^\zeta$, happens to be 1 only by chance. (The real proof is slightly more involved.) The randomization techniques from Sect. 2.7 also help the construction and the security proof in such a way that the signature elements are uniform conditioned on satisfying the verification equations.

The message space of our signature scheme is $\tilde{\mathbb{G}}^k$ for an arbitrary constant k. Let in the following $msg := (M_1, \dots, M_k) \in \tilde{\mathbb{G}}^k$ be a message to be signed.

[Scheme CSIG]

Setup(1^λ)**:** Run $\Lambda := (p, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, e, G, \tilde{G}) \leftarrow \mathcal{G}(1^\lambda)$. Output $gk := \Lambda$.

Key(1^λ)**:** Choose $H_u \leftarrow \mathbb{G}^*$. For $i = 1, \ldots, k$, choose $\gamma_i, \delta_i \leftarrow \mathbb{Z}_p^*$ and compute $G_i := G^{\gamma_i}$ and $H_i := H_u^{\delta_i}$. Choose $\gamma_z, \delta_z \leftarrow \mathbb{Z}_p^*$ and compute $G_z := G^{\gamma_z}$ and $H_z := H_u^{\delta_z}$. Also choose $\alpha, \beta \leftarrow \mathbb{Z}_p^*$ and compute $\{A_i, \tilde{A}_i\}_{i=0}^1 \leftarrow \mathsf{Extend}(G, \tilde{G}^\alpha)$ and $\{B_i, \tilde{B}_i\}_{i=0}^1 \leftarrow \mathsf{Extend}(H_u, \tilde{G}^\beta)$. Set $vk := (\Lambda^*, G_z, H_z, H_u, \{G_i, H_i\}_{i=1}^k, \{A_i, \tilde{A}_i, B_i, \tilde{B}_i\}_{i=0}^1)$ and $sk := (vk, \alpha, \beta, \gamma_z, \delta_z, \{\gamma_i, \delta_i\}_{i=1}^k)$. Output (vk, sk).

Sign(sk, msg)**:** Choose $\zeta, \rho, \tau, \varphi, \omega$ randomly from $\mathbb{Z}_p$ and set:

$$\tilde{Z} := \tilde{G}^\zeta,\ \tilde{R} := \tilde{G}^{\alpha-\rho\tau-\gamma_z\zeta} \textstyle\prod_{i=1}^k M_i^{-\gamma_i},\ S := G^\rho,\ \tilde{T} := \tilde{G}^\tau,$$
$$\tilde{U} := \tilde{G}^{\beta-\varphi\omega-\delta_z\zeta} \textstyle\prod_{i=1}^k M_i^{-\delta_i},\ V := H_u^\varphi,\ \tilde{W} := \tilde{G}^\omega.$$

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

Vrf(vk, msg, σ)**:** Output 1 if

$$e(A_0, \tilde{A}_0)\, e(A_1, \tilde{A}_1) = e(G_z, \tilde{Z})\, e(G, \tilde{R})\, e(S, \tilde{T}) \prod_{i=1}^k e(G_i, M_i), \quad \text{and} \tag{38}$$

$$e(B_0, \tilde{B}_0)\, e(B_1, \tilde{B}_1) = e(H_z, \tilde{Z})\, e(H_u, \tilde{U})\, e(V, \tilde{W}) \prod_{i=1}^k e(H_i, M_i). \tag{39}$$

hold; output 0 otherwise. ¶

Correctness is verified by inspecting that

$$\begin{aligned}(\text{right-hand of (38)}) &= e(G_z, \tilde{Z})\, e(G, \tilde{R})\, e(S, \tilde{T}) \prod_{i=1}^k e(G_i, M_i) \\ &= e\left(G^{\gamma_z}, \tilde{G}^\zeta\right) \\ &\quad e\left(G, \tilde{G}^{\alpha-\rho\tau-\gamma_z\zeta} \prod_{i=1}^k M_i^{-\gamma_i}\right) e\left(G^\rho, \tilde{G}^\tau\right) \prod_{i=1}^k e\left(G^{\gamma_i}, M_i\right) \\ &= e\left(G, \tilde{G}^\alpha\right) \\ &= (\text{left hand of (38)}).\end{aligned}$$

holds. Relation (39) is verified in the same manner.

Given a signature $(\tilde{Z}, \tilde{R}, S, \tilde{T}, \tilde{U}, V, \tilde{W})$, one can randomize every element except for $\tilde{Z}$ by applying the sequential randomization technique. Let $(\tilde{R}', S', \tilde{T}', \tilde{U}', V', \tilde{W}') \leftarrow \mathsf{SigRand}(\tilde{R}, S, \tilde{T}, \tilde{U}, V, \tilde{W})$ be an algorithm defined as follows. If $\tilde{T} = 1$, set $S = 1$ and choose $\tilde{T} \leftarrow \tilde{\mathbb{G}}^*$. Then, choose $\varrho \leftarrow \mathbb{Z}_p$ and compute $\tilde{R}' = \tilde{R}\,\tilde{T}^\varrho$ and $(S', \tilde{T}') \leftarrow \mathsf{Rand}(SG^{-\varrho}, \tilde{T})$; compute $(\tilde{U}', V', \tilde{W}')$ from $(\tilde{U}, V, \tilde{W})$ analogously. The resulting $(\tilde{R}', S', \tilde{T}', \tilde{U}', V', \tilde{W}')$ distributes uniformly over $(\tilde{\mathbb{G}} \times \mathbb{G} \times \tilde{\mathbb{G}})^2$ under the constraint that $e(G, \tilde{R})\, e(S, \tilde{T}) = e(G, \tilde{R}')\, e(S', \tilde{T}')$ and $e(H_u, \tilde{U})\, e(V, \tilde{W}) = e(H_u, \tilde{U}')\, e(V', \tilde{W}')$.

Accordingly, $(S', \tilde{T}', V', \tilde{W}')$ is independent of $\tilde{Z}$, the message and the verification key. This is a useful property, since it can reduce the size of proofs of knowledge of signatures: The randomized $(S', \tilde{T}', V', \tilde{W}')$ can be exposed, as they do not reveal anything about the hidden signature.

Theorem 9. *CSIG is EUF-CMA against adversaries making up to q signing queries if q-SFPA holds for $\mathcal{G}$.*

Proof. Let $\mathcal{A}$ be an adversary that after adaptively querying the signing oracle on messages $\vec{M}_j$, for $j = 1, \ldots, q$, and receiving signatures σ_j has non-negligible advantage of forging a signature on a message $\vec{M}^\dagger \notin \{\vec{M}_j\}_{j=1}^q$. We construct a reduction algorithm which takes an input Λ, G_z, H_z, H_u, $(A, \tilde{A})$, $(B, \tilde{B})$ and uniformly chosen tuples I_j for $j = 1, \ldots, q$ as defined in Assumption 8, and simulates the view of $\mathcal{A}$ in the attack environment as follows:

Simulating $\mathsf{CSIG.Key}$: Use (G_z, H_z, H_u) as given in the input. For $i = 1, \ldots, k$ set $G_i := G_z^{\chi_i} G^{\gamma_i}$ and $H_i := H_z^{\chi_i} H_u^{\delta_i}$, where $\chi_i, \gamma_i, \delta_i \leftarrow \mathbb{Z}_p$. As the probability that any G_i or H_i, $i = 1, \ldots, k$, is equal to $1_{\mathbb{G}}$ is negligible, the reduction simply aborts in such cases. Otherwise, all group elements are uniformly random in $\mathbb{G}^*$, like in the key generation algorithm. Then, select ζ, ρ, $\varphi \leftarrow \mathbb{Z}_p$, and compute $((A_0, \tilde{A}_0), (A_1, \tilde{A}_1)) \leftarrow \mathsf{RandSeq}((G_z^{\zeta} G^{\rho}, \tilde{G}), (A, \tilde{A}))$ and $((B_0, \tilde{B}_0), (B_1, \tilde{B}_1)) \leftarrow \mathsf{RandSeq}((H_z^{\zeta} H_u^{\varphi}, \tilde{G}), (B, \tilde{B}))$.

For convenience, denote $G_z^{\zeta} G^{\rho}$ by A' and $H_z^{\zeta} H_u^{\varphi}$ by B'.

The verification key is $vk = (G_z, H_z, H_u, \{G_i, H_i\}_{i=1}^k, \{A_i, \tilde{A}_i, B_i, \tilde{B}_i\}_{i=0}^1)$.

Simulating $\mathsf{CSIG.Sign}$: Given a message $\vec{M}$, take a fresh tuple $I_j = (\tilde{Z}_j, \tilde{R}_j, \tilde{U}_j, S_{ij}, \tilde{T}_{ij}, V_{ij}, \tilde{W}_{ij}) \in \tilde{\mathbb{G}}^* \times \tilde{\mathbb{G}} \times \tilde{\mathbb{G}} \times \mathbb{G} \times \tilde{\mathbb{G}} \times \mathbb{G} \times \tilde{\mathbb{G}}$ from the input instance. Then, compute

$$\tilde{Z} := \tilde{Z}_j\, \tilde{G}^{\zeta} \prod_{i=1}^{k} M_i^{-\chi_i}, \quad \tilde{R} := \tilde{R}_j\, \tilde{G}^{\rho} \prod_{i=1}^{k} M_i^{-\gamma_i}, \quad S := S_j, \quad \tilde{T} := \tilde{T}_j,$$
$$\tilde{U} := \tilde{U}_j\, \tilde{G}^{\varphi} \prod_{i=1}^{k} M_i^{-\delta_i}, \quad V := V_j, \quad \tilde{W} := \tilde{W}_j.$$

The signature is $\sigma := (\tilde{Z}, \tilde{R}, S, \tilde{T}, \tilde{U}, V, \tilde{W})$. It is easy to verify that the signature satisfies the verification equations.

When $\mathcal{A}$ outputs $(\vec{M}^\dagger, (\tilde{Z}^\dagger, \tilde{R}^\dagger, S^\dagger, \tilde{T}^\dagger, \tilde{U}^\dagger, V^\dagger, \tilde{W}^\dagger))$, compute

$$\tilde{Z}^\star := \tilde{Z}^\dagger\, \tilde{G}^{-\zeta} \prod_{i=1}^{k} (M_i^\dagger)^{\chi_i}, \quad \tilde{R}^\star := \tilde{R}^\dagger\, \tilde{G}^{-\rho} \prod_{i=1}^{k} (M_i^\dagger)^{\gamma_i}, \quad \tilde{U}^\star := \tilde{U}^\dagger\, \tilde{G}^{-\varphi} \prod_{i=1}^{k} (M_i^\dagger)^{\delta_i},$$

and set $S^\star := S^\dagger$, $\tilde{T}^\star := \tilde{T}^\dagger$, $V^\star := V^\dagger$, and $\tilde{W}^\star := \tilde{W}^\dagger$. If any of the parameters $\chi_1, \ldots, \chi_k$ is 0, the reduction algorithm aborts; otherwise, it outputs $(\tilde{Z}^\star, \tilde{R}^\star, S^\star, \tilde{T}^\star, \tilde{U}^\star, V^\star, \tilde{W}^\star)$. The probability of aborting is negligible because the parameters are chosen uniformly at random. We can therefore ignore those cases in our analysis without affecting the overall outcome. This completes the description of the reduction algorithm.

The above signatures are correctly distributed, and thus, $\mathcal{A}$ outputs a successful forgery with a non-negligible probability. Then, for the output of the reduction algorithm, it holds that

$$\begin{aligned} & e(G_z, \tilde{Z}^\star)\, e(G, \tilde{R}^\star)\, e(S^\star, \tilde{T}^\star) \\ & = e\left(G_z, \tilde{Z}^\dagger\, \tilde{G}^{-\zeta} \prod_{i=1}^{k} (M_i^\dagger)^{\chi_i}\right) e\left(G, \tilde{R}^\dagger\, \tilde{G}^{-\rho} \prod_{i=1}^{k} (M_i^\dagger)^{\gamma_i}\right) e\left(S^\dagger, \tilde{T}^\dagger\right) \\ & = e\big(G_z^{-\zeta} G^{-\rho}, \tilde{G}\big)\, e\big(G_z, \tilde{Z}^\dagger\big)\, e\big(G, \tilde{R}^\dagger\big) e\big(S^\dagger, \tilde{T}^\dagger\big) \prod_{i=1}^{k} e\big(G_i, M_i^\dagger\big) \\ & = e\big(G_z^{\zeta} G^{\rho}, \tilde{G}\big)^{-1} \prod_{i=0}^{1} e(A_i, \tilde{A}_i) \;=\; e(A, \tilde{A}). \end{aligned}$$

One can also verify that $e(G_z, \tilde{Z}^\star)\, e(H_u, \tilde{U}^\star)\, e(V^\star, \tilde{W}^\star) = e(B, \tilde{B})$ holds in the same way.

It remains to show that $\tilde{Z}^\star$ is not in $\{1, \tilde{Z}_1, \ldots, \tilde{Z}_q\}$. For that, first notice that the parameters ζ and $\{\chi_i\}_{i=1}^k$ are independent from adversary $\mathcal{A}$'s view, as proved in Lemma 6 below. Namely, for any view of the adversary and for any choice of ζ and χ_i, for $i = 1, \ldots, k$, there exist unique and consistent parameters $\rho, \varphi, \gamma_i, \delta_i, i = 1, \ldots, k$ and $\tilde{Z}_j, \tilde{R}_j, \tilde{U}_j, j = 1, \ldots, q$.

We show that the probability that $\tilde{Z}^\star \in \{\tilde{Z}_1, \ldots, \tilde{Z}_q\}$ is negligible. For every $\tilde{Z}_j$ and signature $\sigma = (\tilde{Z}, \tilde{R}, S, \tilde{T}, \tilde{U}, V, \tilde{W})$ on a message $\vec{M}$ simulated by using $\tilde{Z}_j$, it holds that

$$\frac{\tilde{Z}^\star}{\tilde{Z}_j} \;=\; \frac{\tilde{Z}^\dagger\, \tilde{G}^{-\zeta} \prod_{i=1}^{k} (M_i^\dagger)^{\chi_i}}{\tilde{Z}\, \tilde{G}^{-\zeta} \prod_{i=1}^{k} M_i^{\chi_i}} \;=\; \frac{\tilde{Z}^\dagger}{\tilde{Z}} \prod_{i=1}^{k} \left(\frac{M_i^\dagger}{M_i}\right)^{\chi_i}.$$

Since $\vec{M}^\dagger \neq \vec{M}$, there exists i such that $M_i^\dagger \neq M_i$. Since $\chi_i \in \mathbb{Z}_p^*$ is information-theoretically hidden from the adversary's view, the probability that $\tilde{Z}^\star = \tilde{Z}_j$ is negligible due to the term $(M_i^\dagger / M_i)^{\chi_i}$ in the above equation. To show that $\tilde{Z}^\star = (\tilde{Z}^\dagger)\, \tilde{G}^{-\zeta} \prod_{i=1}^k (M_i^\dagger)^{\chi_i}$ is equal to $1_{\tilde{\mathbb{G}}}$ only with negligible probability, notice that ζ is also independent from the adversary's view and the claim holds due to the uniform choice of ζ. Therefore, when the reduction algorithm does not abort, the probability that $\tilde{Z}^\star \notin \{1, \tilde{Z}_1, \ldots, \tilde{Z}_q\}$ is overwhelming. □

Lemma 6. *The parameters ζ, χ_1, χ_2, ..., χ_k chosen by the reduction algorithm in Theorem 9 are independent from $\mathcal{A}$'s view, that is, independent from the verification key, the signed messages and the signatures.*

Proof. Let $vk := (G_z, H_z, H_u, \{G_i, H_i\}_{i=1}^k, \{A_i, \tilde{A}_i, B_i, \tilde{B}_i\}_{i=0}^1)$ be the verification key the adversary sees, $\vec{M}_1, \ldots, \vec{M}_q$ be the messages on which $\mathcal{A}$ queries the signing oracle, and $\sigma_1, \ldots, \sigma_q$ be the corresponding signatures. Furthermore, assume that $(A, \tilde{A})$ and $(B, \tilde{B})$ given to the reduction algorithm are also fixed, though $\mathcal{A}$ does not see them. That yields unique A' and B' such that

$$\begin{aligned} A_T &= e(A_0, \tilde{A}_0)\, e(A_1, \tilde{A}_1) = e(A', \tilde{G})\, e(A, \tilde{A}) \\ B_T &= e(B_0, \tilde{B}_0)\, e(B_1, \tilde{B}_1) = e(B', \tilde{G})\, e(B, \tilde{B}) \end{aligned}$$

For any choice $\zeta^\star, \chi_i^\star \in \mathbb{Z}_p$ of the parameters ζ, χ_i , for $i = 1, \ldots, k$, there exists a *unique* coin toss $\rho^\star, \varphi^\star, \gamma_i^\star, \delta_i^\star$ such that $A' = G_z^{\zeta^\star} G^{\rho^\star}$, $B' = H_z^{\zeta^\star} H_u^{\varphi^\star}$, $G_i = G_z^{\chi_i^\star} G^{\gamma_i^\star}$, and $H_i = H_z^{\chi_i^\star} H_u^{\delta_i^\star}$. This shows that the verification key and the parameters are independent. Next, we show that the chosen parameters remain independent from $\mathcal{A}$'s view even after signing q adaptively chosen messages due to the uniform choice of the tuples $I_j = (\tilde{Z}_j, \tilde{R}_j, S_j, \tilde{T}_j, \tilde{U}_j, V_j, \tilde{W}_j)$, $j = 1, \ldots, q$, as defined in Assumption 8.

Let the j-th message be $\vec{M}$ and the corresponding signature be $\sigma = (\tilde{Z}, \tilde{R}, S, \tilde{T}, \tilde{U}, V, \tilde{W})$. From the specification of the reduction algorithm we have $(S, \tilde{T}) = (S_j, \tilde{T}_j)$ and $(V, \tilde{W}) = (V_j, \tilde{W}_j)$, where $I_j = (\tilde{Z}_j, \tilde{R}_j, S_j, \tilde{T}_j, \tilde{U}_j, V_j, \tilde{W}_j)$ is the j-th tuple given as input. And for the fixed view, ζ, $\{\chi_i\}_{i=1}^k$ determine *uniquely* the values of $\tilde{Z}_j = \tilde{Z}\,\tilde{G}^{-\zeta} \prod_{i=1}^k M_i^{\chi_i}$, $\tilde{R}_j = \tilde{R}\,\tilde{G}^{-\rho} \prod_{i=1}^k M_i^{\gamma_i}$, and $\tilde{U}_j = \tilde{U}\,\tilde{G}^{-\varphi} \prod_{i=1}^k M_i^{\delta_i}$. Regardless of the particular choice of parameters $\zeta^\star$, $\{\chi_i^\star\}_{i=1}^k$, since σ satisfies the signature-verification equations:

$$\begin{aligned} A_T &= e(G_z, \tilde{Z})\, e(G, \tilde{R})\, e(S, \tilde{T}) \prod_{i=1}^k e(G_i, M_i) \quad \text{and} \\ B_T &= e(H_z, \tilde{Z})\, e(H_u, \tilde{U})\, e(V, \tilde{W}) \prod_{i=1}^k e(H_i, M_i), \end{aligned}$$

it is true that the corresponding tuple $I_j^\star = (\tilde{Z}_j^\star, \tilde{R}_j^\star, S_j, \tilde{T}_j, \tilde{U}_j^\star, V_j, \tilde{W}_j)$ satisfies:

$$\begin{aligned} e(A, \tilde{A}) &= e(G_z, \tilde{Z}_j^\star)\, e(G, \tilde{R}_j^\star)\, e(S_j, \tilde{T}_j) \quad \text{and} \\ e(B, \tilde{B}) &= e(H_z, \tilde{Z}_j^\star)\, e(H_u, \tilde{U}_j^\star)\, e(V_j, \tilde{W}_j). \end{aligned} \tag{40}$$

It remains to show that the uniform choice of $\zeta^\star$, $\{\chi_i^\star\}_{i=1}^k$ together with $\mathcal{A}$'s view yields uniform distribution for the tuples $I_j^\star$, for $j = 1, \ldots, q$, as specified by the assumption description. If that is indeed the case, each set of tuples which could have been given as input to the reduction algorithm is chosen with the same probability. And because for any choice of $\zeta^\star, \chi_1^\star, \ldots, \chi_k$, there exists *unique* set $\{I_j^\star\}_{j=1}^q$, those imply that each parameter selection looks equally likely for $\mathcal{A}$.

To see the uniformity of $I_j^\star$, note again that $(S_j, \tilde{T}_j)$ and $(V_j, \tilde{W}_j)$ are determined uniquely from the view regardless of the choice of parameters. We define the following homomorphism ϕ:

$$\phi_{\tilde{G},\vec{M}}(\zeta^\star, \chi_1^\star, \ldots, \chi_k^\star) = \tilde{G}^{-\zeta^\star} \prod_{i=1}^k M_i^{\chi_i^\star}.$$

It is easy to verify that for uniformly chosen parameters, the range of ϕ is uniformly distributed over $\tilde{\mathbb{G}}$. This in turn implies that for a fixed $\tilde{Z}$ and uniformly chosen parameters, $\tilde{Z}_j^\star = \tilde{Z}, \phi(\zeta^\star, \chi_1, \ldots, \chi_k)$ is uniformly distributed over $\tilde{\mathbb{G}}$. And because $I_j^\star$ satisfies (40), the values of $\tilde{R}_j^\star$ and $\tilde{U}_j^\star$ are determined uniquely by the other tuple values, which for a fixed view means determined by $\tilde{Z}_j^\star$. To sum it up, for a fixed view, the uniform random choice of the parameters gives uniformly distributed $\tilde{Z}_j^\star$ which implies the uniformity of $I_j^\star$. □

5.2. *Extension*

Our scheme CSIG in Sect. 5.1 is not automorphic since the size of a message vector is a priori bounded by the verification key and the verification key itself exceeds the bound. There is a standard method to sign unbounded-size messages by using signature schemes with limited message space. The following scheme, denoted by USIG, is a minor variation of the one in [65]. Suppose that CSIG is set up to sign messages of size at most $k \geq 3$. To sign message $\vec{M} = (M_1, \ldots, M_n)$ for $n > k$, first encode the size of the message into a group element $M_0 := \langle |\vec{M}| \rangle$. Then, select a random group element t and compute

$$\begin{aligned} \sigma_0 &:= \mathsf{CSIG.Sign}(sk, t \,||\, \langle 1 \rangle \,||\, M_0, ..., M_{k-3}), \\ \sigma_1 &:= \mathsf{CSIG.Sign}(sk, t \,||\, \langle 2 \rangle \,||\, M_{k-2}, ..., M_{2k-5}), \ \ldots \end{aligned} \tag{41}$$

The resulting signature is $\sigma = (t, \sigma_0, \sigma_1, \ldots)$, which consists of $\lceil \frac{n+1}{k-2} \rceil \cdot 7 + 1$ group elements.

The scheme USIG is automorphic in the symmetric setting but falls short in the asymmetric setting $\Lambda = \Lambda_{\mathsf{sxdh}}$ where a message vector may consist of elements from $\mathbb{G}$ and $\tilde{\mathbb{G}}$. We construct a scheme, XUSIG, that signs messages consisting of elements in $\mathbb{G}$ and $\tilde{\mathbb{G}}$ for $\Lambda = \Lambda_{\mathsf{sxdh}}$ as follows. Let USIG1 and USIG2 be schemes for unbounded-size messages in $\mathbb{G}$ and $\tilde{\mathbb{G}}$, respectively. To sign $\vec{M} = (M_1, \ldots, M_n, \tilde{M}_1, \ldots, \tilde{M}_{\tilde{n}}) \in \mathbb{G}^n \times \tilde{\mathbb{G}}^{\tilde{n}}$, first pick random tags $T \in \mathbb{G}$ and $\tilde{T} \in \tilde{\mathbb{G}}$ such that $e(T, \tilde{G}) = e(G, \tilde{T})$ holds. Then, sign $(T, M_1, \ldots, M_n)$ and $(\tilde{T}, \tilde{M}_1, \ldots, \tilde{M}_{\tilde{n}})$ by using USIG1 and USIG2. The resulting signature is a concatenation of two signatures from USIG1 and USIG2. The verification function first checks if $e(T, \tilde{G}) = e(G, \tilde{T})$ holds and then follows the verification for USIG1 and USIG2.

It is not hard to see that, if the underlying schemes are existentially unforgeable against chosen-message attacks, so are USIG and XUSIG. Thus, schemes USIG and XUSIG are secure automorphic signature schemes in symmetric and asymmetric bilinear-group setting, respectively.

6. Automorphic Signatures

In this section, we present an efficient construction of an automorphic signature scheme, which was also the first efficient structure-preserving signature scheme in the literature. In this scheme public keys are Diffie–Hellman pairs, that is, $(G^x, \tilde{G}^x)$, for some $x \in \mathbb{Z}_p$.

Messages are also of this form: $(G^m, \tilde{G}^m)$. We then present a generic method to extend the message space so we can sign several messages of that form at once.

6.1. *Construction*

Boneh and Boyen [22] show that their SDH assumption implies a signature scheme which is existentially unforgeable against adversaries that only get signatures on random messages (the scheme corresponds to Problem 1 stated before Assumption 9; page 15). Analogously, ADH-SDH immediately yields an (automorphic) scheme secure against "random-message attacks" if we consider $(X, \tilde{Y})$ as the public key, $(V, \tilde{W})$ as a message in $\mathcal{DH} = \{(G^v, \tilde{G}^v) \mid v \in \mathbb{Z}_p\}$ and $(A, B, \tilde{D})$ as the signature. We show how to transform this into a strongly EUF-CMA-secure signature scheme by assuming AWF-CDH. We add some more randomness to the signature that in our reduction lets us map a query for a message chosen by the adversary to a given tuple $(A_i, B_i, \tilde{D}_i, V_i, \tilde{W}_i)$ from an ADH-SDH instance. AWF-CDH then asserts that the adversary cannot produce a new signature/message pair $\big((A^*, B^*, \tilde{D}^*, R^*, \tilde{S}^*), (M^*, \tilde{N}^*)\big)$ that maps back to a tuple from the instance (see the proof of Theorem 10). We get the following efficient automorphic signature construction, whose signatures are in $\mathbb{G}^3 \times \tilde{\mathbb{G}}^2$. The scheme can also be defined over symmetric bilinear groups, in which case we replace $\tilde{G}$ by a random generator H of $\mathbb{G}$.

[Scheme ASIG]

Setup(1^λ)**:** Run $\Lambda := (p, \mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_T, e, G, \tilde{G}) \leftarrow \mathcal{G}(1^\lambda)$. Choose $F, K, T \leftarrow \mathbb{G}^*$ and output $gk := (\Lambda, F, K, T)$. The message space is defined as $\mathcal{DH} := \{(G^m, \tilde{G}^m) \mid m \in \mathbb{Z}_p\}$.

Key(gk)**:** Choose a random $x \leftarrow \mathbb{Z}_p$ and compute $(X, \tilde{Y}) := (G^x, \tilde{G}^x)$. Output $vk := (X, \tilde{Y})$ and $sk := (gk, vk, x)$.

Sign(sk, msg)**:** Given secret key $sk = x$ and message $msg = (M, \tilde{N}) \in \mathcal{DH}$, choose $c \leftarrow \mathbb{Z}_p \backslash \{-x\}$ and $r \leftarrow \mathbb{Z}_p$, and compute

$$A := (K \cdot T^r \cdot M)^{\frac{1}{x+c}}, \; B := F^c, \; \tilde{D} := \tilde{G}^c, \; R := G^r, \; \tilde{S} := \tilde{G}^r. \qquad (42)$$

Output $\sigma := (A, B, \tilde{D}, R, \tilde{S})$.

Vrf(gk, vk, msg, σ)**:** For $msg \in \mathcal{DH}$, output 1 if the following hold and 0 otherwise:

$$e(A, \tilde{Y} \cdot \tilde{D}) = e(K \cdot M, \tilde{G})\, e(T, \tilde{S}) \quad e(B, \tilde{G}) = e(F, \tilde{D}) \quad e(R, \tilde{G}) = e(G, \tilde{S}) \quad (43)$$

¶

Theorem 10. *Signature scheme* ASIG *is sEUF-CMA against adversaries making at most* $q - 1$ *adaptive chosen-message queries if* q*-ADH-SDH and AWF-CDH hold for* $\mathcal{G}$.

Proof. Consider an adversary that after receiving parameters (Λ, F, K, T) and public key $(X, \tilde{Y})$ is allowed to ask for $q - 1$ signatures $(A_i, B_i, \tilde{D}_i, R_i, \tilde{S}_i)$ on messages $(M_i, \tilde{N}_i) \in \mathcal{DH}$ of its choice and outputs $(M, \tilde{N}) \in \mathcal{DH}$ and a valid signature

$(A, B, \tilde{D}, R, \tilde{S})$ on it, such that either $(M, \tilde{N})$ was never queried, or $(M, \tilde{N}) = (M_i, \tilde{N}_i)$ and $(A, B, \tilde{D}, R, \tilde{S}) \neq (A_i, B_i, \tilde{D}_i, R_i, \tilde{S}_i)$.

We distinguish two kinds of forgers: An adversary is called of Type A if its output satisfies

$$e(T, \tilde{S} \cdot \tilde{S}_i^{-1}) \neq e(M_i \cdot M^{-1}, \tilde{G}) \quad \vee \quad B \neq B_i \tag{44}$$

for all $i \in \{1, \ldots, q-1\}$. Otherwise, it is called of Type B. We will use the first type to break q-ADH-SDH and the second type to break AWF-CDH.

Adversary Type A. Let $\big(\Lambda, F, K, X, \tilde{Y}, (A_i, B_i, \tilde{D}_i, V_i, \tilde{W}_i)_{i=1}^{q-1}\big)$ be a q-ADH-SDH challenge. By (7), it satisfies

$$e(A_i, \tilde{Y} \cdot \tilde{D}_i) = e(K \cdot V_i, \tilde{G}), \quad e(B_i, \tilde{G}) = e(F, \tilde{D}_i), \quad e(V_i, \tilde{G}) = e(G, \tilde{W}_i) \tag{45}$$

for all $i \in \{1, \ldots, q-1\}$. Let $\mathcal{A}$ be a forger of Type A. Choose $t \leftarrow \mathbb{Z}_p$ and give parameters $(\Lambda, F, K, T := G^t)$ and the public key $(X, \tilde{Y})$ to $\mathcal{A}$. The i-th query for $(M_i, \tilde{N}_i) \in \mathcal{DH}$ is answered as

$$\big(A_i, B_i, \tilde{D}_i, R_i := (V_i \cdot M_i^{-1})^{\frac{1}{t}}, \tilde{S}_i = (\tilde{W}_i \cdot \tilde{N}_i^{-1})^{\frac{1}{t}}\big). \tag{46}$$

This satisfies the verification equations (43): $e(B_i, \tilde{G}) \overset{(45)}{=} (F, \tilde{D}_i)$,

$$\begin{aligned} e(R_i, \tilde{G}) &= e((V_i \cdot M_i^{-1})^{\frac{1}{t}}, \tilde{G}) = e(V_i, \tilde{G})^{\frac{1}{t}}\, e(M_i, \tilde{G})^{-\frac{1}{t}} \\ &= e(G, \tilde{W}_i)^{\frac{1}{t}}\, e(G, \tilde{N}_i)^{-\frac{1}{t}} = e(G, \tilde{S}_i), \end{aligned} \tag{47}$$

where the third equation follows from $(V_i, \tilde{W}_i), (M_i, \tilde{N}_i) \in \mathcal{DH}$. Finally

$$\begin{aligned} e(A_i, \tilde{Y} \cdot \tilde{D}_i) \overset{(45)}{=} e(K \cdot V_i, \tilde{G}) \overset{(46)}{=} e(K \cdot (R_i^t \cdot M_i), \tilde{G}) = e(K \cdot M_i, \tilde{G})\, e(R_i \cdot \tilde{G})^t \\ \overset{(47)}{=} e(K \cdot M_i, \tilde{G})\, e(G \cdot \tilde{S}_i)^t = e(K \cdot M_i, \tilde{G})\, e(T \cdot \tilde{S}_i). \end{aligned}$$

Moreover, the oracle answer in (46) it is correctly distributed since v_i is uniformly random in the ADH-SDH instance. The signing oracle is thus perfectly simulated.

If the adversary produces a valid signature/message pair $((A, B, \tilde{D}, R, \tilde{S}), (M, \tilde{N}))$ then by the last 2 equations of (43), there exist c, r such that $B = F^c, \tilde{D} = \tilde{G}^c, R = G^r, \tilde{S} = \tilde{G}^r$, and

$$e(A, \tilde{Y} \cdot \tilde{D}) = e(K \cdot M, \tilde{G})\, e(T, \tilde{S}). \tag{48}$$

The tuple $(A, B, \tilde{D}, V := R^t \cdot M, \tilde{W} := \tilde{S}^t \cdot \tilde{N})$ satisfies the equations of ADH-SDH tuples in (7), since $(B, \tilde{D})$ and $(V, \tilde{W})$ are Diffie–Hellman pairs and

$$e(K \cdot V, \tilde{G}) = e(K \cdot (G^r)^t \cdot M, \tilde{G}) = e(K \cdot M, \tilde{G})\, e(T, \tilde{S}) \overset{(48)}{=} e(A, \tilde{Y} \cdot \tilde{D}).$$

Moreover, it is a solution for the ADH-SDH instance, since it is a *new* tuple: assume that for some i we have $B = B_i$ and $\tilde{W} = \tilde{W}_i$, that is $\tilde{S}^t \cdot \tilde{N} = \tilde{S}_i^t \cdot \tilde{N}_i$. Since $(M, \tilde{N}), (M_i, \tilde{N}_i) \in \mathcal{DH}$, we have $e(T, \tilde{S})\, e(M, \tilde{G}) = e(G^t, \tilde{S})\, e(G, \tilde{N}) = e(G, \tilde{S}^t \cdot$

$\tilde{N}) = e(G, \tilde{S}_i^t \cdot \tilde{N}_i) = e(T, \tilde{S}_i)\, e(G, \tilde{N}_i) = e(T, \tilde{S}_i)\, e(M_i, \tilde{G})$. We have thus $e(T, \tilde{S} \cdot \tilde{S}_i^{-1}) = e(M_i \cdot M^{-1}, \tilde{G})$ and $B = B_i$ which contradicts (44) and thus the fact that $\mathcal{A}$ is of Type A.

Adversary Type B. Let $(\Lambda, T = G^t)$ be an AWF-CDH instance (note that T corresponds to A in Assumption 3). Let $\mathcal{A}$ be a forger of Type B. Pick $F, K \leftarrow \mathbb{G}$ and $x \leftarrow \mathbb{Z}_p$, set $X := G^x$, $\tilde{Y} := \tilde{G}^x$ and give the adversary parameters (Λ, F, K, T) and public key $(X, \tilde{Y})$. Answer a signing query on $(M_i, \tilde{N}_i) \in \mathcal{DH}$ by returning a signature $(A_i, B_i, \tilde{D}_i, R_i, \tilde{S}_i)$ produced by $\mathsf{ASIG.Sign}$ with x as a secret key. Suppose $\mathcal{A}$ returns $((A, B, \tilde{D}, R, \tilde{S}), (M, \tilde{N}))$ satisfying (43) such that for some i:

$$e(T, \tilde{S} \cdot \tilde{S}_i^{-1}) = e(M_i \cdot M^{-1}, \tilde{G}) \qquad B = B_i \tag{49}$$

Then, $(M^* := M_i \cdot M^{-1}, \tilde{N}^* := \tilde{N}_i \cdot \tilde{N}^{-1}, R^* := R \cdot R_i^{-1}, \tilde{S}^* := \tilde{S} \cdot \tilde{S}_i^{-1})$ is an AWF-CDH solution:

It satisfies the equations in (1): $e(T, \tilde{S}^*) = e(T, \tilde{S} \cdot \tilde{S}^{-1}) \overset{(49)}{=} e(M_i \cdot M_i^{-1}, \tilde{G}) = e(M^*, \tilde{G})$, and the two remaining equations in (1) follow since $(M_i, \tilde{N}_i)$, $(R_i, \tilde{S}_i)$, $(M, \tilde{N})$ and $(R, \tilde{S})$ are all Diffie–Hellman pairs and therefore $(M^*, \tilde{N}^*)$ and $(R^*, \tilde{S}^*)$ are in $\mathcal{DH}$ well. Moreover, $(M^*, \tilde{N}^*, R^*, \tilde{S}^*)$ is non-trivial: If $M^* = 1 = R^*$, then $M = M_i$ and $R = R_i$; since, moreover, $B = B_i$ and since the values M, B and R completely determine a message/signature pair, this means that $\mathcal{A}$ returned a message/signature pair from one of its queries, meaning $\mathcal{A}$ did not break strong unforgeability. □

6.2. *Extension*

In this section, we show how to extend the message space of automorphic signature schemes. The messages of the scheme from the previous section consist of one pair of group elements. Known generic methods for extending the message space require extra elements in the message allowing to glue message blocks together as shown in Sect. 5.2; they thus do not work here. Our approach is to generate a fresh ephemeral key for each message block and authenticate these keys by using the persistent key. As this does not guarantee the order of the message blocks, we bind the message-block number to the corresponding ephemeral key using the group operation.

Let ASIG be an automorphic signature scheme whose verification key consists of an element of a group $\mathbb{H}$ of order p. For the scheme in Sect. 6.1, this group $\mathbb{H}$ is $\mathcal{DH} := \{(G^x, \tilde{G}^x) \mid x \in \mathbb{Z}_p\}$, a subgroup of the direct product of source groups $\mathbb{G} \times \tilde{\mathbb{G}}$. Let $\langle n \rangle$ denote an efficiently computable injective mapping from $n \in \{1, \ldots, n_{max}\}$ to $\mathbb{H}^*$ where n_{max} is larger than any polynomial in λ. We require that, for any n and n' in $\{1, \ldots, n_{max}\}$, it holds that $\langle n \rangle \cdot \langle n' \rangle \neq 1 \in \mathbb{H}$. (A simple example for $n_{max} \ll p$ would be $\langle n \rangle := H^n$, where H generates $\mathbb{H}$.) To simplify notation when dealing with several keys that also act as messages, we write them as subscript when they act as keys; for instance, we write $\mathsf{ASIG.Sign}_{sk}(vk_0)$ for signing message vk_0 with key sk.

In the following, we construct $\mathsf{ASIG2}$ that signs messages $msg = \{M_1, \ldots, M_n\} \in \mathbb{H}^n$ for an arbitrary n that does not depend on the verification-key length. It outputs a signature of size $(3n + 2) \cdot |\sigma_{\mathsf{asig}}| + (n + 1) \cdot |vk_{\mathsf{asig}}|$ where $|\sigma_{\mathsf{asig}}|$ and $|vk_{\mathsf{asig}}|$ denote the size of a signature and a verification key of the underlying scheme ASIG, respectively.

[Scheme $\mathsf{ASIG2}$]

$\mathsf{Setup}(1^\lambda)$**:** Same as $\mathsf{ASIG.Setup}$. It takes security parameter 1^λ and outputs a parameters gk.

$\mathsf{Key}(gk)$**:** Same as $\mathsf{ASIG.Key}$. It takes gk and outputs verification key $vk \in \mathbb{H}$ and secret key sk.

$\mathsf{Sign}(sk, msg)$**:** On input message $msg = \{M_1, \dots, M_n\} \in \mathbb{H}^n$, do the following.

- $(vk_0, sk_0) \leftarrow \mathsf{ASIG.Key}(gk)$, $\delta_0 \leftarrow \mathsf{ASIG.Sign}_{sk}(vk_0)$, $\xi_0 \leftarrow \mathsf{ASIG.Sign}_{sk_0}(\langle n \rangle)$;
- for $i = 1, \dots, n$: $(vk_i, sk_i) \leftarrow \mathsf{ASIG.Key}(gk)$, $\delta_i \leftarrow \mathsf{ASIG.Sign}_{sk_0}(vk_i)$, $\xi_i \leftarrow \mathsf{ASIG.Sign}_{sk_0}(vk_i \cdot \langle i \rangle)$, $\gamma_i \leftarrow \mathsf{ASIG.Sign}_{sk_i}(M_i)$.

Output $\sigma = (vk_i, \delta_i, \xi_i, \gamma_i)_{i=0}^n$ (γ_0 is empty for notational convenience in the whole section.)

$\mathsf{Vrf}(vk, msg, \sigma)$**:** On input (msg, σ), parse msg as $\{M_1, \dots, M_n\} \in \mathbb{H}^n$ and σ as $(vk_i, \delta_i, \xi_i, \gamma_i)_{i=0}^n$. Check if $1 = \mathsf{ASIG.Vrf}_{vk}(vk_0, \delta_0) = \mathsf{ASIG.Vrf}_{vk_0}(\langle n \rangle, \xi_0)$ holds. For $i = 1, \dots, n$, check if $1 = \mathsf{ASIG.Vrf}_{vk_i}(M_i, \gamma_i) = \mathsf{ASIG.Vrf}_{vk_0}(vk_i, \delta_i)$ $= \mathsf{ASIG.Vrf}_{vk_0}(vk_i \cdot \langle i \rangle, \xi_i)$. Output 1 if all verifications succeeded. Output 0 otherwise.

¶

Theorem 11. *If ASIG is EUF-CMA, then so is $\mathsf{ASIG2}$.*

Proof. We follow the game transformation style for proving the theorem. Starting from the chosen-message attack against $\mathsf{ASIG2}$, we change the game slightly and eventually see that the adversary can never win the game.

Game 0. (Standard EUF-CMA game.) An adversary is given verification key vk and whenever it makes signing query for msg, signing oracle $\mathcal{O}_{\mathsf{sign}}$ returns $\sigma \leftarrow \mathsf{ASIG2.Sign}(sk, msg)$. The adversary eventually outputs a forgery, $\sigma^\dagger = (vk_i^\dagger, \delta_i^\dagger, \xi_i^\dagger, \gamma_i^\dagger)_{i=0}^{n^\dagger}$ and $msg^\dagger = (M_1^\dagger, \dots, M_{n^\dagger}^\dagger)$.

Let us define some notations. Let $\Pr[i]$ denote the probability that the adversary outputs a valid forgery in Game i. In particular, $\Pr[0] = \epsilon$ is the probability of breaking $\mathsf{ASIG2}$.

Let $Q_{m,\sigma} = \{(msg, \sigma)\}$ be the transcript exchanged between the adversary and $\mathcal{O}_{\mathsf{sign}}$. Let q_s and n_{max} denote the number of signing queries and the maximum message length, which are bounded by polynomials in λ. For a public key vk with corresponding sk, let $\mathcal{I}_{vk}$ denote messages given as input to $\mathsf{ASIG.Sign}_{sk}(\cdot)$ while $\mathcal{O}_{\mathsf{sign}}$ is computing signatures. If vk is not used by $\mathcal{O}_{\mathsf{sign}}$, then $\mathcal{I}_{vk} = \emptyset$.

Let ϵ_{vk} be the highest probability that a key is generated by $\mathsf{ASIG.Key}$. We assume that ϵ_{vk} is negligible in the security parameter, which is the case for the scheme in Sect. 6.1, where $\epsilon_{vk} = 1/p$.

Game 1. Abort the game if there is a public key vk_i that appears more than once in $Q_{m,\sigma}$.

As each signature in $Q_{m,\sigma}$ includes at most $n_{max} + 1$ randomly chosen public keys, the probability of aborting is bounded by $\left(q_s(n_{max}+1)\right)^2 \cdot \epsilon_{vk}$. We thus have $|\Pr[0] - \Pr[1]| \leq \left(q_s(n_{max} + 1)\right)^2 \cdot \epsilon_{vk}$, which is negligible in the security parameter.

Game 2. Abort the game if there is a signature $(vk_i, \delta_i, \xi_i, \gamma_i)_{i=0}^n$ in $Q_{m,\sigma}$ such that $vk_i = vk_j \cdot \langle j \rangle$ holds for some $0 < i, j \le n$.

Observe that $vk_i = vk_i \cdot \langle i \rangle$ does not happen since $\langle i \rangle \ne 1$. For every fixed i, the probability that there exists j for $vk_i = vk_j \cdot \langle j \rangle$ is bounded by $(n_{max} - 1) \cdot \epsilon_{vk}$. As there are at most q_s signatures, the probability of aborting is upper bounded by $q_s(n_{max} - 1) \cdot \epsilon_{vk}$. We thus have $|\Pr[1] - \Pr[2]| < q_s(n_{max} - 1) \cdot \epsilon_{vk}$, which is negligible.

Game 3. Abort the game if there is a signature $(vk_i, \delta_i, \xi_i, \gamma_i)_{i=0}^n$ in $Q_{m,\sigma}$ such that $vk_i = \langle n \rangle$ holds for some $0 < i \le n$.

For each signature, this happens with probability $n\,\epsilon_{vk}$ due to the randomness of vk_j. Accumulating to q_s signatures with maximal length of messages, the probability of aborting for this case is upper bounded by $q_s \cdot n_{max} \cdot \epsilon_{vk}$. Thus, $|\Pr[2] - \Pr[3]| < q_s \cdot n_{max} \cdot \epsilon_{vk}$, which is negligible.

Game 4. Abort the game if there is a signature $(vk_i, \delta_i, \xi_i, \gamma_i)_{i=0}^n$ in $Q_{m,\sigma}$ such that $\langle n' \rangle \in \mathcal{I}_{vk_0}$ holds for some $n' < n$.

The case that $\langle n' \rangle \in \{vk_1, \dots, vk_n, vk_1 \cdot \langle 1 \rangle, \dots, vk_n \cdot \langle n \rangle\}$ happens with probability at most $2n_{max} \cdot \epsilon_{vk}$ for each signature. Thus, $|\Pr[3] - \Pr[4]| < 2q_s \cdot n_{max} \cdot \epsilon_{vk}$, which is negligible.

Game 5. Abort if the output of the adversary satisfies $vk_0^\dagger \notin \mathcal{I}_{vk}$.

This case breaks EUF-CMA of ASIG since $vk_0^\dagger$ is a fresh message correctly signed with respect to vk. We thus have $|\Pr[4] - \Pr[5]| < \epsilon_{euf}$, which is negligible as we assume ASIG is unforgeable.

Now if Game 5 is completed, there exists $msg^\star = (M_1^\star, \dots, M_{n^\star}^\star)$ and unique $\sigma^\star = (vk_i^\star, \delta_i^\star, \xi_i^\star, \gamma_i^\star)_{i=0}^{n^\star}$ in $Q_{m,\sigma}$ such that $vk_0^\star = vk_0^\dagger$.

Game 6. Abort if $\{vk_1^\dagger, \dots, vk_{n^\dagger}^\dagger\} \not\subseteq \{vk_1^\star, \dots, vk_{n^\star}^\star\}$.

Suppose that there exists $vk_i^\dagger$ with $vk_i^\dagger \notin \{vk_1^\star, \dots, vk_{n^\star}^\star\}$. Observe that $vk_i^\dagger \notin \{\langle n^\star \rangle, vk_1^\star \cdot \langle 1 \rangle, \dots, vk_{n^\star}^\star \cdot \langle n^\star \rangle\}$ holds since the game did not meet the abort conditions in Game 2 and 3. Then, $vk_i^\dagger$ is a new message signed with $vk_0^\star$ $(= vk_0^\dagger)$ since $vk_0^\star$ is used only for messages $\{vk_1^\star, \dots, vk_{n^\star}^\star\} \cup \{\langle n^\star \rangle, vk_1^\star \cdot \langle 1 \rangle, \dots, vk_{n^\star}^\star \cdot \langle n^\star \rangle\}$. Accordingly, the abort condition in Game 5 is met only if ASIG2 is broken with respect to key $vk_0^\star$. Taking the probability loss for guessing the target key, the probability of abort in this case is upper bounded by $q_s\epsilon_{euf}$. We thus have $|\Pr[5] - \Pr[6]| < q_s\epsilon_{euf}$ which is negligible.

At this point, we observe that for all $0 < i \le n^\dagger : vk_i^\dagger = vk_i^\star$. Suppose, on the contrary, that $vk_i^\dagger = vk_j^\star$ happens for some $0 < i \le n^\dagger, i \ne j$, and $0 < j \le n^\star$. We then have $vk_j^\star \cdot \langle i \rangle \in \mathcal{I}_{vk_0^\star}$, which would have caused an abort in Game 2. Furthermore, it holds that $n^\dagger = n^\star$; otherwise, the game would have aborted as in Game 4. Accordingly, we have $(vk_1^\dagger, \dots, vk_{n^\dagger}^\dagger) = (vk_1^\star, \dots, vk_{n^\star}^\star)$.

Game 7. Abort if $\exists i$ s.t. $M_i^\dagger \ne M_i^\star$.

Since $vk_i^\star$ is used only for signing $M_i^\star$, this case breaks EUF-CMA of ASIG w.r.t. key $vk_i^\star$. Taking the probability loss of guessing the target key into account, we have $|\Pr[6] - \Pr[7]| < q_s n_{max} \epsilon_{euf}$.

In Game 7, $msg^{\dagger} = msg^{\star}$ holds and the output cannot be a valid forgery. Accumulating the above bounds, we have the probability of successful forgery is negligible if ASIG is EUF-CMA. □

7. Applications

This section presents two applications that highlight the benefits of structure-preserving signatures. The first one, group signatures with concurrent join, features generic use of structure-preserving signature schemes combined with non-interactive proofs. The second one, round-optimal blind signatures, uses the automorphic signature scheme from Sect. 6 in a specific manner to gain efficiency.

7.1. *Group Signatures with Concurrent Join*

A group signature, GSIG, consists of 6 algorithms Setup, Join, Sign, Vrf, Open and Judge such that:

- Setup is an algorithm run in a trusted manner. It takes a security parameter and generates a group verification key vk_{g}, a certification key sk_{c} and an opening key sk_{o}. The group verification key is published, the certification key is privately given to an authority called the issuer, whereas the opening key is given privately to another authority called the opener.
- Join is a pair of interactive algorithms run by the issuer and a user who requests membership of the group. When the protocol is completed, the member obtains a secret user key, *usk*, and the issuer obtains a public user key, *upk*, that consists of some parts of the interaction. The public user key and the identity of the member is stored to the registration record, *reg*.
- Sign is a signing algorithm run by a group member. On input a secret user key *usk* and a message *msg*, it outputs a group signature π of the message.
- Vrf is a verification algorithm that on input the group verification key vk_{g}, a message *msg*, and a signature π, outputs 1 or 0 meaning acceptance or rejection, respectively.
- Open is an opening algorithm run by the opener. It takes the opening key sk_{o}, a signature π and *reg* as input and outputs opening information *open* and identity id of a group member in *reg*.
- Judge is an algorithm that takes a group signature and an output of Open as input and outputs 1 or 0 that indicate acceptance and rejection, respectively.

Correctness requires that Vrf outputs 1 for all legitimately generated keys and signatures for any message, and Judge outputs 1 for all outputs of Open with legitimate input if $id \neq 0$ and outputs 0 for $id = 0$.

There are several variations and extensions of group signature schemes and their security notions. We focus on anonymity, traceability and non-frameability as essential security notions and follow the definitions in [55].

- Anonymity is defined by a game where an adversary is given a group signature created by one of two honest and correctly registered members and must distinguish which member created the signature. The issuing key and all secret user keys can be exposed to the adversary.

Definition 6. (*Anonymity*) A group signature scheme is anonymous if for any probabilistic polynomial-time adversary $\mathcal{A}$

$$\Pr\left[\begin{array}{l}(vk_{\mathrm{g}}, sk_{\mathrm{c}}, sk_{\mathrm{o}}) \leftarrow \mathsf{Setup}(1^{\lambda});\\ b \leftarrow \{0,1\}\\ \tilde{b} \leftarrow \mathcal{A}^{\mathcal{O}(b)}(vk_{\mathrm{g}}, sk_{\mathrm{c}})\end{array} : b = \tilde{b}\right] - \frac{1}{2}$$

is negligible where oracle $\mathcal{O}$ works as follows.

- On receiving a request, it executes the protocol Join as the issuer and an honest user. The view of the honest user is returned to $\mathcal{A}$.
- Given *upk* for a new corrupt user, it stores it to *reg*.
- Given a message and two distinct identities, id_0 and id_1, of honest registered members, it returns a signature created by id_b.

The above notion is referred to as CPA-anonymity. When the oracle provides the additional functionality that, on receiving a valid signature and message, it returns the result of Open on the input, we call the strengthened notion CCA-anonymity [23].

- Traceability is defined by a game where an adversary corrupting the opener and arbitrary members attempts to create a signature that verifies correctly but yields invalid opening information that does not identify any registered member.

Definition 7. (*Traceability*) A group signature scheme is traceable if for any probabilistic polynomial-time adversary $\mathcal{A}$

$$\Pr\left[\begin{array}{l}(vk_{\mathrm{g}}, sk_{\mathrm{c}}, sk_{\mathrm{o}}) \leftarrow \mathsf{Setup}(1^{\lambda});\\ (msg, \pi) \leftarrow \mathcal{A}^{\mathcal{O}}(vk_{\mathrm{g}}, sk_{\mathrm{o}})\\ (id, open) \leftarrow \mathsf{Open}(sk_{\mathrm{o}}, msg, \pi)\end{array} : \begin{array}{l}1 = \mathsf{Vrf}(vk_{\mathrm{g}}, m, \pi) \ \wedge\\ 0 = \mathsf{Judge}(vk_{\mathrm{g}}, \pi, msg, id, open, reg)\end{array}\right]$$

is negligible, where oracle $\mathcal{O}$ plays the role of the honest issuer in the protocol Join.

- Non-frameability is defined by a game where an adversary corrupting the opener attempts to create a group signature on a message together with an opening of it that identifies a group member who never signed the message. The issuer is supposed to be honest in the sense that it honestly certifies association between uncorrupted members' identities and their own public key. This is formalized by not allowing the adversary to write to *reg*.

Definition 8. (*Non-Frameability*) A group signature scheme is non-frameable if for any probabilistic polynomial-time adversary $\mathcal{A}$

$$\Pr\left[\begin{array}{l}(vk_{\mathrm{g}}, sk_{\mathrm{c}}, sk_{\mathrm{o}}) \leftarrow \mathsf{Setup}(1^{\lambda});\\ (m, \pi, id, open) \leftarrow \mathcal{A}^{\mathcal{O}}(sk_{\mathrm{c}}, sk_{\mathrm{o}})\end{array} : \begin{array}{l}1 = \mathsf{Vrf}(vk_{\mathrm{g}}, m, \pi) \ \wedge\\ 1 = \mathsf{Judge}(vk_{\mathrm{g}}, m, \pi, id, open, reg) \wedge\\ id \in \mathcal{H} \wedge m \notin \mathcal{L}_{id}\end{array}\right]$$

is negligible where oracle $\mathcal{O}$ works as follows.

- On receiving a request, it invokes Join as a member with a new identity, id, and interacts with $\mathcal{A}$ playing the corrupt issuer. When the member-side protocol is successfully completed with usk obtained, the oracle records (id, upk) to reg and (id, usk) to $\mathcal{H}$, the list of honest users. (Note that reg is not available to $\mathcal{A}$; also note that we assume that upk is obtained by the successful interaction.)
- On receiving $id \in \mathcal{H}$, it returns corresponding usk and removes the record from $\mathcal{H}$.
- On receiving (id, msg), if $(id, usk) \in \mathcal{H}$ for some usk, it returns a signature on msg by using usk and records m to $\mathcal{L}_{id}$.

(The definition in [19] is slightly stronger in the sense that an adversary attempts to create a signature that, on opening, identifies a group member who actually did not create the signature, though the member may have signed the same message before. The stronger notion can be captured by modifying the definition to $(msg, \pi) \notin \mathcal{L}_{id}$.)

A technical difficulty arises when the group manager, Alice, issues a certificate to a group member, Bob. Alice signs Bob's public key but for the sake of provable security she should be convinced of the independence of Bob's key from anyone else's. The independence is assured by implementing the protocol in a way that Bob can retrieve the certificate only when he knows some critical information. Thus, the protocol often uses zero-knowledge proofs of knowledge, which lose efficiency when composed concurrently. The issue of efficient concurrent join has been addressed in the literature. A single-round certification protocol that allows concurrent execution is sketched in [32]. In their framework, the join protocol is as simple as letting user Bob obtaining a certificate from Alice on Bob's public key. When Bob signs a document, he first signs the document using his own key and then proves that the signature is correct with respect to a key certified by Alice. The proof has to be zero knowledge (thus hiding the public key and the certificate) and non-interactively so that it can be included in the signature. Such a proof of knowledge, however, is not easy to instantiate efficiently due to high complexity of the relation to prove. The first efficient instantiation in the random-oracle model is presented in [68]. Using structure-preserving signatures as Alice's certificates and applying the Groth–Sahai proof system for proving Bob's knowledge of a correct certificate, this framework can be efficiently instantiated without random oracles. In the following, we refine the framework by incorporating ideas from [55].

Let $\mathsf{SIGauth}$ and SIGmem be signature schemes, and let NIWI be a witness-indistinguishable proof-of-knowledge system that allows one to extract a witness by using a trapdoor.

[Scheme GSIG]

Setup(1^λ)**:** Run $(vk_\mathrm{c}, sk_\mathrm{c}) \leftarrow \mathsf{SIGauth.Key}(1^\lambda)$, and set up a CRS Σ_{niwi} and an extraction trapdoor sk_o for NIWI. Then output group verification key $vk_\mathrm{g} := (vk_\mathrm{c}, \Sigma_{\mathsf{niwi}})$, certification key sk_c and opening key sk_o.

Join: (Member's side) Generate a key pair $(vk_\mathrm{u}, sk_\mathrm{u}) \leftarrow \mathsf{SIGmem.Key}(1^\lambda)$ and send vk_u to the issuer. After receiving a certificate σ_c from the issuer, output $usk := (vk_\mathrm{u}, sk_\mathrm{u}, \sigma_\mathrm{c})$.

(Issuer's side) On receiving vk_u, run $\sigma_\mathrm{c} \leftarrow \mathsf{SIGauth.Sign}(sk_\mathrm{c}, vk_\mathrm{u})$ and store

$upk := (vk_u, \sigma_c)$ and the identity of the member to *reg*. Then send σ_c to the member.

Sign(usk, msg)**:** Parse $usk = (vk_u, sk_u, \sigma_c)$. Run $\sigma_u \leftarrow \mathsf{SIGmem.Sign}(sk_u, msg)$, and generate a NIWI proof π of knowledge of $(vk_u, \sigma_c, \sigma_u)$ such that

$$1 = \mathsf{SIGauth.Vrf}(vk_c, \underline{vk_u}, \underline{\sigma_c}) \quad \wedge \quad 1 = \mathsf{SIGmem.Vrf}(\underline{vk_u}, msg, \underline{\sigma_u}) \qquad (50)$$

Then output π as a group signature.

Vrf(vk_g, msg, π)**:** Parse $vk_g = (vk_c, \Sigma_{\mathsf{niwi}})$. Verify π as a NIWI proof for relation (50).

Open(sk_o, π, reg)**:** Run the knowledge extractor of the NIWI proof system on π to obtain witness $open := (vk_u, \sigma_c, \sigma_u)$. If $(id, (vk_u, \sigma_c))$ for some id is in *reg*, output $(id, open)$. Output $(0, \emptyset)$, otherwise.

Judge$(vk_g, \pi, msg, id, open, reg)$**:** Parse *open* into $(vk_u, \sigma_c, \sigma_u)$. Output 1 if $1 = \mathsf{Vrf}(vk_g, msg, \pi)$, both equations in (50) hold, and $(id, (vk_u, \sigma_c))$ is in *reg*. Output 0, otherwise.

¶

Theorem 12. *Group signature scheme* GSIG *is CPA-anonymous, traceable and non-frameable if* SIGauth *and* SIGmem *are EUF-CMA and* NIWI *is witness-indistinguishable and knowledge sound. Furthermore,* GSIG *allows to run the* GSIG.Join *protocol concurrently.*

Proof. CPA-anonymity follows directly from the (computational) WI property [59] of the proof system NIWI. For traceability, suppose that there is a valid signature π on message *msg*. Due to knowledge soundness of NIWI, the opener can extract $(vk_u, \sigma_c, \sigma_u)$ from π, which satisfy $1 = \mathsf{SIGauth.Vrf}(vk_c, vk_u, \sigma_c)$. If vk_u does not point to any group member registered through GSIG.Join, σ_c is a valid forgery for SIGauth, which contradicts EUF-CMA of SIGauth. In more detail, this is shown by constructing a reduction to EUF-CMA that simulates the joining protocol by using the signing oracle of SIGauth. Clearly, this is possible even if GSIG.Join is executed concurrently with several members since the protocol consists of only one round of message flow between the issuer and the member. Thus, vk_u allows tracing with concurrent execution of the join protocol. For non-frameability, suppose that the opener extracts $(vk_u, \sigma_c, \sigma_u)$ from a group signature on message *msg*. If $1 = \mathsf{SIGmem.Vrf}(vk_u, msg, \sigma_u)$ holds, but the owner of vk_u has never signed *msg*, it is a valid forgery for SIGmem, contradicting EUF-CMA of SIGmem. □

By instantiating SIGauth with our structure-preserving signature scheme CSIG (given on page 30) and NIWI with the GS proof system, we can instantiate the above generic construction efficiently. Furthermore, using CSIG for SIGauth allows the inclusion of a human-readable "warrant" in σ_c so that the signing policy given to a group member is explicit. A warrant is a written policy that defines validity of signatures, e.g., specific period of time in which the signature is valid. Since CSIG has constant-size signatures, this extension can be done without impacting the size of the group signature (except for the warrant itself) at all.

Table 2. Summary of efficiency and properties of group signature schemes with CPA-anonymity.

Scheme	Concurrent Join	Non-Frameability	Signature Size	Assumptions
BW07 [29]	Yes	No	$6^{[N]}$	SD, HSDH
Gro07 [55]	No	Yes	$30^{[1]}$	SDH, q-U, DLIN
GSIG ([38]+BB [22])	Yes	Yes	$231^{[1]} + 1^{[p]}$	DLIN,SDH,HSDH
GSIG (CSIG+BB [22])	Yes	Yes	$40^{[1]} + 1^{[p]}$	SFP, SDH

The signature size counts the number of elements and indicating the groups they belong to ([1], $[N]$ and $[p]$, respectively, for $\mathbb{G}$, $\mathbb{Z}_N$ and $\mathbb{Z}_p$). SD: subgroup decision assumption [27]. q-U: See [55]

We assess the efficiency in the setting $\Lambda = \Lambda_{\mathsf{sym}}$ as follows. Let SIGmem be a signature scheme whose verification key vk_{u} and signature σ_{u} consist of α and β group elements, respectively. Let γ be the number of group elements needed to prove relation $1 = \mathsf{SIGmem.Vrf}(vk_{\mathrm{u}}, msg, \sigma_{\mathrm{u}})$ including GS commitments for vk_{u} and σ_{u}. Regardless of the size of vk_{u} to be certified, our $\mathsf{SIGauth}$ outputs σ_{c} of size 7. Since 4 out of the 7 elements in σ_{c} can be perfectly randomized and given in the clear (as observed in Sect. 5.1), we need only 3 GS commitments to prove relation $1 = \mathsf{SIGauth.Vrf}(vk_{\mathrm{c}}, vk_{\mathrm{u}}, \sigma_{\mathrm{c}})$, which consists of two one-sided pairing-product equations and costs 6 elements in a proof. (Commitments of vk_{u} are already included in γ.) In total we have (Group Sig Size) $= 19 + \gamma$. For comparison, we propose to instantiate $\mathsf{SIGauth}$ by using a signature scheme in [38], which has $9\alpha + 4$ elements in σ_{c} and $3\alpha + 3$ one-sided and 3α double-sided pairing-product equations in $\mathsf{SIGauth.Vrf}$. In that case, the size of a group signature is (Group Sig Size) $= 63\alpha + 21 + \gamma$.

If we instantiate SIGmem with the fully EUF-CMA Boneh-Boyen signature scheme [22], vk_{u} consists of $\alpha = 3$ group elements (including the bases). A signature consists of one group element and one scalar value, but the scalar value is totally random and independent of the verification key. So we have $3 + 1$ GS commitments in proving $1 = \mathsf{SIGmem.Vrf}(vk_{\mathrm{u}}, msg, \sigma_{\mathrm{u}})$. The verification predicate consists of a double-sided pairing-product equation, which yields 9 group elements in a proof. In total, we have $\gamma = 21$ and a group signature consists of 40 group elements and 1 scalar value.

Table 2 summarizes some efficient group signature schemes that provide CPA-anonymity in the standard model under non-interactive assumptions. Our construction GSIG(CSIG+BB[22]) yields a signature that contains 10 more group elements than that of [55]. This is the price for achieving the concurrent join property and allowing a simple and modular security argument without dedicated assumptions. For comparison, we also consider the case where a signature scheme in [38] is used as $\mathsf{SIGauth}$. With the same setting ($\alpha = 3$, $\gamma = 21$), the signature size results in 231 ($= 63 \times 3 + 21 + 21$) group elements and 1 scalar value.

Finally, we remark that CCA-anonymity is obtained by following the approach in [55], which uses a strong one-time signature scheme and a selective-tag CCA-secure tag-based public key encryption scheme. A decryption key of the tag-based encryption is given to the opener, and the public key is published. When a member issues a signature, vk_{u} is encrypted by the tag-based encryption under a one-time verification key as a tag. Then, a NIZK is generated to prove that the ciphertext decrypts to the same value, i.e., vk_{u}, committed for NIWI proof π. The one-time key is used to sign π, the ciphertext and the NIZK proof. By using the same instantiation as in [55], this strengthening costs 15 extra group elements in a signature. (Committing to 3 random coins used in tag-based encryption costs 3×3 elements, and proving 3 relations in multi-exponentiations costs 3×2 elements.) Accordingly, we have a CCA-anonymous group signature scheme with concurrent join whose signature consists of $55 (= 40 + 15)$ group elements and one scalar value.

7.2. *Round-Optimal Blind Signatures*

A blind-signature scheme BS allows a user to obtain signatures on messages which remain hidden from the signer. It is defined by five algorithms $\mathsf{BS} = \mathsf{BS}.\{\mathsf{Setup}, \mathsf{Key}, \mathsf{User}, \mathsf{Signer}, \mathsf{Vrf}\}$, where Setup, on input 1^λ, generates the common parameters gk; Key, on input gk, generates a key pair (vk, sk); User is a signature-request algorithm, taking input gk, vk and a message, which interacts with the signer-side algorithm Signer, having input sk, as a signature-generation protocol; $\mathsf{BS}.\mathsf{Vrf}$ that verifies a signature on a message w.r.t. gk and vk. The notion of completeness is defined as for ordinary digital signature schemes; that is, when $gk \leftarrow \mathsf{Setup}(1^\lambda)$ and $(vk, sk) \leftarrow \mathsf{Key}(gk)$ then for every message msg and every σ obtained by running $\mathsf{User}(gk, vk, msg) \leftrightarrow \mathsf{Signer}(sk)$ we have: $1 \leftarrow \mathsf{Vrf}(gk, vk, msg, \sigma)$.

A blind-signature scheme BS is *round-optimal* if in the signature-generation protocol there is only one round of communication: User sends a message to Signer and the latter sends one message back; this means that Signer is a (stateless) algorithm that takes a secret key sk and a message msg_{U} from User and outputs a message msg_{S} that is given to User, which outputs a signature σ.

The two standard security notions are unforgeability and blindness as defined in, e.g., [44,61,84]. Blindness means that no adversarial signer can associate a valid signature to an execution of the signature-generation protocol. Unforgeability states that no adversarial user can obtain more valid signatures than the number of completed signature-generation protocol executions.

Definition 9. (*Blindness*) A (round-optimal) blind-signature scheme satisfies *blindness* if for any probabilistic polynomial-time adversary $\mathcal{A}$

$$\Pr\left[\begin{array}{l} b \leftarrow \{0,1\};\ gk \leftarrow \mathsf{Setup}(1^\lambda);\ (vk, m_0, m_1, st_{\mathcal{A}}) \leftarrow \mathcal{A}(gk) \\ \text{For } i = b, 1-b \text{ do} \\ \quad (msg_{\mathsf{U}}, st_{\mathsf{U}}) \leftarrow \mathsf{User}(gk, vk, m_i) \\ \quad (msg_{\mathcal{A}}, st_{\mathcal{A}}) \leftarrow \mathcal{A}(st_{\mathcal{A}}, msg_{\mathsf{U}});\ \sigma_i \leftarrow \mathsf{User}(st_{\mathsf{U}}, msg_{\mathcal{A}}) \\ \text{If } \sigma_0 = \perp \text{ or } \sigma_1 = \perp \text{ then } (\sigma_0, \sigma_1) := (\perp, \perp) \\ b^\star \leftarrow \mathcal{A}(st_{\mathcal{A}}, (\sigma_0, \sigma_1)) \end{array} : b^\star = b \right] - \frac{1}{2}$$

is negligible.

Definition 10. (*Blind-Signature Unforgeability*) A (round-optimal) blind-signature scheme is *unforgeable* if for any probabilistic polynomial-time adversary $\mathcal{A}$

$$\Pr\left[\begin{array}{ll} gk \leftarrow \mathsf{Setup}(1^\lambda);\ (vk, sk) \leftarrow \mathsf{Key}(gk) & m^\star \neq m_j^\star \text{ for all } i \neq j \\ ((m_1^\star, \sigma_1^\star), \ldots, (m_q^\star, \sigma_q^\star)) & : 1 \leftarrow \mathsf{Vrf}(vk, m_i^\star, \sigma_i^\star) \\ \qquad \leftarrow \mathcal{A}^{\mathsf{Signer}(sk,\cdot)^{(q-1)}}(vk) & \quad \text{for } 1 \leq i \leq q \end{array}\right]$$

is negligible, where the adversary is allowed $q - 1$ oracles calls to Signer.

Fischlin [43] gives a generic construction for concurrently executable blind-signature schemes with optimal round complexity in the common reference string (CRS) model. The construction relies on commitment, encryption and signature schemes and generic NIZK proofs for NP-languages. In the signature-issuing protocol, the user first sends a commitment to the message to the signer, who responds with a signature on the commitment. The user then constructs the blind signature as follows: She encrypts the commitment and the signature and adds an NIZK proof that the signature is valid on the commitment and that the committed value is the message.

Following [61], Abe and Ohkubo [10] replace the NIZK proof in Fischlin's construction by a witness-indistinguishable proof and concretely suggest Groth–Sahai (GS) proofs. (Note that GS commitments on group elements can be "decrypted" using the extraction key.) To be compatible, the signature scheme must have messages and signatures consisting of group elements and verification must amount to evaluating pairing-product equations. However, they only mention the highly inefficient scheme from [54] as a feasibility result and leave open the problem of an efficient construction. Structure-preserving signatures satisfy all the compatibility requirements and enable thus an efficient instantiation of round-optimal blind signatures; it suffices to construct a commitment scheme such that commitments lie in the message space of the signature and correct opening is verifiable by pairing-product equations.

We directly construct a scheme based on ASIG (Sect. 6.1) which has smaller blind signatures than an instantiation of the generic construction. This is because in the end of our issuing protocol the user holds a signature on the actual message rather than on a commitment to it. To make this possible, the user sends a *randomization* of the message to the signer in addition to the commitment. From this, the signer makes a "pre-signature" and sends it to the user, who, knowing the randomizer, can turn it into an actual signature on the message. The blind signature is then a GS proof of knowledge of a signature on the message (rather than a commitment), which avoids a proof that the commitment opens to the message.

In more details, given parameters (Λ, F, K, T) a user who wants to obtain a blind signature on $(M, \tilde{N})$ chooses a random $\rho \leftarrow \mathbb{Z}_p$, and *blinds* M by the factor T^ρ. In addition to $U := T^\rho \cdot M$, she sends a GS proof of knowledge of $(M, \tilde{N})$ and $(G^\rho, \tilde{G}^\rho)$. The signer now formally produces a "signature" 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 $(M, \tilde{N})$ 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.

Note that it is the particular form of signatures of ASIG that allow turning a signature on a "half-message" U into a signature on any $M = T^{-\rho}$, if one knows ρ. This does not contradict unforgeability, since the user does not obtain a *signature* on U (unless $U = M$), since U is not an element of the message space $\mathcal{DH}$. And to produce $(U, \tilde{G}^{\log_G U}) \in \mathcal{DH}$, the user would have to break AWF-CDH.

[Scheme BS]

$\mathsf{Setup}(1^\lambda)$: Run $gk_{\mathsf{ASIG}} := (\Lambda, F, K, T) \leftarrow \mathsf{ASIG.Setup}(1^\lambda)$ and generate a common reference string Σ_{GS} for the GS proof system based on Λ. Output $gk := (gk_{\mathsf{ASIG}}, \Sigma_{\mathsf{GS}})$. The message space is defined as $\mathcal{DH} := \{(G^m, \tilde{G}^m) \mid m \in \mathbb{Z}_p\}$.

$\mathsf{Key}(gk)$: Parse gk as $(gk_{\mathsf{ASIG}}, \Sigma_{\mathsf{GS}})$, run $\mathsf{ASIG.Key}(gk_{\mathsf{ASIG}})$ to obtain $vk = (X, \tilde{Y})$ and $sk' = (gk_{\mathsf{ASIG}}, vk, x)$. Output $(vk, sk := (gk, vk, x))$.

$\mathsf{Signer}(sk) \leftrightarrow \mathsf{User}(gk, vk, (M, \tilde{N}))$:

User: Choose $\rho \leftarrow \mathbb{Z}_p$. Compute $P := G^\rho$, $\tilde{Q} := \tilde{G}^\rho$ and $U := T^\rho \cdot M$ and a GS proof ϕ of knowledge of $(M, \tilde{N}, P, \tilde{Q})$ such that

$$e(\underline{M}, \tilde{G}) = e(G, \underline{\tilde{N}}) \quad \wedge \quad e(\underline{P}, \tilde{G}) = e(G, \underline{\tilde{Q}}) \\ \wedge \quad e(T, \underline{\tilde{Q}})\, e(\underline{M}, \tilde{G}) = e(U, \tilde{G}) \tag{51}$$

Send (U, ϕ) to Signer.

Signer: Verify (U, ϕ) with respect to relation (51). If it is valid, run $(A, B, \tilde{D}, R', \tilde{S}') \leftarrow \mathsf{ASIG.Sign}(sk, (U, -))$, where U is taken as (part of) a message without internally checking its validity. Send $(A, B, \tilde{D}, R', \tilde{S}')$ to User.

User: On receiving $(A, B, \tilde{D}, R', \tilde{S}')$, compute $R := R' \cdot P$ and $\tilde{S} := \tilde{S}' \cdot \tilde{Q}$. Compute a GS NIWI proof of knowledge π of $(A, B, \tilde{D}, R, \tilde{S})$ which satisfies $1 = \mathsf{ASIG.Vrf}\big(vk, (M, \tilde{N}), (A, B, \tilde{D}, R, \tilde{S})\big)$, that is

$$e(\underline{A}, \tilde{Y} \cdot \underline{\tilde{D}}) = e(K \cdot M, \tilde{G})\, e(T, \underline{\tilde{S}}) \quad \wedge \quad e(\underline{B}, \tilde{G}) = e(F, \underline{\tilde{D}}) \\ \wedge \quad e(\underline{R}, \tilde{G}) = e(G, \underline{\tilde{S}}) \tag{52}$$

(If $(A, B, \tilde{D}, R, \tilde{S})$ does not satisfy (52) then output $\perp$.) Output π as a signature for message $(M, \tilde{N})$.

$\mathsf{Vrf}(\Sigma, vk, (M, \tilde{N}), \pi)$: Accept if $(M, \tilde{N}) \in \mathcal{DH}$ and π is a valid proof for (52).

¶

The protocol is correct: The signer sends $A = (K \cdot T^r \cdot U)^{\frac{1}{x+c}} = (K \cdot T^{r+\rho} \cdot M)^{\frac{1}{x+c}}$, $B = F^c$, $\tilde{D} = \tilde{G}^c$, $R' = G^r$, $\tilde{S}' = \tilde{G}^r$ and the user sets $R := R' \cdot P = G^{r+\rho}$ and $\tilde{S} := \tilde{S}' \cdot \tilde{Q} = \tilde{G}^{r+\rho}$, which makes it a valid signature on $(M, \tilde{N})$ with randomness $r + \rho$.

The round complexity of the scheme is optimal [43]. The user message is in $\mathbb{G}^{17} \times \tilde{\mathbb{G}}^{16}$, the signer message in $\mathbb{G}^3 \times \tilde{\mathbb{G}}^2$ and a blind signature is in $\mathbb{G}^{18} \times \tilde{\mathbb{G}}^{16}$. Note that the scheme remains automorphic, since commitments and proofs are composed of group elements that are verified by checking pairing-product equations and the verification keys and messages are those from ASIG. It is the automorphic property of the scheme that allowed its use to implement (blinded) certification chains in the construction of non-interactively delegatable anonymous credentials in [46].

Theorem 13. *Blind-signature scheme* BS *is unforgeable and blind if ADH-SDH and SXDH hold for* $\mathcal{G}$.

Proof. **Blindness.** In the blindness game, we choose a random $b \leftarrow \{0, 1\}$, which the adversary must guess. After setting up gk we are given vk, m_0 and m_1 by the adversary and run User for m_b and for m_{1-b}. If both executions succeed, yielding π_b and π_{1-b}, the adversary is given (π_0, π_1) and must guess b.

We modify the game in Definition 9 by replacing Σ_{GS} by a simulated CRS, which leads to perfectly WI commitments and proofs. This modification is indistinguishable by SXDH. We claim that in this modified game the adversary's probability of guessing b is exactly $\frac{1}{2}$, because neither what is sent by User nor the final signatures contain any information about b.

First consider (U, ϕ) given to the adversary during the signature-generation protocol. To see that $(M, \tilde{N})$ is perfectly hidden, observe that for every possible $M \in \mathbb{G}$, there exists a (unique) value $\rho \in \mathbb{Z}_p$ that satisfies $U = T^{\rho} \cdot M$. In other words, for every $(M, \tilde{N}) \in \mathcal{DH}$, there exist uniquely determined $(P, \tilde{Q})$ which satisfy the equations in (51). Since by perfect WI, the proof ϕ is independent of its witness $(M, \tilde{N}, P, \tilde{Q})$, (U, ϕ) information-theoretically hides the bit b.

If the adversary did not produce two valid (pre-signatures), it receives $(\perp, \perp)$ at the end. Otherwise, the experiment computes two tuples $(A, B, \tilde{D}, R, \tilde{S})$, which satisfy (52) and gives the adversary two GS proofs of knowledge of them. Again by perfect WI, the proofs are distributed as if they were computed using a random witness (in this case a fresh signature); the proofs are thus independent of the signature-generation protocol.

Unforgeability. We show that, after running the protocol $q - 1$ times with an honest signer, no adversary can output q different messages and valid blind signatures on them. To do so, we reduce unforgeability to the security of the signature scheme ASIG, which follows from ADH-SDH and AWF-CDH (the latter being implied by SXDH). Given parameters $gk_{\mathsf{ASIG}} := (\Lambda, F, K, T)$ and a public key $(X, \tilde{Y})$ for ASIG, we first generate a key Σ_{GS} for the GS proof system as specified by Setup, but together with extraction key ek. We run the adversary on $\Sigma = (gk_{\mathsf{ASIG}}, \Sigma_{\mathsf{GS}})$. Whenever $\mathcal{A}$ sends a request (U, ϕ), we extract $(M, \tilde{N}, P, \tilde{Q})$ from ϕ using ek. Soundness of the proof ensures that by (52) there exist $m, \rho \in \mathbb{Z}_p$ such that $M = G^m$, $\tilde{N} = \tilde{G}^m$, $P = G^{\rho}$, $\tilde{Q} = \tilde{G}^{\rho}$ and $U = T^{\rho} \cdot M$. We then send $(M, \tilde{N})$ to the signing oracle of ASIG and receive a signature $(A, B, \tilde{D}, R, \tilde{S})$. We return to the adversary $(A, B, \tilde{D}, R' := R \cdot P^{-1}, \tilde{S}' := \tilde{S} \cdot \tilde{Q}^{-1})$. This perfectly simulates Signer: Since the oracle produced a correct signature, there exist c and $\hat{r}$ such that $B = F^c$, $\tilde{D} = \tilde{G}^c$, $R = G^{\hat{r}}$, $\tilde{S} = \tilde{G}^{\hat{r}}$ and $A = (K \cdot T^{\hat{r}} \cdot M)^{\frac{1}{x+c}} = (K \cdot T^{\hat{r}-\rho} \cdot U)^{\frac{1}{x+c}}$. Thus, $R' = G^{\hat{r}-\rho}$ and $\tilde{S}' = \tilde{G}^{\hat{r}-\rho}$, which corresponds to a real reply by the Signer oracle using randomness c and $r := \hat{r} - \rho$ (which is uniform).

After at most $q - 1$ requests, the adversary outputs q valid signatures on different messages. Among those messages, there exist at least one message that was not sent to the ASIG oracle. We extract the signature corresponding to that message using ek. By the soundness of the GS proof, it is a valid signature and can thus be returned as a valid forgery for ASIG. □

Acknowledgements

The authors would like to thank the anonymous reviewers of this paper. We also would like to express our appreciation to the program committee and the anonymous reviewers for CRYPTO 2010. The first author thanks Sherman S. M. Chow for his comment on group signatures in Sect. 7.1.

References

[1] M. Abe, J. Camenisch, M. Dubovitskaya, R. Nishimaki, Universally composable adaptive oblivious transfer (with access control) from standard assumptions, in *Digital Identity Management*, pp. 1–12 (2013)

[2] M. Abe, M. Chase, B. David, M. Kohlweiss, R. Nishimaki, M. Ohkubo, Constant-size structure-preserving signatures generic constructions and simple assumptions, in X. Wang and K. Sako, editors, *Advances in Cryptology—ASIACRYPT 2012*. LNCS, vol. 7658 (Springer, Berlin, 2012), pp. 4–24.

[3] M. Abe, B. David, M. Kohlweiss, R. Nishimaki, M. Ohkubo, Tagged one-time signatures: tight security and optimal tag size, in K. Kurosawa, G. Hanaoka, editors, *Public Key Cryptography—PKC 2013*. LNCS, vol. 7778 (Springer, Berlin, 2013), pp. 312–331

[4] M. Abe, S, Fehr, Perfect NIZK with adaptive soundness, in S. P. Vadhan, editor, *Theory of Cryptography–TCC2007*. LNCS, vol. 4392 (Springer, Berlin, 2007), pp. 118–136

[5] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, M. Ohkubo, Structure-preserving signatures and commitments to group elements, in T. Rabin, editor, *Advances in Cryptology—CRYPTO 2010*. LNCS, vol. 6223, pp. 209–237 (2010)

[6] M. Abe, J. Groth, K. Haralambiev, M. Ohkubo, Optimal structure-preserving signatures in asymmetric bilinear groups, in P. Rogaway, editor, *Advances in Cryptology—CRYPTO '11*. LNCS, vol. 6841 (Springer, Berlin, 2011), pp. 649–666

[7] M. Abe, K. Haralambiev, M. Ohkubo, *Signing on group elements for modular protocol designs*. IACR ePrint Archive, Report 2010/133, (2010). http://eprint.iacr.org

[8] M. Abe, K. Haralambiev, M. Ohkubo, Efficient message space extension for automorphic signatures, in *Information Security–ISC 2010*. LNCS, vol. 6531 (Springer, Berlin, 2011), pp. 319–330

[9] M. Abe, K. Haralambiev, M. Ohkubo, Group to group commitments do not shrink, in D. Pointcheval, T. Johansson, editors, *Advances in Cryptology—EUROCRYPT 2012*. LNCS, vol. 7237 (Springer, Berlin, 2012), pp. 301–317

[10] M. Abe, M. Ohkubo, A framework for universally composable non-committing blind signatures. *IJACT* **2**(3), 229–249 (2012)

[11] J. Alwen, Y. Dodis, D. Wichs, Survey: leakage resilience and the bounded retrieval model, in K. Kurosawa, editor, *Information Theoretic Security*. LNCS, vol. 5973 (Springer, Berlin, 2010), pp. 1–18

[12] G. Ateniese, J. Camenisch, S. Hohenberger, B. de Medeiros, *Practical group signatures without random oracles*. IACRePrint Archive, Report 2005/385 (2005). http://eprint.iacr.org

[13] N. Attrapadung, B. Libert, T. Peters, Computing on authenticated data: new privacy definitions and constructions, in X. Wang, K. Sako, editors, *Advances in Cryptology—ASIACRYPT 2012*. LNCS, vol. 7658 (Springer, Berlin, 2012), pp. 367–385

[14] N. Attrapadung, B. Libert, T. Peters, Efficient completely context-hiding quotable and linearly homomorphic signatures, in K. Kurosawa, G. Hanaoka, editors, *Public Key Cryptography—PKC 2013*. LNCS, vol. 7778 (Springer, Berlin, 2013) pp. 386–404

[15] M. Belenkiy, M. Chase, M. Kohlweiss, A. Lysyanskaya, P-signatures and noninteractive anonymous credentials, in R. Canetti, editor, *Theory of Cryptography—TCC 2008*. LNCS, vol. 4948, (Springer, Berlin, 2008), pp. 356–374

[16] M. Bellare, D. Micciancio, B. Warinschi, Foundations of group signatures: formal definitions, simplified requirements and a construction based on general assumptions, in E. Biham, editor, *Advances in Cryptology—EUROCRPYT '03*, LNCS, vol. 2656, pp. 614–629 (2003)

[17] M. Bellare, C. Namprempre, D. Pointcheval, M. Eko, Theone-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *J. Cryptol.***16**(3), 185–215 (2003)

[18] M. Bellare, P. Rogaway, Random oracles are practical: a paradigm for designing efficient protocols, in *First ACM Conference on Computer and Communication Security*. (Association for Computing Machinery, 1993), pp. 62–73

[19] M. Bellare, H. Shi, C. Zhang, Foundations of group signatures: the case of dynamic groups, in A. Menezes, editor, *Topics in Cryptology—CT-RSA 2005*. LNCS, vol. 3376 (Springer, Berlin, 2005), pp. 154

[20] D. Bernhard, G. Fuchsbauer, E. Ghadafi, Efficient signatures of knowledge and DAA in the standard model, in M.J. Jacobson Jr., M.E. Locasto, P. Mohassel, R. Safavi-Naini, editors, *Applied Cryptography and Network Security—ACNS 2013*. LNCS, vol. 7954. (Springer, Berlin, 2013), pp. 518–533

[21] O. Blazy, S. Canard, G. Fuchsbauer, A. Gouget, H. Sibert, J. Traoré, Achieving optimal anonymity in transferable e-cash with a judge, in A. Nitaj, D. Pointcheval, editors, *Progress in Cryptology—AFRICACRYPT 2011*. LNCS, vol. 6737 (Springer, Berlin, 2011), pp. 206–223

[22] D. Boneh, X. Boyen. Short signatures without random oracles, in C. Cachin, J. Camenisch, editors, *Advances in Cryptology—EUROCRYPT 2004*. LNCS, vol. 3027 (Springer, Berlin, 2004), pp. 56–73.

[23] D. Boneh, X. Boyen, H. Shacham, Short group signatures, in M. Franklin, editor, *Advances in Cryptology—CRYPTO '04*. LNCS, vol. 3152 (Springer, Berlin, 2004), pp. 41–55

[24] D. Boneh, M. Franklin, Identity-based encryption from the Weil pairing, in J. Kilian, editor, *Advances in Cryptology—Crypto 2001*. LNCS, vol. 2139 (Springer, Berlin, 2001), pp. 213–229

[25] D. Boneh, B. Lynn, H. Shacham, Short signatures from the Weil pairing, in C. Boyd, editor, *Advances in Cryptology—ASIACRYPT 2001*. LNCS, vol. 2248 (Springer, Berlin, 2001), pp. 514–532

[26] D. Boneh, C. Gentry, B. Lynn, H. Shacham, Aggregate and verifiably encrypted signatures from bilinear maps, in E. Biham, editor, *Advances in Cryptology—EUROCRYPT 2003*, LNCS, vol. 2656 (Springer, Berlin, 2003), pp. 416–432

[27] D. Boneh, E.-J. Goh, K. Nissim, Evaluating 2-DNF formulas on ciphertexts, in J. Kilian, editor, *Theory of Cryptography Conference—TCC'2005*. LNCS, vol. 3378 (Springer, Berlin, 2005), pp. 325–341

[28] X. Boyen, B. Waters, Compact group signatures without random oracles, in *Advances in Cryptology—EUROCRYPT 2006*. LNCS, vol. 4004 (Springer, Berlin, 2006), pp. 427–444

[29] X. Boyen, B. Waters, Full-domain subgroup hiding and constant-size group signatures, in *Public Key Cryptography—PKC 2007*, LNCS, vol. 4450 (Springer, Berlin, 2007), pp. 1–15

[30] S. Brands, *Rethinking public key infrastructure and digital certificates—building privacy*. Ph.D. thesis, (Eindhoven Institute of Technology, The Netherlands, 1999)

[31] J. Camenisch, A. Lysyanskaya, Signature schemes and anonymous credentials from bilinear maps, in *Advances in Cryptology—CRYPTO '04*. LNCS, vol. 3152 (Springer, Berlin, 2004), pp. 56–72

[32] J. Camenisch, M. Stadler. Efficient group signature schemes for large groups, in B.S. Kaliski Jr., editor, *Advances in Cryptology—CRYPTO'97*. LNCS, vol. 1294 (Springer, Berlin, 1997), pp. 410–424

[33] J. Camenisch, M. Dubovitskaya, R.R. Enderlein, G. Neven. Oblivious transfer with hidden access control from attribute-based encryption, in I. Visconti, R. De Prisco, editors, *SCN*. LNCS, vol. 7485 (Springer, Berlin, 2012), pp. 559–579

[34] J. Camenisch, K. Haralambiev, M. Kohlweiss, J. Lapon, V. Naessens, Structure preserving CCA secure encryption and applications, in D. H. Lee, X. Wang, editors, *Advances in Cryptology—ASIACRYPT 2011*. LNCS, vol. 7073 (Springer, Berlin, 2011), pp. 89–106

[35] J. Camenisch, M. Kohlweiss, C. Soriente, An accumulator based on bilinear maps and efficient revocation for anonymous credentials, in *Public Key Cryptography—PKC2009*. LNCS, vol. 5443. (Springer, Berlin, 2009), pp. 481–500

[36] J. Camenisch, M. Koprowski, B. Warinschi, Efficient blind signatures without random oracles, in C. Blundo, S. Cimato, editors, *Security in Communication Networks—SCN 2004*. LNCS, vol. 3352. (Springer, Berlin, 2005), pp. 134–148

[37] R. Canetti, O. Goldreich, S. Halevi, The random oracle methodology, revisited, in *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 209–218 (1998)

[38] J. Cathalo, B. Libert, M. Yung, Group encryption: Non-interactive realization in the standard model, in M. Matsui, editor, *Advances in Cryptology—ASIACRYPT 2009*. LNCS, vol. 5912, pp. 179–196 (2009)

[39] M. Chase, M. Kohlweiss, A. Lysyanskaya, S. Meiklejohn, Malleable proof systems and applications, in D. Pointcheval, T. Johansson, editors, *Advances in Cryptology—EUROCRYPT 2012*. LNCS, vol. 7237 (Springer, Berlin, 2012), pp. 281–300

[40] D. Chaum, Blind signatures for untraceable payments, in D. Chaum, R. Rivest, A. Sherman, editors, *Advances in Cryptology—Proceedings of Crypto'82*. (Prenum Publishing Corporation, 1982), pp. 199–204

[41] D. Chaum, E. Van Heyst, Group signatures, in D.W. Davies, editor, *Advances in Cryptology—EUROCRYPT '91*. LNCS, vol. 547 (Springer, Berlin, 1991), pp. 257–265

[42] S. Chow, Real traceable signatures, in *Selected Areas in Cryptography—SAC '09*. LNCS, vol. 5867 (Springer, Berlin, 2009), pp. 92–107

[43] M. Fischlin, Round-optimal composable blind signatures in the common reference model, in C. Dwork, editor, *Advances in Cryptology—CRYPTO 2006*. LNCS, vol. 4117 pp. 60–77 (2006)

[44] M. Fischlin, D. Schröder, Security of blind signatures under aborts, in *Public Key Cryptography—PKC2009*. LNCS, vol. 5443 (Springer, Berlin, 2009), pp. 297–316

[45] G. Fuchsbauer, *Automorphic signatures in bilinear groups*. Cryptology ePrint Archive, Report 2009/320 (2009). http://eprint.iacr.org/

[46] G. Fuchsbauer. Commuting signatures and verifiable encryption, in K.G. Paterson, editor, *Advances in Cryptology—EUROCRYPT 2011*. LNCS, vol. 6632 (Springer, Berlin, 2011), pp. 224–245

[47] G. Fuchsbauer, D. Pointcheval, Anonymous proxy signatures, in R. Ostrovsky, R. De Prisco, I. Visconti, editors, *Security in Communication Networks—SCN 2008*. LNCS, vol. 5229 (Springer, Berlin, 2008), pp. 201–217

[48] G. Fuchsbauer, D. Pointcheval, D. Vergnaud. Transferable constant-size fair e-cash, in J.A. Garay, A. Miyaji, A. Otsuka, editors, *Cryptology and Network Security—CANS 2009*. LNCS, vol. 5888 (Springer, Berlin, 2009), pp. 226–247

[49] G. Fuchsbauer, D. Vergnaud, Fair blind signatures without random oracles, in D.J. Bernstein, T. Lange, editors, *Progress in Cryptology—AFRICACRYPT 2010*. LNCS, vol. 6055 (Springer, Berlin, 2010), pp. 16–33

[50] J. Furukawa, K. Sako, An efficient scheme for proving a shuffle, in J. Kilian, editor, *Advances in Cryptology—CRYPTO 2001*. LNCS, vol. 2139 (Springer, Berlin, 2001), pp. 368–387

[51] S.D. Galbraith, K.G. Paterson, N.P. Smart, Pairings for cryptographers. *Discrete Appl. Math.***156**(16), 3113–3121 (2008)

[52] S. Goldwasser, S. Micali, R. Rivest, A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.***17**(2), 281–308 (1988)

[53] M. Green, S. Hohenberger, Universally composable adaptive oblivious transfer, in J. Pieprzyk, editor, *Advances in Cryptology—ASIACRYPT*. LNCS, vol. 5350, pp. 179–197 (2008)

[54] J. Groth, Simulation-sound NIZK proofs for a practical language and constant size group signatures, in X. Lai, K. Chen, editors, *Advances in Cryptology—ASIACRYPT 2006*. LNCS, vol. 4284 (Springer, Berlin, 2006), pp. 444–459

[55] J. Groth, Fully anonymous group signatures without random oracles, in *Advances in Cryptology–ASIACRYPT 2007*. LNCS, vol. 4833. (Springer, Berlin, 2007), pp. 164–180

[56] J. Groth, *Homomorphic trapdoor commitments to group elements*. IACR ePrint Archive, Report 2009/007, January 2009. Update version available from the author's homepage

[57] J. Groth, Linear algebra with sub-linear zero-knowledge arguments, in *Advances in Cryptology—CRYPTO 2009*. LNCS, vol. 5677, pp. 192–208 (2009)

[58] J. Groth, Efficient zero-knowledge arguments from two-tiered homomorphic commitments, in *Advances in Cryptology—ASIACRYPT 2011*. LNCS (Springer, Berlin, 2011)

[59] J. Groth, A. Sahai, Efficient non-interactive proof systems for bilinear groups, in N.P. Smart, editor, *Advances in Cryptology—EUROCRYPT 2008*. LNCS, vol. 4965 (Springer, Berlin, 2008), pp. 415–432

[60] L.C. Guillou, J.-J. Quisquater, A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory, in C.G. Günther, editor, *Advances in Cryptology—EUROCRYPT '88*. LNCS, vol. 330 (Springer, Berlin, 1988), pp. 123–128

[61] C. Hazay, J. Katz, C. Koo, and Y. Lindell, Concurrently-secure blind signatures without random oracles or setup assumptions, in *Theory of Cryptography Conference—TCC 2007*. LNCS, vol. 4392 (Springer, Berlin, 2007), pp. 323–341

[62] D. Hofheinz, T. Jager, Tightly secure signatures and public-key encryption, in *Advances in Cryptology—CRYPTO 2012*. LNCS, vol. 7417 (Springer, Berlin, 2012), pp. 590–607

[63] A. Joux, A one round protocol for tripartite Diffie–Hellman, in W. Bosma, editor, *Algorithmic Number Theory—ANTS-IV 2000*. LNCS, vol. 1838 (Springer, Berlin, 2000), pp. 385–394

[64] A. Juels, M. Luby, R. Ostrovsky, Security of blind digital signatures, in B.S. Kaliski Jr., editor, *Advances in Cryptology—CRYPTO '97*. LNCS, vol. 1294. (Springer, Berlin, 1997), pp. 150–164

[65] J. Katz, *Digital Signatures*, (Springer, Berlin, 2010)

[66] A. Kiayias, H. Zhou, Concurrent blind signatures without random oracles, in *Security in Communication Networks—SCN 2006*. LNCS, vol. 4116 (Springer, Berlin, 2006), pp. 49–62

[67] A. Kiayias, H. Zhou. Equivocal blind signatures and adaptive UC-security, in R. Canetti, editor, *Theory of Cryptography Conference—TCC 2008*. LNCS, vol. 4948 (Springer, Berlin, 2008), pp. 340–355

[68] A. Kiayias, M. Yung, Group signatures with efficient concurrent join, in R. Cramer, editor, *Advances in Cryptology—EUROCRYPT 2005*. LNCS, vol. 3494 (Springer, Berlin, 2005), pp. 198–214

[69] H. Krawczyk, T. Rabin, *Chameleon Hashing and Signatures*. Technical Report 1998/010, IACR ePrint archive (1998)

[70] S. Kunz-Jacques, D. Pointcheval, About the security of MTI/C0 and MQV, in R. De Prisco, M. Yung, editors, *Security in Communication Networks—SCN 2006*. LNCS, vol. 4116 (Springer, Berlin, 2006), pp. 156–172

[71] B. Libert, T. Peters, M. Joye, M. Yung, Linearly homomorphic structure-preserving signatures and their applications, in R. Canetti, J. Garay, editors, *Advances in Cryptology—CRYPTO 2013*. LNCS (Springer, Berlin, 2013)

[72] B. Libert, T. Peters, M. Yung, Group signatures with almost-for-free revocation, in R. Safavi-Naini, R. Canetti, editors, *Advances in Cryptology—CRYPTO 2012*. LNCS, vol. 7417 (Springer, Berlin, 2012), pp. 571–589

[73] B. Libert, T. Peters, M. Yung, Scalable group signatures with revocation, in *Advances in Cryptology—EUROCRYPT 2012*. LNCS (Springer, Berlin, 2012)

[74] B. Libert, D. Vergnaud, Multi-use unidirectional proxy re-signatures, in P. Ning, P. F. Syverson, S. Jha, editors, *ACM Conference on Computer and Communications Security* (ACM, 2008), pp. 511–520

[75] B. Libert, D. Vergnaud, Group signatures with verifier-local revocation and backward unlinkability in the standard model, in *Cryptology and Network Security—CANS 2009* (Springer, Berlin, 2009), pp. 498–517

[76] B. Libert, D. Vergnaud, Group signatures with verifier-local revocation and backward unlinkability in the standard model, in *Cryptology and Network Security—CANS 2009*. (Springer, Berlin, 2009), pp. 498–517

[77] H. Lipmaa, Verifiable homomorphic oblivious transfer and private equality test, in C.-S. Laih, editor, *Advances in Cryptology—ASIACRYPT 2003*, LNCS, vol. 2894 (Springer, Berlin, 2003), pp. 416–433

[78] A. Lysyanskaya, R.L. Rivest, A. Sahai, S. Wolf, Pseudonym systems, in *Selected Areas in Cryptography—SAC '99*. LNCS, vol. 1758 (Springer, Berlin, 2000), pp. 184–199

[79] P. Mohassel, One-time signatures and chameleon hash functions, in A. Biryukov, G. Gong, D.R. Stinson, editors, *Selected Areas in Cryptography—SAC 2010*. LNCS, vol. 6544 (Springer, Berlin, 2011), pp. 302–319

[80] M. Naor. On cryptographic assumptions and challenges, in D. Boneh, editor, *Advances in Cryptology—CRYPTO 2003*. LNCS, vol. 2729 (Springer, Berlin, 2003), pp. 96–109

[81] C.A. Neff, A verifiable secret shuffle and its application toe-voting, in M.K. Reiter, P. Samarati, editors, *ACM Conference on Computer and Communications Security—CCS 2001* (ACM, 2001), pp. 116–125

[82] T. Okamoto, Efficient blind and partially blind signatures without random oracles, in S. Halevi, T. Rabin, editors, *Theory of Cryptography Conference—TCC 2006*. LNCS, vol. 3876 (Springer, Berlin, 2006), pp. 80–99. Full version available onePrint archive

[83] T.P. Pedersen, A threshold cryptosystem without a trusted party, in D.W. Davies, editor, *Advances in Cryptology—EUROCRYPT'91*. LNCS, vol. 547 (Springer, Berlin, 1991), pp. 522–526

[84] D. Pointcheval, J. Stern, Security arguments for digital signatures and blind signatures. *J. Cryptol.***13**(3), 339–360 (2000)

[85] M. Rückert, D. Schröder, Security of verifiably encrypted signatures and a construction without random oracles, in H. Shacham, B. Waters, editors, *Pairing-Based Cryptography—PAIRING 2009*. LNCS, vol. 5671 (Springer, Berlin, 2009), pp. 17–34

[86] R. Sakai, M. Kasahara, Cryptosystems based on pairing over elliptic curve (in japanese), in *Symposium on Cryptography and Information Security*. SCIS, vol. SCIS00-C20 (2000)

[87] J.T. Schwartz, Fast probabilistic algorithms for verification of polynomial identities. *J. ACM* **27**(4) (1980)

[88] V. Shoup. Lower bounds for discrete logarithms and related problems, in W. Fumy, editor, *Advances in Cryptology—EUROCRYPT '97*. LNCS, vol. 1233 (Springer, Berlin, 1997), pp. 256–266

[89] J. Zhang, Z. Li, H. Guo. Anonymous transferable conditional e-cash, in A.D. Keromytis, R. Di Pietro, editors, *Secure Comm. LNICST*. vol. 106 (Springer, Berlin, 2012), pp. 45–60