A Domain Transformation for Structure-Preserving Signatures on Group Elements

Melissa Chase and Markulf Kohlweiss

[1] Microsoft Research, melissac@microsoft.com
[2] KU Leuven, ESAT-COSIC / IBBT, markulf.kohlweiss@esat.kuleuven.be

Abstract. We present a generic transformation that allows us to use a large class of pairing-based signatures to construct schemes for signing group elements in a structure preserving way. As a result of our transformation we obtain a new efficient signature scheme for signing a vector of group elements that is based only on the well established decisional linear assumption (DLIN). Moreover, the public keys and signatures of our scheme consist of group elements only, and a signature is verified by evaluating a set of pairing-product equations. In combination with the Groth-Sahai proof system, such a signature scheme is an ideal building block for many privacy-enhancing protocols.

To do this, we start by proposing a new stateful signature scheme for signing vectors of exponents that is F-unforgeable under weak chosen message attacks. This signature scheme is of independent interest as it is compatible with Groth-Sahai proofs and secure under a computational assumption implied by DLIN. Then we give a general transformation for signing group elements based on signatures (for signing exponents) with efficient non-interactive zero-knowledge proofs. This transform also removes any dependence on state in the signature used to sign exponents. Finally, we obtain our result by instantiating this transformation with the above signature scheme and Groth-Sahai proofs.

1 Introduction

Computational assumptions are essential for cryptography, however, our goal as cryptographers is to base our constructions on the weakest possible assumption. Sometimes we must make compromises for the sake of efficiency — if no efficient scheme is known based on weak assumptions, in many cases it may be necessary to accept a stronger assumption in order to obtain a scheme that is practical. But it is important to understand what these tradeoffs are, and to continue to look for efficient constructions based on the weakest possible assumptions [Nao03]. In this work we consider the problem of structure preserving signatures, and show that, contrary to what was previously believed, it is possible to design schemes based on significantly weaker assumptions with only a relatively small loss in efficiency.

***Structure Preserving Signatures*[3].** In most settings it is straightforward to sign elements of any message space. We simply view the message as a binary string and apply a collision resistant hash function to map it into the desired range (usually $\mathbb{Z}_p$ or $\mathbb{Z}_n$) at which point it can be signed using constructions based on number theoretic primitives. For most applications this works well – it allows us to sign messages in any message space, and as an apparent added benefit the hash function destroys all structure in the values that are being signed, which in many cases allows us to require weaker properties of the number theoretic primitive.

However, in some applications there is also a disadvantage to eliminating the structure of the message space. In particular, without this structure it seems to be much more difficult to build efficient protocols for dealing with signatures on hidden messages, e.g. for proving knowledge of a signature on a hidden message, or issuing a signature given only the commitment to the message (as in blind signatures).

[3] This term was introduced recently in [AHO10], but it nicely captures the idea behind a lot of earlier work as well (as we will discuss).

Such protocols are essential in numerous privacy-enhancing applications such as group signatures [ACJT00], anonymous credentials [CL01,BCL04], compact e-cash [CHL05,CHL06,CLM07], range proofs [CCS08], oblivious database access [CGH09], and others [CHK+06,TS06,CGH06]. All of these schemes make use of a form of privacy enhanced certification: instead of revealing a certificate, which might include private information, one can prove knowledge of a certificate with certain properties. One of the key elements in this approach is the ability to prove that certain hidden values have been signed without revealing the signature nor all of the certified values. This is generally done by committing to or encrypting the desired values, and then giving a proof of knowledge of an opening to the commitments and a signature on these committed values.

While such protocols are extremely useful, there are relatively few known efficient constructions. Of course one could construct these protocols based on general commitment schemes and proofs of knowledge. However, these general building blocks are extremely inefficient. A far more practical approach is to consider particular languages for which we can generate efficient proofs and efficient protocols using Σ-protocols [CDS94,Cra97,Dam02] or the recent proof system of Groth and Sahai [GS08]. These protocols rely on the structure of the underlying groups to generate efficient proofs for large classes of statements.

This is where hash functions cease to be useful as universal domain extenders for digital signatures. If the original message must be first hashed and then signed, then a proof that a committed message has been signed must not only prove knowledge of a valid signature on the resulting hash, but must also prove that the pre-image of this value is contained in the given commitment. For most modern hash functions it is completely unclear how to do this efficiently.

Consequently, the known efficient signature schemes used in the above applications, which are sometimes referred to as CL-signatures [CL02], focus on signing elements of $\mathbb{Z}_p$ or $\mathbb{Z}_n$, where no hashing is necessary so that protocols can take advantage of the structure of the underlying message space.

As described above, CL-signatures have been very useful in a wide variety of applications. However, they do have significant limitations. First, the resulting proof systems must be either interactive or in the random oracle model. This means, among other things, that it will be impossible to give a proof of knowledge of a proof that a message has been signed. This is unfortunate, since such an approach seems to be the key to allowing delegation in anonymous scenarios [CG08,CL06,FP08]. Furthermore, in many cases we need to prove knowledge of a signature on a public key, a ciphertext, a commitment, or another signature. This can be difficult since these values are often group elements and thus not elements of the original message space. An additional disadvantage is that the known efficient constructions of CL-signatures require significantly stronger assumptions than traditional signature schemes.

Pairing Based Constructions Because of these limitations, there have been a number of efforts in recent years to look for alternate constructions. Many of these efforts have focused on constructions in bilinear groups because of their rich mathematical structure. In this setting public keys, ciphertexts, and signatures are usually group elements, and so the ideal scheme would be one whose message space is the elements of the bilinear group.

Recently there has been significant interest in the construction of efficient signature schemes for signing group elements: Cathalo, Libert, and Yung [CLY09] presented a construction based on the combination of the q-Hidden Strong Diffie-Hellman (q-HSDH), Flexible Diffie-Hellman (FDH) and the Decisional Linear (DLIN) assumption, and in independent parallel work Abe, Haralambiev and Ohkubo [AHO10] propose structure-preserving signature scheme based on the new q-Simultaneous Flexible Pairing (q-SFP) assumption. [4] In a very recent work, Abe,

[4] The works of [Fuc09] and [AHO10] have been combined in [AFG+10].

Groth, Haralambiev and Ohkubo [AGHO11] give a scheme in which signatures consist of only 3 group elements, whose security is based on an interactive assumption that can be justified in the generic group model. There were also several earlier protocols which made use of adhoc structure-preserving signature schemes that relied on very strong assumptions [AWSM07,ASM08,GH08].

However, all known efficient schemes are based on so-called "q-type" or interactive assumptions that are primarily justified based on the Generic Group model.[5]

Thus, we ask whether it is possible to construct structure preserving signatures for bilinear group elements based on weaker assumptions. Ideally we would like to be able to base privacy-protecting cryptography on the same assumptions as conventional pairing-based cryptography.

One partial result in this direction is the scheme by Groth [Gro06], which satisfies the standard notion of EUF-CMA security and is based on the decisional linear assumption(DLIN). DLIN is one of the weakest assumptions used in the pairing-based setting, and is also one of the assumptions underlying the Groth-Sahai proof system, so it seems a fairly natural choice. However, while asymptotically efficient, a signature in Groth's scheme requires as confirmed by the author himself [Gro07] "thousands if not millions of group elements" per signature, so it is mainly of theoretical interest.

We focus on achieving *efficient* constructions based on the DLIN assumption. We do pay some price for our weaker assumptions; however we can show that the difference in efficiency is not as great as was previously thought: protocols based on our primitives are within an order of magnitude or two of the most efficiency of the efficient protocols mentioned above. Thus, our work helps to explore the security/efficiency tradeoff in this setting. In cases where high efficiency is critical it may be best to use one of the more efficent schemes and hope that the q-type assumptions remain secure. On the other hand, in other cases where a slightly higher time/space cost is acceptable, we provide the alternative option of a safer construction based on a weaker assumption.

Our results. We show how to transform any signature for signing elements of $\mathbb{Z}_p$ (with certain additional properties) into a structure preserving signature scheme for signing bilinear group elements. Signature schemes for signing elements of $\mathbb{Z}_p$ seem to be simpler to construct, and there are a number of constructions based on various assumptions [Wat05,BCKL08,BCKL09,Fuc09]. Thus, this already generates a range of structure preserving signatures schemes.

However, all of these possible underlying signature constructions are based on fairly strong q-type assumptions, and thus they don't help us to achieve our final goal. Instead, we construct a new DLIN based signature scheme with the necessary properties based on the scheme of Hohenberger and Waters (HW) [HW09a].

Combining this with our transformation yields our final result: a structure preserving signature scheme whose security is based on the DLIN assumption, which is among the weakest assumptions used in the bilinear group setting.[6]

A significant advantage of our proposal is that we provide a general transform for building signature schemes for signing group elements from any signature schemes for signing exponents in $\mathbb{Z}_p$ that support efficient non-interactive zero-knowledge proofs of signature possession. This means that any progress in these areas, e.g., a new more efficient NIZK proof system for statements about group elements, or a more efficient signature scheme for exponents based on weak assumptions, will automatically result in improved signature schemes and proof protocols for group elements based on those assumptions.

[5] The parameter q influences the instance size of the assumption and depends on the number of signatures an adversary is allowed to see.

[6] Alternatively if we use a different instantiation of GS proofs, we can also prove our scheme secure based on the SXDH assumption and an additional computational assumption that is implied by DLIN in the asymmetric pairing setting.

Our approach. Our solution makes use of pairwise independent hash functions and signatures (for signing exponents) that have efficient zero-knowledge proofs of knowledge. Intuitively instead of hashing messages and signing the hash, we certify a pairwise independent hash function $f_z : \mathcal{M} \to \mathbb{G}$ where $\mathcal{M} = \mathbb{G}^\ell$, and append the output of the hash $S = f_z(M)$ to the certificate. Each hash-function is used exactly once. If we can guarantee that the adversary cannot learn any useful knowledge from the certification process about f_z, then the adversary's probability of guessing a correct $\tilde{S}$ for a message $\tilde{M} \neq M$ is bounded by $1/|\mathbb{G}|$.

For the certification of f_z we make use of the signature scheme for exponents and its zero-knowledge proof of knowledge protocol. Our transformation has the additional advantage that even if this underlying scheme is stateful and only weakly secure, the result will be a structure preserving signature scheme which is stateless and secure under the standard EUF-CMA definition.

Applications. Structure preserving signatures on group elements have found many cryptographic applications [GH08,CLY09,Fuc09,AHO10,AFG+10]. A common element in all of these constructions is the use of structure preserving signatures to sign cipher-texts, commitments, and public keys; to preserve privacy the recipient of the signature only proves possession of a signature, while keeping the signatures and the cipher-texts, commitments, and public keys hidden.

For example, universally composable oblivious transfer [GH08] is obtained using an assisted decryption technique in which the recipient proves that he is asking for the decryption of a valid (signed) ciphertext. A universally composable blind signature scheme [Fis06] is obtained by signing a commitment, and proving possession of a signature on a commitment that opens to the blindly signed message. Signing public keys allows for an implementation of group signatures that fulfills the strong security definitions of [BMW03] and supports the concurrent join of new users [KY05]. Similar improvements carry over to P-signatures [BCKL08], anonymous proxy signatures [FP08], and delegatable anonymous credentials [BCC+09]. For details on these constructions we refer to [AFG+10] and to Appendix A.

2 Background

Our structure preserving signatures construction will make use of three main building blocks: the Groth-Sahai pairing-based proof system [GS08], a pairwise independent hash function, and a signature scheme for signing elements of $\mathbb{Z}_p$.

2.1 The Groth-Sahai proof system.

Pairing-based cryptography has led to several cryptographic advancements. One of these advancements is the development of powerful and efficient non-interactive zero-knowledge proofs in the common reference string model. The basic premise behind this approach is to hide the values for the evaluation of the bilinear map in a commitment whose parameters are given as part of the reference string. Using different commitment schemes, this idea was used to build non-interactive proof systems under the sub-group hiding [GOS06b] and under the decisional linear assumption [GOS06a].

These proof systems prove circuit satisfiability, and thus by the Cook-Levin theorem [Coo71] allow one to prove membership for every language in **NP**. The size of the common reference string and the proofs is relatively small, however transforming a statement into a Boolean circuit causes a considerable overhead.

Groth and Sahai [GS08] extended this construction and gave proof systems under the sub-group hiding, decisional linear, and external Diffie-Hellman assumptions that allow one to directly prove the pairing product equations common in pairing-based cryptography.

While proofs for such equations can also be implemented using interactive proofs about discrete logarithms, such proofs can only be made non-interactive in the random oracle model. Moreover, Groth-Sahai proofs give us something that we do not know how to obtain with random oracles: the randomizability property introduced by [BCC+09].

Groups with a bilinear map. Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be groups of prime order p. A bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ must satisfy the following properties: (a) *Bilinearity*: a map e is bilinear if $e(a^x, b^y) = e(a, b)^{xy}$; (b) *Non-degeneracy*: for all generators $g \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$, $e(g, h)$ generates $\mathbb{G}_T$; (c) *Efficiency*: There exists a p.p.t. algorithm to generate the bilinear group setup $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h)$ and an efficient algorithm to compute $e(a, b)$ for any $a \in \mathbb{G}_1$, $b \in \mathbb{G}_2$.

If there exist two efficiently computable homomorphisms that map elements of $\mathbb{G}_1$ to elements of $\mathbb{G}_2$ and elements of $\mathbb{G}_2$ to elements of $\mathbb{G}_1$, we speak about a symmetric bilinear map and simplify the notation to $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$.

DLIN and SXDH assumption. We recall the DLIN and SXDH assumption. In our analysis we refer to, but do not make use of, a variety of other, much stronger assumptions. We summarize these assumptions in the full version of this work.

Definition 1 (Decision Linear (DLIN) [BBS04]).
Given $g, g^a, g^b, g^{ac}, g^{bd}, Z \in \mathbb{G}$, for random exponents $a, b, c, d \in \mathbb{Z}_p$, decide whether $Z = g^{c+d}$ or a random element in $\mathbb{G}$. The Decision Linear assumption holds if all p.p.t. algorithms have negligible (with respect to the bit length of p) advantage in solving the above problem.

Definition 2 (External Diffie-Hellman (XDH)).
The XDH assumption requires that the DDH assumption holds for a group with a bilinear map. By necessity this can only be the case for an asymmetric bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Moreover, w.l.o.g., say that DDH should hold for $\mathbb{G}_1$, there must not exist efficiently computable homomorphisms that map elements of $\mathbb{G}_1$ to elements of $\mathbb{G}_2$.

If homomorphisms in both directions are excluded, and if DDH is also required to hold for $\mathbb{G}_2$, the combined assumption is called Symmetric XDH (SXDH) assumption.

Groth-Sahai proofs. The Groth-Sahai proof system can generate non-interactive zero-knowledge proofs of knowledge of values satisfying pairing product equations. We denote a proof π that proves knowledge of secret values $x_1, \ldots, x_N$ that fulfill a pairing product equation with constants $\{a_i\}_{i=1..N} \in \mathbb{G}, t \in \mathbb{G}_T$ and $\{\gamma_{i,j}\}_{i=1..N, j=1..N}$ by

$$\pi \leftarrow \mathsf{NIZKPK}\{(x_1, \ldots, x_N) : \prod_{i=1}^{N} e(a_i, x_i) \prod_{i=1}^{N} \prod_{j=1}^{N} e(x_i, x_j)^{\gamma_{i,j}} = t\} \ .$$

In a nutshell, Groth-Sahai proofs work by committing to all secret elements using either Linear [BBS04] or ElGamal [EG85] commitments (depending on the assumption used). The homomorphic properties of these commitments allow one to evaluate the pairing product equation in the committed domain. In addition, a Groth-Sahai proof contains a constant number of group elements that allow a verifier to check that the result of this computation corresponds to t. The verification algorithm only consists of pairings between the group elements of the commitments and these additional proof elements.

Linear and ElGamal commitments are extractable. Given a setup with an extraction trapdoor, we can extract the committed value x_i from a proof, but not the opening $open_i$.

Randomizing Groth-Sahai proofs. Both [BCC+09] and [FP09] observe that Groth-Sahai proofs can be rerandomized. A rerandomized proof is indistinguishable from a freshly generated proof of the same statement, even given all secret information about the original proof.

Belenkiy et al. [BCC+09] formally define rerandomizable non-interactive proofs. The randomization algorithm $\mathsf{RandProof}((C_1,\ldots,C_\ell),(open'_1,\ldots,open'_\ell),\pi)$ takes a Groth-Sahai proof, a list of commitments, and a list of opening updates as input. It outputs a proof π' that looks like a freshly generated proof for the same equations but for randomized commitments $C_1 \odot \mathsf{Com}(1, open'_1),\ldots, C_\ell \odot \mathsf{Com}(1, open'_\ell)$. For a commitment $C = \mathsf{Com}(x, open)$, $C \odot \mathsf{Com}(1, open') = \mathsf{Com}(x, open + open')$. Linear and ElGamal commitments with randomness 0 correspond to the committed value itself. As a consequence, the pairing product equation constants $\{a_q\}_{q=1\ldots Q} \in \mathbb{G}$ can also be randomized and thus turned into commitments to secret values.

We give more details on Linear and ElGamal commitments, Groth-Sahai proofs, and randomizing Groth-Sahai proofs in Appendix F.

2.2 Signatures for signing exponents

Our construction will also require a signatures scheme for signing elements of $\mathbb{Z}_p^\ell$ which is F-unforgeable under a weak chosen message attack. Intuitively, F-unforgeability guarantees that it is hard for the adversary to produce $F(m)$ and a signature on m for an m that wasn't signed; this is important because when the message space is $\mathbb{Z}_p$, Groth-Sahai proofs only allow one to efficiently prove knowledge of some function of the message (e.g. g^m). We will see later that the signature schemes for $\mathbb{Z}_p$ will only be used to sign random message, thus security under weak chosen message attacks will suffice. We now formally define these notions:

Definition 3 (Unforgeability under Weak Chosen Message Attacks [BB04,HW09b]). *In a weak chosen message attack, we require that the adversary submit all signature queries before seeing the public key. A signature scheme is unforgeable under weak chosen message attacks if for all $\mathcal{A}_1$, $\mathcal{A}_2$ there exists a negligible function ν such that*

$$\begin{aligned}\Pr[&(m_1,\ldots,m_Q,state) \leftarrow \mathcal{A}_1(1^k); (sk,pk) \leftarrow \mathsf{SigKg}(1^k); \sigma^{(i)} = \mathsf{Sign}(sk,m_i) \text{ for } i=1,\ldots,Q;\\ &(\tilde{\sigma},\tilde{m}) \leftarrow \mathcal{A}_2(state, pk, \sigma^{(1)},\ldots,\sigma^{(Q)}) :\\ &\tilde{m} \notin \{m_1,\ldots,m_Q\} \wedge \mathsf{SigVerify}(pk,\tilde{m},\tilde{\sigma}) = \mathsf{accept}] = \nu(k)\ .\end{aligned}$$

This definition is generalized to Weak CMA F-Unforgeability for some bijection F in the natural way. Instead of $\tilde{m}$, $\mathcal{A}_1$ only has to output $\tilde{f}$, such that $F^{-1}(\tilde{f}) \notin \{m_1,\ldots,m_Q\} \wedge \mathsf{SigVerify}(pk, F^{-1}(\tilde{f}), \tilde{\sigma}) = \mathsf{accept}$.

We present a new signature scheme satisfying this definition based on DLIN in Section 3.1, and discuss some other possible instantiations in Section 3.4.

2.3 Pairwise independent hash functions.

The final ingredient will be a family of pairwise independent hash functions. This will be a family of functions parameterized by a "key" z. Intuitively, pairwise independence means that knowing the result of a random hash function on any one input gives no information about the result of that function on any other point. More formally:

Definition 4. *A family of hash-functions $\{f_z\}_{z\in\mathcal{Z}}$, where $f_z : \mathcal{M} \to \mathcal{R}$ is called pairwise independent if $\forall x \neq y \in \mathcal{M}$ and $\forall a, b \in \mathcal{R}$, the probability*

$$Pr[z \leftarrow \mathcal{Z} : f_z(x) = a \wedge f_z(y) = b] = \frac{1}{|\mathcal{R}|^2}\ .$$

We will need a pairwise independent family of hash-functions $\{f_z\}$, where $f_z : \mathcal{M} \to \mathbb{G}$ with $\mathcal{M} = \mathbb{G}^\ell$ and $z \in \mathbb{Z}_p^{\ell+1}$. The function we propose is computed as

$$f_z(M_1, \ldots, M_\ell) = g^{z_0} \prod_{i=1..\ell} M_i^{z_i}$$

, where $z = (z_0, \ldots, z_\ell)$. We show that this function family is indeed pairwise independent:

Theorem 1. *The above function family is pairwise independent.*

Proof. Let us express the probability

$$Pr[z \leftarrow \mathcal{Z} : f_z(x) = a \wedge f_z(y) = b] = \frac{|\{z_0, \ldots, z_\ell \mid g^{z_0} \prod_{i=1..\ell} x_i^{z_i} = a \wedge g^{z_0} \prod_{i=1..\ell} y_i^{z_i} = b\}}{|\mathbb{Z}_p|^{\ell+1}}$$

We have to show that the numerator equals $|\mathbb{Z}_p|^{\ell-1}$. This can be seen by looking at $g^{z_0} \prod_{i=1..\ell} x_i^{z_i} = a$ and $g^{z_0} \prod_{i=1..\ell} y_i^{z_i} = b$ as independent linear equations over the variables $z_0, \ldots, z_\ell$ (independence follows from $x \neq y$). As there are $\ell+1$ variables and 2 equations, the solution set has $\ell-1$ dimensions and thus has size $|\mathbb{Z}_p|^{\ell-1}$. For a more formal proof see Appendix C.

3 A Signature Scheme for Signing Group Elements under Standard Assumptions

Our main result is to show how to construct a signature scheme for signing group elements based on an efficient zero-knowledge proof system and two basic building blocks. The first is a signature scheme $\mathsf{Sig}_{n\cdot exp}$ for signing $\ell + 1$ exponents that has an efficient zero-knowledge proof of knowledge (NIZKPK) of a signature on a committed message. The second building block is a pairwise independent family of hash-functions $\{f_z\}$, where $f_z : \mathcal{M} \to \mathbb{G}$ with $\mathcal{M} = \mathbb{G}^\ell$ and $z \in \mathbb{Z}_p^{\ell+1}$.

The basic idea is that, instead of hashing messages and signing the hash, we certify the key $z = (z_0, \ldots, z_\ell)$ of a pairwise independent hash function and append the output of the hash $S = f_z(M)$ to the certificate. Each hash-function key z is used exactly once so the hash value S does not help an attacker to find the hash (under the same key) of any other message. Then, for the certification of z we make use of the signature scheme for exponents and its zero-knowledge proof of knowledge protocol. This allows us to guarantee that the adversary cannot learn any useful knowledge from the certification process about z and thus even given many signatures, he is not able to guess a hash value S' for any message M' different from M.

The organization of this section is as follows: We first describe an instantiation of the first building block that is based on a weak assumption in Section 3.1, then Section 3.2, we describe the transform which we will use to construct a structure preserving signature scheme from these two primitives.

3.1 A Signature Scheme with an Efficient NIPK under Standard Assumptions

We will base our exponent-signature scheme $\mathsf{Sig}_{n\cdot exp}$ on the Hohenberger and Waters [HW09a] stateful signature scheme which was proved secure under the CDH assumption. In that scheme, each signature is indexed by a unique index s that is initialized to 0, and increased before each signing. A signature with message m, secret key a, public bases u, v, d, w, z, and randomness t, r consists of two group elements $\sigma_1 = (u^m v^r d)^a (w^{\lceil \lg(s) \rceil} z^s h)^t$ and $\sigma_2 = g^t$, and the two exponents $r, s \in \mathbb{Z}_p$. We adapt their scheme to obtain a stateful signature that is F-unforgeable under weak chosen message attacks (Weak CMA F-unforgeable) under the Randomized Computational Diffie-Hellman (RCDH) assumption, a new assumption which is implied by the DLIN

assumption. We also show how to reuse the state to sign multiple message blocks. Interestingly, when we apply the transformation presented in Section 3.2, the result will be a fully secure, stateless signature scheme for signing group elements.

Remark 1. Hohenberger and Waters pose the construction of a signature scheme with efficient protocols [CL02] (also known as a CL-signature scheme) based on their signature scheme as an interesting open problem. We give a partial answer by describing an efficient zero-knowledge proof protocol for such a scheme that is more efficient than a similar protocol for the Waters [Wat05] signature scheme as, e.g., described by [FP09]: In that scheme the number of group elements in the public key and the proof grows linearly with the number of bits in the message while the cost of our scheme only depends on the bit-size of the state s (and the number of *group elements* required to represent the message).

Simplifying the Hohenberger and Waters scheme. Recall that in the HW scheme, signatures include elements $\sigma_1 = (u^m v^r d)^a (w^{\lceil \lg(s) \rceil} z^s h)^t$ and $\sigma_2 = g^t$, and the two exponents $r, s \in \mathbb{Z}_p$. When building a zero-knowledge proof of knowledge of signature possession, we must prove that the signature is well formed, which in this case requires proving the correspondence between $\lceil \lg(s) \rceil$ and s. This typically involves two steps: 1) proving that a commitment contains the value $2^{\lceil \lg(s) \rceil}$, and 2) proving that this value is bigger than s. The range proof technique by [Bou00] for interactively proving the latter relation for large s uses hidden order groups and is based on the Strong RSA assumption. To obtain a scheme that is based purely on CDH, one has to use alternative range proof techniques, e.g. [BCDvdG87]. While such proofs can be efficiently computed ([Bou00] estimates a proof size of 27.5 kB), we are primarily interested in non-interactive proofs based on the Groth-Sahai proof system.

As pointed out in [HW09a], instead of signing $\lg(s)$ as part of σ_1 one can also sign s using a signature scheme that is already CMA secure under the CDH assumption, e.g. by employing the Waters signature [Wat05]. [7] While this approach may be slightly circular, it gives us a performance advantage, as the expected number of signatures is usually much smaller than the size of the message space $\mathbb{Z}_p$. Moreover, as we will see, when many messages are signed with related state (e.g. when we sign multiple message blocks at once), we need only sign a single state value, thus resulting in greater advantage.

Finally, we will show that for our transformation we only require a weak signature scheme; thus we can simplify the resulting signature scheme further by replacing the Chameleon hash $u^m v^r$ with u^m itself.

Our construction. Let $\mathbb{G}$ be a symmetric bilinear group with pairing operation $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Let $g, \hat{g}$ be random generators for $\mathbb{G}$. The resulting signature scheme is as follows:

$\mathsf{SigKg}_{exp}(1^k)$ outputs secret key $sk = (a, sk_w)$ and a public key $pk = (g, \hat{g}, g^a, u, d, z, h, pk_w)$. (The initial value of s is 0.)

$\mathsf{Sign}^s_{exp}(sk, m)$. The stateful signature algorithm Sign increases the state s. To sign a message m, it computes $\sigma_1 = (u^m d)^a (z^s h)^t$, $\sigma_2 = g^t$, and a Waters signature σ_3 on s. The algorithm outputs $\sigma = (\sigma_1, \sigma_2, \sigma_3, s)$.

$\mathsf{SigVerify}_{exp}(pk, m, \sigma)$. The verification algorithm parses σ as $(\sigma_1, \sigma_2, \sigma_3, i)$ and checks that signature σ_3 on i is valid. Then it uses the bilinear map to check that $e(\sigma_1, g) = e(u^m d, g^a) e(\sigma_2, z^i h)$.

[7] The Waters signature operates bit-by-bit on it's message, and directly proving knowledge of a valid Waters signature has cost proportional to the bit-length of the message. Thus, proving correctness of our resulting signature will thus have cost proportional to the bit-length of the maximum possible value of s rather than the bit length of the message.

Note: We write $\mathsf{Sign}^{s}_{exp}(sk, m)$ to indicate that we run the signing algorithm on state s. Notationally we assume a call-by-reference evaluation strategy, i.e. s might change it's value during the run of the algorithm and this new value will be automatically used in the next run of the algorithm. In our transformation to a stateless scheme we will write $\mathsf{Sign}^{s=0}_{exp}(sk, m)$ to indicate that the state is reset to the initial state before the algorithm is run.

A new assumption implied by DLIN. For simplicity, we introduce a new assumption that will allow us to prove the F-unforgeability of the above signature scheme. We show in Appendix B that it is implied by the DLIN assumption.

Assumption 1 (Randomized Computational Diffie-Hellman (RCDH)) *Let $\mathbb{G}$ be a group of prime order $p \in \Theta(2^k)$. For all p.p.t. adversaries $\mathcal{A}$, the following probability is negligible in k:*

$$\Pr[g, \hat{g} \leftarrow \mathbb{G}; a, b \leftarrow \mathbb{Z}_p; (R_1, R_2, R_3) \leftarrow \mathcal{A}(g, \hat{g}, g^a, g^b) : \exists r \in \mathbb{Z}_p \text{ such that } R_1 = g^r, R_2 = \hat{g}^r, R_3 = g^{abr}]$$

Theorem 2. *In groups with a symmetric bilinear pairing RCDH is implied by DLIN.* The proof can be found in Appendix B.

Security of our construction. We show that this signature scheme is unforgeable under weak chosen message attacks, and moreover, that it is F-unforgeable under such attacks for a simple function F that maps exponents to group elements. (Recall that F-unforgeability means that it is impossible produce $F(m)$ and a forged signature on m. This allows us to prove a contradiction even when we can extract only $F(m)$ and not m as is the case when we use the Groth-Sahai proof system.)

Theorem 3. *Our* $(\mathsf{SigKg}_{exp}, \mathsf{Sign}^{s}_{exp}, \mathsf{SigVerify}_{exp})$ *signature scheme is unforgeable under weak chosen message attacks under the CDH assumption.* The proof is omitted. It follows very closely the proof of F-unforgeability presented below.

We note that, as part of their result, Hohenberger and Waters [HW09b] give a generic transformation from Weak CMA security to CMA security based on Chameleon hashes. We will however show that Weak CMA F-unforgeable signatures are sufficient to obtain a CMA secure signature scheme for signing group elements via our transform.

We now state and prove security of the above signature scheme:

Theorem 4. *Let $F(m) = (g^m, \hat{g}^m)$. Our* $(\mathsf{SigKg}_{exp}, \mathsf{Sign}^{s}_{exp}, \mathsf{SigVerify}_{exp})$ *signature scheme is Weak CMA F-unforgeable under the RCDH assumption.*

Proof. A successful adversary $\mathcal{A}$ outputs a forgery $\tilde{\sigma} = (\tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{\sigma}_3, \tilde{i})$. If the signature on index $\tilde{i}$ was never created, we break the signature scheme that is used to sign the index s. Thus we concentrate on the case where the adversary reuses one of the s values from the signing queries as $\tilde{i}$. The first step in a reduction to RCDH will be to guess this $\tilde{i}$. (Here we have at most a polynomial loss in the tightness of the reduction.)

Setup: As we consider a weakly secure signature scheme, the game starts with the adversary outputting polynomially many messages $m_1, \ldots, m_Q$, $Q \leq poly(k)$. The reduction chooses a random index i^*, $1 \leq i^* \leq Q$. Given (g, g^a, g^b) as specified in the RCDH assumption, the parameters are set up as follows. Choose random $y_d \in \mathbb{Z}_p$ and set $u = g^b$, $d = g^{-bm_{i^*}} g^{y_d}$, then choose random $x_z, x_h \in \mathbb{Z}_p$, and set $z = g^b g^{x_z}$, $h = g^{-bi^*} g^{x_h}$. The reduction outputs $pk = (g, g^a, u, d, z, h)$.

Sign: The adversary is now given signatures on messages $m_1, \ldots, m_Q$, $Q \leq poly(k)$, that are computed as follows:

For $s = i^*$, choose random t and form $\sigma_1 = (g^a)^{y_d}(z^s h)^t$, $\sigma_2 = g^t$. Note that this results in a correctly distributed signature as

$$\begin{aligned}(g^a)^{y_d}(z^s h)^t =& \\ ((g^{ab})^{m_{i^*}-m_{i^*}})(g^a)^{y_d}(z^s h)^t =& \\ ((g^b)^{m_{i^*}}(g^{-bm_{i^*}}g^{y_d}))^a(z^s h)^t =& (u^{m_{i^*}}d)^a(z^s h)^t\ .\end{aligned}$$

For $s \neq i^*$, choose a random value t', and form $\sigma_1 = (g^a)^{y_d}T^{x_z s+x_h}(g^b)^{t'(s-i^*)}$, $\sigma_2 = T$ for $T = g^{t'}/(g^a)^{(m_s-m_{i^*})/(s-i^*)}$. Let implicitly $t = t' - a(m_s - m_{i^*})/(s-i^*)$, then $T = g^{t'-a(m_s-m_{i^*})/(s-i^*)} = g^t$ and

$$\begin{aligned}(g^a)^{y_d}T^{x_z s+x_h}(g^b)^{t'(s-i^*)} =& \\ (g^{y_d})^a(g^{x_z s}g^{x_h})^t(g^b)^{t'(s-i^*)} =& \\ (u^{m_s}d)^a(g^{x_z s}g^{x_h})^t(g^b)^{t'(s-i^*)}(g^{-ab})^{(m_s-m_{i^*})} =& \\ (u^{m_s}d)^a(g^{x_z s}g^{x_h})^t(g^{b(s-i^*)})^t =& \\ (u^{m_s}d)^a(g^{(b+x_z)s}g^{-bi^*+x_h})^t =& (u^{m_s}d)^a(z^s h)^t\ .\end{aligned}$$

Response: Eventually the adversary responds with a forgery $\tilde{\sigma} = (\tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{\sigma}_3, \tilde{i})$, $g^{\tilde{m}}$, $\hat{g}^{\tilde{m}}$, such that $\tilde{m} \notin \{m_1, \ldots, m_Q\}$. If $\tilde{i} \neq i^*$ the reduction aborts. Otherwise it outputs $g^{\tilde{m}}/g^{m_{i^*}}$, $\hat{g}^{\tilde{m}}/\hat{g}^{m_{i^*}}$ and $\tilde{\sigma}_1/g^{ay_d}\tilde{\sigma}_2^{(x_z\tilde{i}-x_h)}$ as a RCDH triple.

Signing Multiple Message Blocks. For our transformation, we actually need to be able to sign vector of exponents, i.e. we need our signature scheme Sign_{exp} to have message space $\mathbb{Z}_p^n$ for $n > 1$. We show how to use the above signature scheme to sign multiple messages $m_1, \ldots, m_n \in \mathbb{Z}_p$ at once.

There is also an efficiency advantage to batching several messages together: We note that the Waters signature on the index s needs to be done only once. The indices of the individual signatures will be set to $n \cdot (s-1) + 1, \ldots, n \cdot (s-1) + n$.

Our multiple message block signature is as follows:

$\mathsf{SigKg}_{exp}(1^k)$ is unchanged.

$\mathsf{Sign}^s_{n \cdot exp}(sk, m_1, \ldots, m_n)$. The signature algorithm increases the state s. To sign message m, it then computes $\sigma_{1,j} = (u^{m_j}d)^a(z^{n(s-1)+j}h)^{t_j}$, and $\sigma_{2,j} = g^{t_j}$, for $j = 1..n$ and $t_j \leftarrow \mathbb{Z}_p$. We also add a Waters signature σ_3 on s. The algorithm outputs $\sigma = (\{\sigma_{1,j}, \sigma_{2,j}\}_{j=1..n}, \sigma_3, s)$.

$\mathsf{SigVerify}_{n \cdot exp}(pk, m_1, \ldots, m_n, \sigma)$. Parse σ as $(\{\sigma_{1,j}, \sigma_{2,j}\}_{j=1..n}, \sigma_3, i)$. The verification algorithm first checks that signature σ_3 on i is valid. It uses the bilinear map to verify $e(\sigma_{1,j}, g) = e(u^{m_j}d, g^a)e(\sigma_{2,j}, z^{n(i-1)+j}h)$, for $j = 1..n$.

Unforgeability and F-unforgeability under weak CMA attacks can be shown via a straightforward extension of the proof for the single message scheme. Note that the reduction now has to guess values i^* and j^*, where $1 \leq i^* \leq Q$ and $1 \leq j^* \leq n$ respectively. The RCDH challenge is embedded into message block j^* of signature query i^*.

Efficient Zero-knowledge Proof of Knowledge. Except for the value s, the signature $\sigma = (\{\sigma_{1,j}, \sigma_{2,j}\}_{j=1..n}, \sigma_3, s)$ consists only of group elements. When employing the Groth-Sahai proof system, the Waters signature σ_3 is proved in a bit-by-bit fashion that allows us to extract s (see [FP09] for further details). It is thus possible to give proofs of knowledge for the above signature scheme using the pairing-product equation proofs in [GS08] in a straightforward way.

3.2 A Transform for Signing Group Elements

Here we present our generic transformation. When instantiated using the signature scheme described in the previous section the result will be a secure structure preserving signature under DLIN. (We can also consider other instantiations; see Section 3.4 for discussion.)

Let $\mathsf{Sig}_{n\cdot exp} = (\mathsf{SigKg}_{n\cdot exp}, \mathsf{Sign}^{s}_{n\cdot exp}, \mathsf{SigVerify}_{n\cdot exp})$ be a (potentially stateful) Weak CMA F-unforgeable signature scheme on message space $\mathbb{Z}_p^{\ell+1}$ for some bijection F. (Note that a stateless signature scheme would suffice - the construction would then simply not use the state s.) Let $\mathsf{Setup}, \mathsf{Prove}, \mathsf{VerifyProof}$ be an F-extractable non-interactive zero knowledge proof of knowledge system. Let $\mathbb{G}$ be a cyclic group of order p, and let g be a generator. We construct a stateless signature scheme with message space $\mathbb{G}^{\ell}$ as follows:

$\mathsf{SigKg}(1^k)$**:** Run $\mathsf{SigKg}_{n\cdot exp}(1^k)$ to generate a key pair (pk_{exp}, sk_{exp}). Generate the common reference string $params_{pk}$ for a NIZKPK proof system. Output $pk = (pk_{exp}, params_{pk})$ and $sk = (sk_{exp}, params_{pk})$.[8]

$\mathsf{Sign}(sk, M_1, \ldots, M_\ell)$**:** Parse $sk = (sk_{exp}, params_{pk})$. Choose random elements $z_0, \ldots, z_\ell \leftarrow \mathbb{Z}_p$. Compute the signature $\sigma' \leftarrow \mathsf{Sign}^{s=0}_{n\cdot exp}(sk, z_0, \ldots, z_\ell)$ and $S = g^{z_0}\prod_{i=1}^{\ell} M_i^{z_i}$. Finally, construct a proof of knowledge of $F(z_0), \ldots, F(z_\ell)$ and the corresponding signature, i.e.:

$$\pi \in \mathsf{NIZKPK}\{(f_0, \ldots, f_\ell, \sigma') : \{\exists (z_0, \ldots, z_\ell) \text{ s.t. for } i \in 0, \ldots \ell, \ f_i = F(z_i) \wedge$$
$$\mathsf{SigVerify}_{exp}(pk, (z_0, \ldots, z_\ell), \sigma') = 1 \wedge S = g^{z_0}\prod_{i=1}^{\ell} M_i^{z_i}\}\}$$

Output $\sigma = (S, \pi)$.

Note that we write $\mathsf{Sign}^{s=0}_{n\cdot exp}$ to indicate that in case of a stateful signature we reset the state to the initial state after each signing operation. We will see below that as the signature is always used inside of a NIZKPK this does not impact security.

$\mathsf{SigVerify}(pk, M_1, \ldots, M_\ell, \sigma)$**:** Parse $pk = (pk_{exp}, params_{pk})$ and $\sigma = (S, \pi)$. Verify the proof π w.r.t. $params_{pk}$ and $S, M_1, \ldots, M_\ell$.

We now prove our main result:

Theorem 5. *Given a (potentially stateful) Weak CMA F-unforgeable signature scheme* $(\mathsf{SigKg}_{exp}, \mathsf{Sign}^{s}_{exp}, \mathsf{SigVerify}_{exp})$ *and a secure NIZKPK proof system* $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{VerifyProof})$*,* $(\mathsf{SigKg}, \mathsf{Sign}, \mathsf{SigVerify})$ *is a stateless CMA unforgeable signature scheme.*

Proof. We formally prove the security of the transformation using a sequence of games. Let $p_i(k)$ be the probability that the adversary succeeds in **Game i**. We let **Game 1** be the EUF-CMA game for the signature scheme described above. We will show via a series of hybrid games that this probability must be negligible.

Game 1: EUF-CMA. This is the original EUF-CMA game for the signature scheme described above, i.e. signing queries are answered using Sign and the adversary succeeds if it can make $\mathsf{SigVerify}$ accept for a message vector that was never signed before.
The adversary succeeds with probability $p_1(k)$.

Game 2: Implement state updates. This game proceeds just as the EUF-CMA game except that Sign uses calls to $\mathsf{Sign}^{s}_{n\cdot exp}$ instead of calls to $\mathsf{Sign}^{s=0}_{n\cdot exp}$. This means that the state is no longer reset. Let $p_2(k)$ be the probability that the adversary succeeds in this game.

[8] Here we describe the signature as a stand alone primitive. If it is used as a building block in a bigger system, one might want to reuse parts of the system setup. For instance, one might reuse existing group parameters.

Lemma 1. *$\Delta_1(k) = |p_2(k) - p_1(k)|$ is negligible by computational witness indistinguishability property of the proof system.*

Proof. Note first that a proof system that is zero-knowledge is also witness indistinguishable. Clearly, both the signatures generated by $\mathsf{Sign}^{s=0}_{n\cdot exp}$ and by $\mathsf{Sign}^{s}_{n\cdot exp}$ correspond to valid witnesses for the NIZKPK in the signing algorithm. We first construct a sequence of hybrid games. In each hybrid an additional call to $\mathsf{Sign}^{s=0}_{n\cdot exp}$ is replaced by $\mathsf{Sign}^{s}_{n\cdot exp}$. Given an adversary $\mathcal{A}$ that has a non-negligible success difference between any of these hybrids, we can build an algorithm $\mathcal{B}$ that breaks the witness indistinguishability property of the proof system. $\mathcal{B}$ computes two witnesses w_0 and w_1 that are based on $\mathsf{Sign}^{s=0}_{n\cdot exp}$ and $\mathsf{Sign}^{s}_{n\cdot exp}$ respectively. $\mathcal{B}$ outputs w_0 and w_1 to the witness indistinguishability challenge game and uses the resulting proof π to respond to the ith signature query. Depending on the bit flipped by the challenge game, $\mathcal{A}$ will interact with one of the two hybrids. If $\mathcal{A}$ succeeds in producing a forgery, $\mathcal{B}$ outputs 1, otherwise 0. It follows that $\Delta_1(k)$ is negligible □

Game 3: reusing $\mathbf{z_i}$. This game will proceed just as **Game 2** except that once the adversary outputs his forgery, $\tilde{M}, \tilde{\sigma} = (\tilde{S}, \tilde{\pi})$, we will extracts $\tilde{f}_0, \ldots, \tilde{f}_\ell$ from $\tilde{\pi}$, and compare them against the values used to answer all of the adversary's queries. The adversary succeeds in this game if and only if the signature verifies, the message is new, and the tuple $(\tilde{f}_0, \ldots, \tilde{f}_\ell))$ corresponds to $(F(z_0), \ldots, F(z_\ell))$ for some tuple of values $(z_0, \ldots, z_\ell)$ used in a previous query. Let $p_3(k)$ be the probability that the adversary succeeds in this game.

Lemma 2. *$\Delta_2(k) = |p_3(k) - p_2(k)|$ is negligible by the F-unforgeability of the signature scheme.*

Proof. The two games differ only in the event **Bad** that $\mathcal{A}$ outputs a forgery from which values $(\tilde{f}_0, \ldots, \tilde{f}_\ell)$ can be extracted that do not correspond to previous signature queries. We give a reduction to show that an attacker for which this event has non-negligible probability can be used to construct an algorithm $\mathcal{B}$ that breaks the security of the underlying Weak CMA F-unforgeable signature scheme.
Let Q correspond to the maximum number of signing queries made by $\mathcal{A}$. $\mathcal{B}$ publishes Q random vectors $\boldsymbol{z_1}, \ldots \boldsymbol{z_Q}$ with $\boldsymbol{z_i} \in \mathbb{Z}_p^{\ell+1}$ to the Weak F-Unforgeability CMA challenger and receives Q signatures in return. It sets up the proof system by providing extraction parameters, and uses these signatures to answer the signing queries of $\mathcal{A}$. If $\mathcal{A}$ is successful in producing event **Bad**, $\mathcal{B}$ extracts $\tilde{\sigma}'$ and $(\tilde{z}_0, \ldots \tilde{z}_\ell) \notin \{\boldsymbol{z_1}, \ldots, \boldsymbol{z_Q}\}$ from $\tilde{\pi}$ and outputs it as a valid Weak CMA F-forgery. Consequently we conclude that $\Delta_2(k) \leq Pr[\mathbf{Bad}]$. □

Game 4: check S. This game will proceed as in **Game 2** except that once the adversary outputs his forgery, $M = (M_1, \ldots, M_\ell), \sigma = (S, \pi)$, we let Z be the set of all $(z_0, \ldots, z_\ell)$ tuples used to answer the adversary's queries. Then we consider all tuples in Z, and verify whether $S = g^{z_0} \prod_{j=1}^{\ell} M_j^{z_j}$ for one such tuple. The adversary succeeds if and only if the signature verifies, the message is new, and this check succeeds (i.e. there is such a tuple). Let $p_4(k)$ be the probability that the adversary succeeds in this game.

Lemma 3. *$p_3(k) \leq p_4(k) + \Delta_3(k)$ for some negligible $\Delta_3(k)$ by the soundness of the proof system.*

Proof. If S is computed correctly from the values $(z_0, \ldots, z_\ell)$ corresponding to the values $(F(z_0), \ldots, F(z_\ell))$ extracted from the proof, **Game 4** will be successful in all cases in which **Game 3** is successful. An attacker $\mathcal{A}$ with a non-negligible $\Delta_3(k)$ can thus be used to break the soundness of the proof system, by outputting those proofs for which verification succeeds but extraction fails to obtain valid $(F(z_0), \ldots, F(z_\ell))$ corresponding to S, M. □

Game 5: simulate proofs. In this game, when the public parameters are generated, the challenger will run $\mathsf{SimSetup}$ to generate parameters $params_{pk}$, and trapdoor sim. When responding to signature queries, the challenger chooses random $z_0, \ldots, z_\ell$ and forms S as in the real signing protocol, but generates the proof using $\mathsf{SimProve}$. As above, we judge the adversary's success by verifying the proof and checking the S component of the signature against the set Z of tuples $(z_0, \ldots z_\ell)$ used in previous queries. Let $p_5(k)$ be the probability that the adversary succeeds in this game.

Lemma 4. *$\Delta_4(k) = |p_5(k) - p_4(k)|$ is negligible by the zero-knowledge property of the proof system.*

Proof. An attacker with non-negligible $\Delta_4(k)$ can be used to break the zero-knowledge property of the proof system. We use the standard definition of multi-theorem zero-knowledge. Given an attacker $\mathcal{A}$ with non-negligible $\Delta_4(k)$, we construct an algorithm $\mathcal{B}$ that can distinguish whether, when interacting with a multi-theorem zero-knowledge challenge game, it is given real proofs or simulated proofs. $\mathcal{B}$ sets up the system using the parameters received from the challenge game; to generate each signature, it chooses random $z_0, \ldots, z_\ell$, generates S, σ as in the signing algorithm, and generate the zero-knowledge proof using an oracle query. If $\mathcal{A}$ succeeds in producing S which does not correspond to any of the $z_0, \ldots z_i$ tuples together with a proof π that verifies, then $\mathcal{B}$ outputs 1. □

Lemma 5. *$p_5(k)$ is negligible because S is computed by a pairwise-independent hash function.*

Proof. Suppose we know S and $M = (M_1, \ldots, M_\ell)$ for some unknown $z = (z_0, \ldots, z_\ell)$. Then for any other S', $M' = (M'_1, \ldots, M'_\ell) \neq M$, the probability (taken over possible values of z) that $S' = g^{z_0} \prod_{i=1}^{\ell} {M'}_i^{z_i}$ is $1/p$ by pairwise independence. Thus, for any tuple $z_0, \ldots, z_\ell$ used by the signer, the probability of $\mathcal{A}$ producing a correct pair S', M' for that tuple is at most $1/p$. Taking a union bound over all tuples used gives q/p where q is the total number of queries made by $\mathcal{A}$. This will be negligible since q is polynomial and p is exponential in k.

By the triangle inequality $p_1(k) \leq \Delta_1(k) + \Delta_2(k) + \Delta_3(k) + \Delta_4(k) + p_5(k)$ is negligible. □

3.3 Proving possession of a signature on group elements.

Note that if the initial signature scheme $\mathsf{Sig}_{n \cdot exp}$ has a public key consisting only of elements in $\mathbb{G}$, then we get what Fuchsbauer [Fuc09] refers to as an automorphic signature, a signature scheme that can sign its own verification key. If $\mathbb{G}$ is a group with a bilinear map, and if a signature of $\mathsf{Sig}_{n \cdot exp}$ consists of group elements and can be verified entirely via pairings, then we can use the Groth-Sahai NIZKPK to get a signature composed entirely of group elements. This in turn means that we will be able to efficiently generate a proof of knowledge of a signature using another Groth-Sahai proof system instance with an independent reference string $params_{pf}$.

In a bit more detail: Here we will assume we are given a public key pk, a vector of messages $M = (M_1, \ldots, M_\ell)$, a signature $\sigma = (\pi_\sigma, S)$, and a commitment $C = \mathsf{Com}(M, open)$ with associated opening information $open$. The goal is to generate a zero knowledge proof of knowledge π_π of a signature under pk on the message contained in C. Such a proof is needed for many privacy enhancing protocols, in particular those mentioned in Section A.

We note that for the proof of knowledge to be meaningful, we have to use an additional set of Groth-Sahai parameters $params_{pf}$ that are independent of the $params_{pk}$ in pk. This is necessary, because in order to argue unforgeability for the signature scheme, the trapdoors for $params_{pk}$ need to be unknown. Thus, if we want to be able to extract σ, we need to use a different set of parameters with a different trapdoor.

We consider two options for proving possession of a signature on group elements:

Generic proof of knowledge. In the first approach we treat the signature $\sigma = (\pi_\sigma, S)$ just like any other signature that can be verified using bilinear maps. The downside to this approach is that it will make the proof longer. We have to commit to each group element in the original signature resulting in a proof of knowledge that is larger by at least a factor of 2 for the instantiation based on XDH and 3 for the instantiation based on DLIN.

Using randomization. Alternatively, we can use the randomization properties of Groth-Sahai proofs as defined by [BCC+09] to obtain a more efficient solution. Recall that the signature consists of a group element S and a proof π_σ. Instead of directly forming a proof of knowledge of π_σ and S, we will commit to S and M and randomize the proof π_σ into a proof π'_σ that hides S and M in commitments.

However, we must be careful how we do this: we still want to guarantee that given the proof π_π, we can extract a valid signature. Let $params_{pk}$ be the Groth-Sahai parameters used by π_σ[9]. Note that in order to argue unforgeability for the signature scheme, the trapdoors for $params_{pk}$ need to be unknown. Thus, in order to be able to extract S, M together with a valid signature, we also commit to S and M using $params_{pf}$. The proof π_π guarantees that π'_σ and these commitments to S and M are consistent.

Finally, recall that the original signature contains a proof about M, S, rather than about commitment to M, S. In order to allow us to transform the proof about commitments back into a proof about M, S, we require that the proof π_π includes commitments to a few additional values that allow us to undo parts of the randomization of π'_σ. For this we make use of some convenient features of Groth-Sahai proofs. Further details on this proof can be found in Appendix D.

Proofs that are only unforgeable. In many situations it is not necessary to be able to extract signatures from proofs and it is sufficient to guarantee that an attacker cannot generate a proof for a committed message that has never been signed, i.e. an attacker that produces a forged proof can be used to break DLIN. To do this, we follow the above approach and simply omit the additional values described in the last step. The proof would follow the proof of Theorem 4 and is omitted here.

3.4 Other Instantiations

Many other instantiations of the signature scheme $(\mathsf{SigKg}_{n\cdot exp}, \mathsf{Sign}_{n\cdot exp}, \mathsf{SigVerify}_{n\cdot exp})$ already exist in the literature. Some are based on fairly weak assumptions, while others trade off stronger assumption for better efficiency. For instance one could use Waters or Naccache signatures [Wat05,Nac07], however this doesn't seem to improve much over the scheme presented in Section 3.1 when used in our transformation: because of bit-by-bit proofs such an approach is less efficient and the resulting signature will still rely on DLIN or SXDH. (These signatures are based on CDH which is implied by the DLIN or SXDH assumptions used for the Groth-Sahai proofs.) Alternatively one could make use of the F-unforgeable multi-block P-signature scheme in [BCKL09] that is secure under the q-BB-HSDH and q-TDH assumptions. This would result in a more efficient, but arguably less secure instantiation. A third instantiation can be obtained by using automorphic signatures on message vectors [Fuc09] [10]. Such a construction would be secure under the DHSDH and HDL assumptions, and would again be more efficient at the cost of stronger assumptions. (Note that contrary to the automorphic signatures in [Fuc09] we would no longer be restricted to signing only Diffie-Hellman pairs.)

[9] Here we assume that the parameters $params_{pk}$ used in π_σ are given in a CRS (rather than being generated by the signer).

[10] The automorphic signatures construction in [Fuc09] requires that all messages in the vector be of the form $(\tilde{g}^m, \tilde{h}^m)$. We can easily use this as the Sign_{exp} in our transform by signing $((\tilde{g}^{z_0}, \tilde{h}^{z_0}), \ldots, (\tilde{g}^{z_n}, \tilde{h}^{z_n}))$.

Instantiation	stateless signature	proof of signature possession	unforgeable proof
DLIN	$100 + 24\ell + 9x$	$183 + 111\ell + 9x$	$129 + 57\ell + 9x$
q-BB-HSDH + q-TDH + DLIN	$79 + 7\ell$	$162 + 96\ell$	$108 + 42\ell$
RCDH + SXDH	$77 + 18\ell + 6x$	$124 + 70\ell + 6x$	$92 + 38\ell + 6x$
q-BB-HSDH + q-TDH + SXDH	$61 + 6\ell$	$108 + 58\ell$	$76 + 26\ell$

Table 1. Estimated size in group elements of a signature and a proof for different versions of our transform: ℓ is the number of group elements signed and $N = 2^x$ is an upper bound on the number of signatures generated per key pair.

3.5 Performance Analysis

For the performance analysis we instantiate our signatures and proofs with two signature schemes – the scheme based on RCDH described in Section 3.1 and one based on q-BB-HSDH and q-TDH described in [BCKL09]. We instantiate the Groth-Sahai proofs under DLIN and SXDH. Here ℓ is the number of signatures, and 2^x is the maximum number of signatures issued. Table 1 gives estimates for the size of a signature and a proof of signature possession (expressed in number of group elements). More details concerning the performance analysis can be found in Appendix E. Both the proof of signature possession and the merely unforgeable proof are based on proof randomization.

We note that while our signatures and proofs are still somewhat expensive, they are still within the realm of feasibility (and not much more expensive than the signature scheme used in [BCKL09] for example). The efficiency of our signatures is within an order of magnitude of those in [Fuc09] and [CLY09] which rely on stronger assumptions and satisfy weaker definitions. The recent scheme of [AFG$^+$10] does achieve constant sized signatures (again based on a stronger q-type assumption) which are significantly more efficient than ours. However, we note that when we consider a zero knowledge proof of knowledge of a signature on a committed message (as we would in many of the applications we discuss), the proof will still need to contain commitments to each element of the signature, additional commitments to each of the messages, and proofs that each these commitments is correct. The resulting efficiency will again be within an order of magnitude of the proofs we present here.

4 Conclusion and Open Problems

We construct a reasonably efficient signature scheme for signing group elements based on DLIN, one of the weakest decisional assumptions in the pairing setting (and the weakest one that was used to construct Groth-Sahai proofs). We show that such a signature scheme is an important building block for numerous cryptographic protocols. As our construction does not make use of "q-type" assumptions, it can be used for instantiations of protocols under weaker assumptions for which as of now only instantiations in the random oracle or generic group model were known.

Thus, we see a tradeoff between efficiency and security, and we argue that in many cases sacrificing an order of magnitude in efficiency for a significantly weaker (and non q-type) and more standard assumption may be a reasonable exchange. Furthermore, this result can be seen as evidence that schemes based on relatively weak assumptions can be practical, and as support for the argument that, while they are very important developments, we need not necessarily be satisfied with schemes based on the generic group model, but rather that we should continue looking for schemes which are *both* efficient *and* based on weak assumptions.

References

[ACJT00] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In Mihir Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 255–270. Springer-Verlag, Berlin, Germany, 2000.

[AFG+10] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In Tal Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 209–236. Springer, 2010.

[AGHO11] Masayuki Abe, Jens Gorth, Kristiyan Haralambiev, and Miyako Ohkubo. Optimal structure-preserving signatures in asymmetric bilinear groups. In *CRYPTO*, 2011.

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

[ASM08] Man Ho Au, Willy Susilo, and Yi Mu. Practical anonymous divisible e-cash from bounded accumulators. In Gene Tsudik, editor, *Financial Cryptography*, volume 5143 of *Lecture Notes in Computer Science*, pages 287–301. Springer, 2008.

[AWSM07] Man Ho Au, Qianhong Wu, Willy Susilo, and Yi Mu. Compact e-cash from bounded accumulator. In Masayuki Abe, editor, *CT-RSA*, volume 4377 of *Lecture Notes in Computer Science*, pages 178–195. Springer, 2007.

[BB04] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 54–73. Springer-Verlag, Berlin, Germany, 2004.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures using strong Diffie-Hellman. In *CRYPTO 2004*, volume 3152 of LNCS, pages 41–55, 2004.

[BCC+09] Mira Belenkiy, Jan Camenisch, Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Hovav Shacham. Randomizable proofs and delegatable anonymous credentials. In *Advances in Cryptology - CRYPTO 2009*, Lecture Notes in Computer Science, page 32, Santa Barbara,CA,USA, 2009. Springer-Verlag.

[BCDvdG87] Ernest F. Brickell, David Chaum, Ivan Damgård, and Jeroen van de Graaf. Gradual and verifiable release of a secret. In Carl Pomerance, editor, *CRYPTO*, volume 293 of *Lecture Notes in Computer Science*, pages 156–166. Springer, 1987.

[BCKL08] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. P-signatures and noninteractive anonymous credentials. In Ran Canetti, editor, *Theory of Cryptography Conference (TCC 2008)*, volume 4948 of *Lecture Notes in Computer Science*, pages 356–374, New York,NY,USA, 2008. Springer-Verlag.

[BCKL09] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. Compact e-cash and simulatable VRFs revisited. In *Pairing-Based Cryptography - Pairing 2009*, Lecture Notes in Computer Science, page 27, Palo Alto,CA,USA, 2009. Springer-Verlag.

[BCL04] Endre Bangerter, Jan Camenisch, and Anna Lysyanskaya. A cryptographic framework for the controlled release of certified data. In *Twelfth International Workshop on Security Protocols*, Lecture Notes in Computer Science. Springer Verlag, 2004.

[BF01] Dan Boneh and Matthew Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229, 2001.

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

[Bou00] Fabrice Boudot. Efficient proofs that a committed number lies in an interval. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 431–444. Springer-Verlag, Berlin, Germany, 2000.

[CCS08] Jan Camenisch, Rafik Chaabouni, and Abhi Shelat. Efficient protocols for set membership and range proofs. In Josef Pieprzyk, editor, *ASIACRYPT '08: Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security*, volume 5350 of *Lecture Notes in Computer Science*, pages 234–252, London, UK, 2008. Springer-Verlag.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187, 1994.

[CG08] Sébastien Canard and Aline Gouget. Anonymity in transferable e-cash. In Steven M. Bellovin, Rosario Gennaro, Angelos D. Keromytis, and Moti Yung, editors, *ACNS*, volume 5037 of *Lecture Notes in Computer Science*, pages 207–223, 2008.

[CGH06] Sébastien Canard, Aline Gouget, and Emeline Hufschmitt. A handy multi-coupon system. In Jianying Zhou, Moti Yung, and Feng Bao, editors, *ACNS*, volume 3989 of *Lecture Notes in Computer Science*, pages 66–81, 2006.

[CGH09] Scott E. Coull, Matthew Green, and Susan Hohenberger. Controlling access to an oblivious database using stateful anonymous credentials. In Stanislaw Jarecki and Gene Tsudik, editors, *Public Key Cryptography*, volume 5443 of *Lecture Notes in Computer Science*, pages 501–520. Springer, 2009.

[CHK+06] Jan Camenisch, Susan Hohenberger, Markulf Kohlweiss, Anna Lysyanskaya, and Mira Meyerovich. How to win the clone wars: Efficient periodic n-times anonymous authentication. In Sabrina De Capitani di Vimercati, Vitaly Shmatikov, and Rebecca N. Wright, editors, *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006*, page 11, Alexandria,VA,USA, 2006. ACM.

[CHL05] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact E-Cash. In *EUROCRYPT*, volume 3494 of LNCS, pages 302–321, 2005.

[CHL06] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Balancing accountability and privacy using e-cash. In *SCN*, 2006.

[CL01] Jan Camenisch and Anna Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer-Verlag, Berlin, Germany, 2001.

[CL02] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In *SCN 2002*, volume 2576 of *Lecture Notes in Computer Science*, pages 268–289, 2002.

[CL06] Melissa Chase and Anna Lysyanskaya. On signatures of knowledge. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 78–96, 2006.

[CLM07] Jan Camenisch, Anna Lysyanskaya, and Mira Meyerovich. Endorsed e-cash. In *IEEE Symposium on Security and Privacy*, pages 101–115. IEEE Computer Society, 2007.

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

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *STOC '71: Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, New York, NY, USA, 1971. ACM.

[Cra97] Ronald Cramer. *Modular Design of Secure yet Practical Cryptographic Protocols*. PhD thesis, University of Amsterdam, Amsterdam, 1997.

[Dam02] Ivan Damgård. On σ-protocols. Available at `http://www.daimi.au.dk/~ivan/Sigma.ps`, 2002.

[Dwo06] Cynthia Dwork, editor. *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*. Springer, 2006.

[EG85] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 10–18, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[Fis06] Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Dwork [Dwo06], pages 60–77.

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

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

[Fuc09] Georg Fuchsbauer. Automorphic signatures in bilinear groups. Cryptology ePrint Archive, Report 2009/320, 2009. `http://eprint.iacr.org/`.

[GH08] Matthew Green and Susan Hohenberger. Universally composable adaptive oblivious transfer. In *ASIACRYPT '08: Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security*, pages 179–197, Berlin, Heidelberg, 2008. Springer-Verlag.

[GH10] Matthew Green and Susan Hohenberger. Practical adaptive oblivious transfer from a simple assumption. Cryptology ePrint Archive, Report 2010/109, 2010. `http://eprint.iacr.org/`.

[GOS06a] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive Zaps and new techniques for NIZK. In Dwork [Dwo06], pages 97–111.

[GOS06b] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In *EUROCRYPT*, pages 339–358, 2006.

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

[Gro07] Jens Groth. Fully anonymous group signatures without random oracles. Cryptology ePrint Archive, Report 2007/186, 2007. http://eprint.iacr.org/.

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel Smart, editor, *EUROCRYPT 2008*, 2008.

[Hal09] Shai Halevi, editor. *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*. Springer, 2009.

[HW09a] Susan Hohenberger and Brent Waters. Realizing hash-and-sign signatures under standard assumptions. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 333–350. Springer, 2009.

[HW09b] Susan Hohenberger and Brent Waters. Short and stateless signatures from the rsa assumption. In Halevi [Hal09], pages 654–670.

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

[Nac07] D. Naccache. Secure and practical identity-based encryption. *Information Security, IET*, 1(2):59–64, 2007.

[Nao03] Moni Naor. On cryptographic assumptions and challenges. In Dan Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 96–109. Springer, 2003.

[TS06] Isamu Teranishi and Kazue Sako. k-times anonymous authentication with a constant proving cost. In *Public Key Cryptography*, pages 525–542, 2006.

[Wat05] B. Waters. Efficient identity-based encryption without random oracles. *Advances in Cryptology-Eurocrypt 2005: 24th Annual International Conference on the Theory And Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, 2005.

A Applications

We present four useful applications for our signature scheme related to the construction of privacy-protecting cryptographic protocols. The advantages are twofold: (1) we arrive at constructions that are based on weaker assumptions and (2) the ability to sign group elements can simplify the construction, e.g., in the oblivious transfer protocol or in the case of two round issuing protocols for P-signatures and delegatable anonymous credentials. The applications to automorphic signatures and P-signatures are the most generic, as these schemes can themselves be used to construct even more privacy-protecting protocols.

A.1 Universal Composable Adaptive OT under XDH and DLIN assumption.

A 1-out-of-n oblivious transfer (OT) scheme allows a sender who possesses multiple messages $m_1, \ldots, m_N$ to interact with a receiver that can select the index $0 \leq i \leq N$ in such a way that the receiver can only learn m_i and the sender learns nothing about i. In an adaptive OT, the receiver can adaptively retrieve multiple messages.

Green and Hohenberger [GH08] give a construction for universally composable adaptive oblivious transfer (OT) and prove the security of their construction under the XDH, DLIN, and q-Hidden LRSW assumptions. Their adaptive OT protocol consists of a commit phase in which the receiver obtains encryptions $C_1, \ldots, C_N$ of messages $m_1, \ldots, m_N$ that he can request in the later transfer phases. The OT uses an assisted decryption approach, in which the sender assists the receiver in the decryption of blinded ciphertexts. To make sure that the receiver only asks for the decryption of single ciphertexts, and not for instance for a combination of ciphertexts, the ciphertexts need to be signed. Green and Hohenberger use the signature scheme CLSign based on the q-Hidden LRSW assumption to sign individual group elements. To guarantee that group elements from different ciphertexts are not mixed, they also need to sign the product of two group elements using the signature scheme BBSign based on the co-CDH assumption.

Expressed more formally using the notation of [GH08, §4], at the core of the oblivious transfer request generated by the receiver is the following non-interactive witness indistinguishable Groth-Sahai proof:

$$\pi = \mathsf{NIWIPK}\{(c_1, c_2, t_1, t_2, sig_1, sig_2, sig_3) : e(c_1, \tilde{h})e(t_1, \tilde{u}1) = e(d_1, \tilde{h}) \wedge e(c_2, \tilde{h})e(t_2, \tilde{u}_2) = e(d_2, \tilde{h}) \wedge \mathsf{CLVerify}_{vk_1}(c_1, sig_1) = 1 \wedge \mathsf{CLVerify}_{vk_2}(c_2, sig_2) = 1 \wedge \mathsf{BBVerify}_{vk_3}(c_1c_2, sig_3) = 1\}$$

At this step the receiver proves that for some encryption C_i parsed as $(c_1, \ldots, c_5, sig_1, sig_2, sig_3)$, the ciphertext components c_1 and c_2 are correctly blinded in d_1 and d_2.

By applying our construction, their protocol can be significantly simplified: instead of the three signatures (on the two ciphertext components and their product), it suffices to compute a single signature on two group elements.

$$\pi = \mathsf{NIWIPK}\{(c_1, c_2, t_1, t_2, sig) : e(c_1, \tilde{h})e(t_1, \tilde{u}_1) = e(d_1, \tilde{h}) \wedge e(c_2, \tilde{h})e(t_2, \tilde{u}_2) = e(d_2, \tilde{h}) \wedge \mathsf{SigVerify}(vk, (c_1, c_2), sig) = 1\}$$

Using our new proposal for a Weak CMA F-unforgeable signature scheme and Groth-Sahai proofs with a DLIN based setup, gives rise to the first universally composable adaptive oblivious transfer scheme based purely on the DLIN assumption, presumably the weakest possible assumption if one wants to use the Groth-Sahai proof system.

Independently from our work [GH10] Green and Hohenberger proposed another oblivious transfer scheme that makes use of the signature scheme of Hohenberger and Waters [HW09a]. However, their scheme has to make use of a different encryption scheme and is not UC-secure.

A.2 Automorphic Signatures in Bilinear Group

Fuchsbauer [Fuc09] introduced *automorphic signatures* in bilinear groups, which have to satisfy the following properties: the verification keys lie in the message space, messages and signatures consist of group elements only, and verification is done by evaluating a set of pairing-product equations. Clearly, our transformation, when instantiated with the signature scheme from Section 3.1 and the Groth-Sahai proof system, results in an automorphic signature. As such it is also suitable for the applications mentioned in [Fuc09]: fully-secure group signatures and anonymous proxy signatures. Note that in addition to removing any restrictions on the message space, our construction can be based on much weaker assumptions, if one is willing to trade off some performance for better security.

A.3 CL and P-signatures

CL-signatures consist of two protocols, (a) an issuing protocol for obtaining a signature on messages that are hidden from the signer in commitments $\boldsymbol{C}$, and (b) a proof protocol for proving possession of a signature on messages that are hidden from the verifier in fresh commitments $\boldsymbol{C'}$. To overcome a weakness of CL-signatures, namely that their proof protocol makes use of Σ-protocols and thus requires a random oracle to obtain a non-interactive zero-knowledge proof, Belenkiy et al. [BCKL08] introduce P-signatures, signatures with an efficient non-interactive **P**roof of knowledge.

Existing CL and P-signature schemes only allow one to sign elements of $\mathbb{Z}_p$ or $\mathbb{Z}_n$. Moreover they are based on much strong assumptions that appear to be stronger than the assumptions required by the commitment scheme: for CL-signatures all known constructions are either based on the strong RSA assumptions or on different "q-type" Diffie-Hellman assumptions. (A well known example of the latter type of assumption is q-SDH [BB04].) For P-signatures [BCKL08] known constructions are based on even stronger assumptions, such as for instance q-BB-HSDH.

The signature schemes and proof techniques of Section 3 can be used to build a P-signature scheme (and thus also a CL-signature scheme) for signing group elements that is based on the DLIN assumption (the same assumption underlying the commitment scheme). The issuing protocol for signing group elements is a two round protocol, in which the user first commits to the group element that should be signed using a DLIN commitment C. The commitment parameters can be based on the $params_{pf}$ of the proof system. Note that such a commitment consists of three group elements C_1, C_2, C_3. The signature on a committed message is a signature on these three group elements. When proving possession of a signature on the message now committed to in a new DLIN commitment C', the NIZKPK of Section 3.3 needs to be extended with a proof that C' commits to the same message as the (now secret, because part of the proof of knowledge) commitment $C = (C_1, C_2, C_3)$:

$$\pi' \in \mathsf{NIZKPK}\{(\sigma, C_1, C_2, C_3, M) : \{\mathsf{SigVerify}(pk, (C_1, C_2, C_3), \sigma) = 1 \wedge \\ \big(\exists open_1, open_2 : (C_1, C_2, C_3) = \mathsf{Com}(M, open_1) \wedge C' = \mathsf{Com}(M, open_2)\big)\}$$

This approach can be extended to P-signatures that allow one to sign multiple group elements.

Although the details of the construction are beyond the scope of this paper, it should be evident that these protocols are a useful tool both for reproving many privacy enhancing protocols under weaker assumptions and for building new privacy-enhancing protocols.

A note on the issuing of signatures.

As our signature scheme allows to sign group elements, for many applications such as group signatures and the delegatable anonymous credential scheme described below, the interactive issuing of signatures on exponents hidden in commitments is no longer needed. Instead, one can simply sign the public key of the user or a commitment to the users secret key, i.e., a pseudonym. Thus, we get a simple two round issuing protocol.

As pointed out by Kiayias and Yung [KY05] for the case of group signature joins, two-move signature issuing protocols with a simple "single message and signature response" interaction between the prospective user and the issuer are the most desirable as they can improve the security of many schemes with respect to concurrent attacks and enable issuing over the Internet (where servers are multi-thread machines). Kiayias and Yung also point out the following application scenario for adding users via a proxy. A security officer of a company can send a file with all registration requests in his company, get back their certificates, and distribute them back to company employees. As described for P-signatures, if one wants to sign group elements without revealing them to the issuer, one can hide them in a commitment and obtain a signature on the commitment.

A.4 Delegatable anonymous credentials

Belenkiy et al. [BCC+09] give an efficient construction for delegatable anonymous credentials. Credentials are delegated similarly to public key certificates that form a certification hierarchy: the owner of a certificate can extend the certification chain by creating a certificate for the next level. In a delegatable anonymous credential system, parties only know each other under unlinkable pseudonyms, and can receive, delegate, and show credentials anonymously using different pseudonyms.

The original construction of delegatable credentials uses a complex signature scheme based on the q-BB-HSDH and q-BB-CDH assumption. They cannot make use of existing P-signatures as this signature scheme requires an additional property called certification security, i.e. it needs to remain secure, even if the attacker can learn signatures on the secret key of honest users. Using signatures for signing group elements, one can sign the pseudonym of a user directly, and

thus avoid this problem. Note that in order to change the pseudonym between obtain and issue transactions, one needs to prove that the hidden pseudonym $Nym = (C_1, C_2, C_3)$, whose three components are signed using our signature scheme, and the pseudonym Nym' that is revealed to the communication partner commit to the same $g^{sk_{\mathcal{U}}}$ value. This can be done as described in Section A.3 for commitments C and C'.

Note that using a group element signature based on the construction in Section 3.1, one obtains a delegatable anonymous credential scheme secure under the DLIN assumption. If we use the straightforward technique for proving knowledge of a signature (just treat the signature and it's verification as a witness that can be verified via pairing product equation), then randomization can be applied directly, exactly as in the [BCC$^+$09] construction. This can also be done using the more efficient randomization based proof of knowledge, however the process is significantly more complex, so we leave the details as a topic for future work.

B The RCDH assumption

In section 3.1 we introduced a new assumption call Randomized CDH (RCDH). However, we can show that this assumption is implied by the DLIN assumption.

The assumptions is as follows:

Assumption 2 (Randomized Computational Diffie-Hellman) *Let $\mathbb{G}$ be a group of prime order $p \in \Theta(2^k)$. For all p.p.t. adversaries $\mathcal{A}$, the following probability is negligible in k:*

$$Pr[g, \hat{g} \leftarrow \mathbb{G}; a, b \leftarrow \mathbb{Z}_p; R_1, R_2, R_3, \leftarrow \mathcal{A}(g, \hat{g}, g^a, g^b) : \exists r \in \mathbb{Z}_p \text{ such that } R_1 = g^r, R_2 = \hat{g}^r, R_3 = g^{abr}]$$

Theorem 6. *RCDH is implied by DLIN.*

Proof. Suppose we are given groups $\mathbb{G}, \mathbb{G}_T$, and a DLIN instance g, f, R, h, S, T where $R = f^r, S = h^s$, for random $f, h \in \mathbb{G}$ and random $r, s \in \mathbb{Z}_p$, and we must determine whether $T = g^{r+s}$ or $T = g^t$ for random t. We assume that there exists an adversary $\mathcal{A}$ who succeeds in the RCDH game with non-negligible probability. Then we proceed as follows:

First we choose random $\gamma, \delta \leftarrow \mathbb{Z}_p$ and run $\mathcal{A}$ on input f, g^γ, g^δ, R. We receive Z_1, Z_2, Z_3, and with non-negligible probability these values will be such that there exists z such that $Z_1 = f^z$, $Z_2 = g^{\gamma z}$, $Z_3 = g^{\delta r z}$. From these we will compute $Z'_1 = Z_1, Z'_2 = Z_2^{1/\gamma}$, and $Z'_3 = Z_3^{1/\delta}$, which, if the adversary's output was correct, will be f^z, g^z, g^{rz}.

Next we choose random $\alpha, \beta \leftarrow \mathbb{Z}_p$ and run $\mathcal{A}$ on input $h, (Z'_3)^\alpha, (Z'_2)^\beta, S$. We receive Y_1, Y_2, Y_3, and with non-negligible probability these values will be such that there exists y such that $Y_1 = h^y$, $Y_2 = (Z'_3)^{\alpha y}$, $Y_3 = (Z'_2)^{\beta s y}$. From these we will compute $Y'_1 = Y_1, Y'_2 = Y_2^{1/\alpha}$, and $Y'_3 = Y_3^{1/\beta}$, which, if the adversary's output was correct in both runs, will be h^y, g^{rzy}, g^{szy}.

Next we choose random $\rho_1, \rho_2, \rho_3, \rho_4 \leftarrow \mathbb{Z}_p$ and run $\mathcal{A}$ on input $g^{\rho_1}, h^{\rho_2}, (Y'_1)^{\rho_3}, Z_2'^{\rho_4}$. We receive X_1, X_2, X_3, and with non-negligible probability these values will be such that there exists x such that $X_1 = g^{\rho_1 x}$, $X_2 = h^{\rho_2 x}$, $X_3 = g^{ab\rho_3\rho_4/\rho_1^2}$, where $Y'_1 = g^a$ and $Z'_2 = g^b$. From these we will compute $X'_1 = (X_1)^{1/\rho_1}, X'_2 = X_2^{1/\rho_2}$, and $X'_3 = X_3^{\rho_1^2/(\rho_3\rho_4)}$, which, if the adversary's output was correct in all 3 runs, will be g^x, h^x, h^{xzy}.

Finally, we compute $e(X'_3, T)$ and $e(Y'_2 Y'_3, X'_2)$. If the adversary's responses are correct, this corresponds to computing $e(h^{xzy}, T)$ and $e(g^{zy(r+s)}, h^x)$. If the resulting values are equal, we will output 1, otherwise we output random $b \leftarrow \{0, 1\}$. Thus, if the adversary succeeds in all 3 queries with non-negligible probability, we will distinguish with non-negligible advantage. Since the queries involve random independent instances of RCDH, this will be true as long as the adversary succeeds in the RCDH game with non-negligible advantage.

C Pairwise independence of f_z.

The family of hash-functions $\{f_z\}$, where $f_z : \mathbb{G}^\ell \to \mathbb{G}$ is computed as $f_z(M_1, \ldots, M_\ell) = g^{z_0} \prod_{i=1..\ell} M_i^{z_i}$ with $z = (z_0, \ldots, z_\ell) \in \mathbb{Z}_p^{n+1}$ is pairwise independent.

Proof. We first rewrite the probability of the definition in the following form

$$Pr[z \leftarrow \mathcal{Z} : f_z(x) = a \wedge f_z(y) = b] = Pr[z \leftarrow \mathcal{Z} : f_z(x) = a] \cdot Pr[z \leftarrow \mathcal{Z} : f_z(y) = b \mid f_z(x) = a]$$

We consider the two probabilities of the product separately:

1. One can see that $Pr[z \leftarrow \mathcal{Z} : f_z(x) = a] = 1/|\mathbb{G}|$. The values $z \in \mathbb{Z}_p^{n+1}$ are chosen uniformly, while $x \in \mathbb{G}^\ell$ and $a \in \mathbb{G}$ are fixed, so $g^{z_0} \prod_{i=1..\ell} x_i^{z_i}$ is uniformly distributed as well.
2. In $Pr[z \leftarrow \mathcal{Z} : f_z(y) = b \mid f_z(x) = a]$ the event $f_z(y) = b$ is considered only under the condition $f_z(x) = a$. When spelling out z as $z_0, \ldots, z_n$ and $F_z(x)$ as well as $F_z(y)$ as $g^{z_0} \prod_{i=1..\ell} x_i^{z_i}$ and $g^{z_0} \prod_{i=1..\ell} y_i^{z_i}$ respectively, one can replace g^{z_0} in $g^{z_0} \prod_{i=1..\ell} y_i^{z_i}$ by $\frac{a}{\prod_{i=1..\ell} x_i^{z_i}}$.

$$\begin{aligned} & Pr[z \leftarrow \mathcal{Z} : f_z(y) = b \mid f_z(x) = a] \\ = Pr[z_0, \ldots, z_n \leftarrow \mathbb{Z}_p^{n+1} : & \frac{a}{\prod_{i=1..\ell} x_i^{z_i}} \prod_{i=1..\ell} y_i^{z_i} = b \mid g^{z_0} = \frac{a}{\prod_{i=1..\ell} x_i^{z_i}}] \\ = Pr[z_0, \ldots, z_n \leftarrow \mathbb{Z}_p^{n+1} : & \prod_{i=1..\ell} (y_i/x_i)^{z_i} = b/a \mid g^{z_0} = \frac{a}{\prod_{i=1..\ell} x_i^{z_i}}] \end{aligned}$$

As z_0 is picked uniformly at random, the event $g^{z_0} = \frac{a}{\prod_{i=1..\ell} x_i^{z_i}}$ is completely irrelevant to $\prod_{i=1..\ell}(y_i/x_i)^{z_i} = b/a$. Consequently $Pr[z_0, \ldots, z_n \leftarrow \mathbb{Z}_p^{n+1} : \prod_{i=1..\ell}(y_i/x_i)^{z_i} = b/a \mid g^{z_0} = \frac{a}{\prod_{i=1..\ell} x_i^{z_i}}] = Pr[z_1, \ldots, z_n \leftarrow \mathbb{Z}_p^n : \prod_{i=1..\ell}(y_i/x_i)^{z_i} = b/a]$
To prove our claim we need to show that $Pr[z_1, \ldots, z_n \leftarrow \mathbb{Z}_p^n : \prod_{i=1..\ell}(y_i/x_i)^{z_i} = b/a] = 1/|\mathbb{G}|$. As $x \neq y$, there is at least one pair $x_j \neq y_j$, and $y_j/x_j \neq 1$.
As z_j is chosen uniformly at random, $\prod_{i=1..\ell}(y_i/x_i)^{z_i}$ is also uniformly distributed, no matter how the other y_i and x_i are chosen, and thus $Pr[z_1, \ldots, z_n \leftarrow \mathbb{Z}_p^n : \prod_{i=1..\ell}(y_i/x_i)^{z_i} = b/a] = 1/|\mathbb{G}|$.

D Details for Proof of Possession of a Signature on Group Elements Based on Randomization

More formally, the proof π_π is computed as follows: (Here we describe the proof for a single message, but the generalization to signatures on many messages is straightforward.)

As described above, the prover gets as input $pk, M, \sigma = (S, \pi_\sigma), C = \mathsf{Com}(params_{pf}, M, open)$, $open$ where $\mathsf{VerifyProof}(pk, M, \sigma) = 1$, and wishes to generate a proof of knowledge π_π of a valid message and signature corresponding to pk and C.

1. Note that the proof π_σ is formed under parameters $params_{pk}$ and includes M and S as constants. The prover will choose random openings $open_M$ and $open_S$ and randomize the proof π_σ to obtain a random-looking proof pi'_σ for commitments $C'_M = \mathsf{Com}(params_{pk}, M, open_M)$ and $C'_S = \mathsf{Com}(params_{pk}, S, open_S)$. When instantiated with Groth-Sahai proofs under the DLIN assumption[11], this means choosing random values r_{M1}, r_{M2}, r_{M3} and r_{S1}, r_{S2}, r_{S3} and computing $C'_M = M \prod u_i^{r_{Mi}}$ and $C'_S = S \prod u_i^{r_{Si}}$, where u_1, u_2, u_3 are defined in the parameters $params_{pk}$.

[11] This approach also works for SXDH, but we focus on DLIN here for simplicity.

2. The prover will generate commitments under $params_{pf}$ to $M, S, open_M, open_S$. More specifically, he will form commitments $C_M = \mathsf{Com}(params_{pf}, M)$, $C_S = \mathsf{Com}(params_{pf}, S)$, and for $i = 1 \ldots 3$: $C_{openMi} = \mathsf{Com}(params_{pf}, g^{r_{Mi}})$, $C_{openSi} = \mathsf{Com}(params_{pf}, g^{r_{Si}})$. It will then use three[12] proofs $\pi_{CM1}, \pi_{CM2}, \pi_{CM3}$ to show that C'_M is correct according to these new commitments, and corresponding proofs $\pi_{CS1}, \pi_{CS2}, \pi_{CS3}$.
3. The prover will use $params_{pf}$ to generate a zero knowledge proof π_M that C_M and C commit to the same value.
4. Finally, the prover must also include the information necessary for undoing the randomization used to obtain C'_M and C'_S. Note that the Groth-Sahai proof in π_σ that directly affects S will be a zero knowledge proof that some internal commitment $D_S = \mathsf{Com}(params_{pk}, S)$. This proof will be of the form ψ_1, ψ_2, ψ_3 such that $e(D_S/S, D_0) = \prod e(u_i, \psi_i)$ where D_0 is a known commitment to 0. The randomization process, as described above, will compute $C'_S = S \prod u_i^{r_{Si}}$. It will also compute updated proof values $\psi'_i = D_0^{r_{Si}} \psi_i$. If we keep track of these values $D_0^{r_{Si}}$, we can convert this proof back into a proof for S. Thus, we will also include commitments to these values, and proofs that they are correctly formed with respect to D_0 and C_{openSi}. Similarly, we must include commitments to update values for M. Call the resulting lists of commitments $\bar{C}_{updateS}$ and $\bar{C}_{updateM}$ and the associated proofs $\bar{\pi}_{updateS}$ and $\bar{\pi}_{updateM}$. (Note that each of these lists will have 9 values, since we have 3 values $r_S i$ and D_0 is composed of 3 elements. Similarly for the lists for M.)
5. The final proof will be (i) the randomized commitments and proofs C'_S, C'_M, π' under parameters $params_{pk}$, (ii) the commitments $C_M, C_S, \{C_{openSi}\}, \{C_{openMi}\}$ under parameters $params_{pf}$ and associated proofs $\pi_{CM1}, \pi_{CM2}, \pi_{CM3}, \pi_{CS1}, \pi_{CS2}, \pi_{CS3}$, (iii) zero knowledge proof π_M under parameters $params_{pf}$, and (iv) commitments $\bar{C}_{updateM}, \bar{C}_{updateS}$ under $params_{pf}$ and associated proofs $\bar{\pi}_{updateM}, \bar{\pi}_{updateS}$.

Theorem 7. *The above construction yields a zero knowledge proof of knowledge of a signature on the given commitment.*

Proof. (sketch) The proof of knowledge property is fairly straightforward. Given a proof π_π, we extract M from C_M and S from C_S, and extract the update values for both. We use the latter values to undo the randomization applied to the proof π_σ; call the result $\hat{\pi}_\sigma$. Then by the perfect soundness of the Groth-Sahai proof system, $\sigma = (\hat{\pi}_\sigma, S)$ will be a valid signature for message M.

The proof of zero knowledge proceeds as follows: We define a simulator which, given pk and C, proceeds as follows: Choose random M' and S'. Generate commitments C'_M and C'_S under $params_{pk}$ to these values and use the zero knowledge simulator to produce a simulated proof π_sigma under $params_{pk}$ that these commitments are correct. Then generated commitments C_M and C_S, $\{C_{openSi}\}, \{C_{openMi}\}$, and $\bar{C}_{updateM}, \bar{C}_{updateS}$, and associated proofs $\pi_{CM1}, \pi_{CM2}, \pi_{CM3}$, $\pi_{CS1}, \pi_{CS2}, \pi_{CS3}, \bar{\pi}_{updateM}, \bar{\pi}_{updateS}$ honestly according to M', S'. Finally, use the zero knowledge simulator to generate the proof π_M. We can argue that the result will be indistinguishable from the real proof by the perfect zero knowledge, witness indistinguishability and randomizability properties of Groth-Sahai proofs.

E Efficiency

We analyze the efficiency of our group element signature scheme when implemented with GS proofs. Essentially, a signature will consist of:

– The value S

[12] Recall that in the DLIN instantiation of the Groth-Sahai proof system, each u_i is a tuple of 3 group elements, and multiplication is component-wise, so we must show that each component of C'_M is correct.

- A GS commitment to S
- A zero knowledge proof that this commitment is correct.
- A GS commitment to the public key for Sign_{exp}.
- A zero knowledge proof that the public key is correct
- Commitments to $F(z_0), \ldots, F(z_\ell)$
- A GS proof that these values are well formed.
- Commitments to σ the signature on $z_0, \ldots z_\ell$.
- A GS proof of correctness for σ w.r.t committed z_i's and public key.
- A GS proof of correctness for the committed S w.r.t. the committed z_i's.

For a given scheme Sign_{exp}:

- Let n_f be the number of group elements in $F(m)$.
- Let p_f be the number of group elements in the proof of correctness for each value $F(z_0)$.
- Let v_f be the number of pairings for verification of this proof.
- Let n_ℓ be the number of group elements in a signature on $\ell + 1$ values.
- Let p_ℓ be the number of group elements in the proof of correctness for the signature on $\ell + 1$ values.
- Let v_ℓ be the number of pairings for verification of this proof.
- Let n_{pk} be the number of elements in the public key.

For a given instantiation of Groth-Sahai proofs:

- Let c_{lin} be the cost of linear pairing product proof
- Let c_{mult} be the cost of multi-exponentiation proof
- Let c_{quadeq} be the cost of quadratic equation proof
- Let c_C be the cost of commitment
- Let c_{pair} be the cost of product pairing equation proof
- Let c_{zk} be the cost of zk proof

Then, our signature will need:

- the element S
- $1 + n_{pk} + (\ell + 1)n_f + n_\ell$ commitments $((1 + n_{pk} + (\ell + 1)n_f + n_\ell) * c_C$ elements.
- One proof of correctness for the signature (p_ℓ elements).
- $\ell + 1$ proofs of correctness for all $F(z_i)$ values. $((\ell + 1) * p_f)$
- One multi-exponentiation proof (to prove correctness of S) c_{mult} elements
- $1 + n_{pk}$ zero knowledge proofs for opening of a commitment $(1 + n_{pk}) * c_{zk}$ elements.
- One zero knowledge setup commitment and proof. c_{setup} elements.

Verification will require

- verifying $\ell + 1$ proofs of correctness for all $F(z_i)$ values: ($v_f * (\ell + 1)$ pairings in $v_f * (\ell + 1)$ equations
- verifying one multi-exponentiation proof with $\ell + 2$ variables (to prove correctness of S) $v_{mult}(\ell + 2)$ pairings in one equation.
- verifying $1 + n_{pk}$ zero knowledge proofs for opening of a commitment $(1 + n_{pk})v_{zk}$ pairings in $(1 + n_{pk})$ equations.
- verifying one zero knowledge setup commitment and proof. v_{setup} pairings in 1 equation.

To transform this into a proof by the first method we get one commitment for each group element in the signature and one for each message, and one pairing product equation for each pairing product required by signature verification. That yields a total of $v_f*(\ell+1)+1+(1+n_{pk})+1$ pairing product proofs, and $1+(1+n_{pk}+(\ell+1)n_f+n_\ell*c_C)+p_\ell+(\ell+1)*p_f+c_{mult}+(1+n_{pk})*c_{zk}+c_{setup}$ commitments. The total number of pairings is increased by a factor of at most $v_{pair}(1)$. To transform this into a proof by the randomization method, we do the following:

- Include the signature, but randomize the proof to put M and S in commitments: this will add $((\ell+1)*c_C-1)$ to the size of the signature.
- add commitments under $params_{pf}$ to $M, S, open_M, open_S$: $(\ell+1)+(\ell+1)c_C$ commitments for a total of $(\ell+1)(1+c_C)*c_C$ elements.
- add proofs $\pi_{CM1}, \pi_{CM2}, \pi_{CM3}$ and $\pi_{CS1}, \pi_{CS2}, \pi_{CS3}$: $c_C(\ell+1)$ linear pairing product proofs for a total of $c_C(\ell+1)c_{lin}$ elements.
- add one zero knowledge proof of equality for each C_M, C, for a total of $\ell * C_{zk} + C_{setup}$ elements.
- add commitments to update values for S and M, one for each value in each pairing product proof: $(\ell+1)*c_{pair}$ commitments for a total of $(\ell+1)*c_{pair}*c_C$ elements.
- add corresponding proofs: $(\ell+1)*c_{pair}$ linear proofs for a total of $(\ell+1)*c_{pair}*c_{lin}$ elements.

F Summary of Groth-Sahai Proofs.

F.1 Linear and ElGamal commitments

Groth-Sahai commitments [GS08] are commitments that behave advantageously when used together with a bilinear map. To commit to a group element x of a prime order group $\mathbb{G}_1$ of size p, the committing party computes a vector of I group elements. We use multiplicative notation, thus two vectors can be multiplied, and individual vectors are scaled using component-wise exponentiation with an element in $\mathbb{Z}_p$. In a first step x is mapped to a vector through an injective function μ. This can for instance be done by mapping the group element to one component of the vector, and setting all other components to the neutral element 1. Let I be the dimension of the vector space $\mathbb{V}_1$. In order to hide the committed element x, the resulting vector is combined with a random linear combination of vectors $u_i \in \mathbb{V}_1$, $1 \le i \le I$.

To commit to an element $x \in \mathbb{G}_1$, choose random opening $open = (r_1, \ldots, r_I) \leftarrow \mathbb{Z}_p^I$, and compute $C = \mu_1(x) \cdot \prod_{i=1}^{I} u_i^{r_i}$. Elements $y \in \mathbb{G}_2$ are committed to in the same way using μ_2 and $v_1, \ldots, v_J \in \mathbb{V}_2$, and an opening vector $open \in R^J$. For simplicity we assume that $\mathsf{GSCommit}(params_{PK}, m, open)$ first determines whether $m \in \mathbb{G}_1$ or $m \in \mathbb{G}_2$ and then follows the appropriate instructions. The same commitment scheme can be used to commit to a value $m \in \mathbb{Z}_p$ using a group element h as the base.

If the subspace generated by the vectors u_i and the range of μ share only the 1 vector, the commitment scheme is *perfectly binding*. Clearly, this requires that the vectors u_i are not all linearly independent. For the commitment scheme to be *strongly computationally hiding*, the vectors u_i generated by $\mathsf{ComSetup}$ need to be computationally indistinguishable from the linearly independent vectors output by $\mathsf{HidingSetup}$. A random combination of linearly independent vectors u_i, $1 \le i \le I$ generates the whole of $\mathbb{V}_1$ and hides the value x perfectly.

The property that makes GS commitments so useful for the construction of non-interactive proofs [GS08], is that they allow for the evaluation of a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ on committed elements in the committed domain. Given a commitment to a and a commitment to b it is possible to compute a vector of elements in $\mathbb{G}_T$ using a map $E : \mathbb{V}_1 \times \mathbb{V}_2 \to \mathbb{V}_T$, that acts as a commitment to the value $e(a, b)$. Intuitively, if the subspace generated by the vectors $E(u_i, v_j)$ is orthogonal to $E(\mu_1(\mathbb{G}_1), \mu_2(\mathbb{G}_2))$, then the resulting commitment scheme is *perfectly binding*. Moreover the commitment is *strongly computationally hiding* if the commitments to a and b are *strongly computationally hiding*.

We instantiate this approach with commitment schemes that are *perfectly binding* and *strongly computationally hiding* under the SXDH and DLIN assumption (see Section B). We also show that based on appropriate parameters and trapdoor information the resulting commitments have the *extraction* property for $\mathsf{GSCommit}$.

ElGamal commitments (SXDH instantiation). In the SXDH setting, one commits to elements in $\mathbb{G}_1$ as follows (committing to elements in $\mathbb{G}_2$ is similar):

Let vector space $\mathbb{V}_1 = \mathbb{G}_1 \times \mathbb{G}_1$. The parameters are generated by choosing random s, z and computing $u_1 = (g, g^z)$ and $u_2 = (g^s, g^{sz})$. The public parameters are u_1, u_2. If extraction is necessary, the trapdoor will be s, z.

One commits to $x \in \mathbb{G}_1$ by choosing random $r_1, r_2 \in Z_p$ and computing $(1, x)u_1^{r_1}u_2^{r_2}$. The commitment is opened by revealing x, r_1, r_2. Given the trapdoor s, z, it is possible to extract x from a commitment (c_1, c_2) by computing c_2/c_1^z. The commitment is perfectly binding and extractable.

The commitment scheme is *strongly computationally hiding*. Perfectly hiding parameters are generated by choosing random $s, z, w \in Z_p$ and computing $u_1 = (g, g^z)$ and $u_2 = (g^s, g^w)$. The public parameters are u_1, u_2. Note that these public parameters will be indistinguishable from those described above under the SXDH assumption and that under these parameters the commitment scheme is perfectly hiding.

Under this setup the commitment scheme is a *chameleon* commitment. We can form commitments for which we can use the trapdoor s, z, w to open the commitment to any value for which we know the discrete logarithm. We compute such a commitment by choosing random $c_1, c_2 \in Z_p$ and computing (g^{c_1}, g^{c_2}). To open this commitment to any value g^ϕ, we need only find a solution (r_1, r_2) to the equations $c_1 = r_1 + sr_2$ and $c_2 = \phi + zr_1 + wr_2$.

Linear commitments (DLIN instantiation). In the DLIN setting one commits to elements in $\mathbb{G}_1$ as follows (committing to elements in $\mathbb{G}_2$ is similar):

Let vector space $\mathbb{V}_1 = \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1$. The parameters are generated by choosing random a, b, z, s and computing $u_1 = (g^a, 1, g)$ and $u_2 = (1, g^b, g)$, and $u_3 = (g^{az}, g^{bs}, g^{z+s})$. The public parameters are u_1, u_2, u_3. If extraction is necessary, the extraction trapdoor will be a, b, z, s.

One commits to $x \in \mathbb{G}_1$ by choosing random $r_1, r_2, r_3 \in Z_p$ and computing $(1, 1, x)u_1^{r_1}u_2^{r_2}u_3^{r_3}$. Opening would reveal x, r_1, r_2, r_3. In this case, given the trapdoor a, b, s, z, we will be able to extract x from a commitment (c_1, c_2, c_3) by computing $c_3/(c_1^{1/a}c_2^{1/b})$. The commitment is *perfectly binding* and *extractable*.

The commitment scheme is *strongly computationally hiding*. Perfectly hiding parameters are generated by choosing random $a, b, s, z, w \in Z_p$ and computing $u_1 = (g^a, 1, g)$ and $u_2 = (1, g^b, g)$ and $u_3 = (g^{az}, g^{bs}, g^w)$. The public parameters will be u_1, u_2, and u_3. Note that these public parameters will be indistinguishable from those described above under the DLIN assumption and that the resulting commitment scheme is perfectly hiding.

Under this setup the commitment scheme is a *chameleon* commitment. We can form commitments for which we can use the chameleon trapdoor a, b, s, z to open to any value for which we know the discrete logarithm. We compute such a commitment by choosing random $c_1, c_2, c_3 \in Z_p$ and computing $(g^{c_1}, g^{c_2}, g^{c_3})$. To open this commitment to any value g^ϕ, we need only find a solution (r_1, r_2, r_3) to the equations $c_1 = ar_1 + azr_3$, $c_2 = br_2 + bsr_3$ and $c_3 = \phi + r_1 + r_2 + (z+s)r_3$.

F.2 Computing a GS Proof

Let $params_{BM} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h)$ be the setup for pairing groups of prime order p, with pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, and g, h generators of $\mathbb{G}_1, \mathbb{G}_2$ respectively.[13]

The GS proof instance consists of the coefficients of the pairing product equation:

$$\{a_q\}_{q=1\ldots Q} \in \mathbb{G}_1, \{b_q\}_{q=1\ldots Q} \in \mathbb{G}_2, t \in \mathbb{G}_T,$$
$$\{\alpha_{q,m}\}_{q=1\ldots Q, m=1\ldots M}, \{\beta_{q,n}\}_{q=1\ldots Q, n=1\ldots N} \in \mathbb{Z}_p \ .$$

[13] There is also an instantiation for composite order groups, but we will not consider it here.

The prover knows a set of values $\{x_m\}_{m=1}^{M}$, $\{z_n\}_{n=1}^{N}$ that satisfy the pairing product equation

$$\prod_{q=1}^{Q} e(a_q \prod_{m=1}^{M} x_m^{\alpha_{q,m}}, b_q \prod_{n=1}^{N} z_n^{\beta_{q,n}}) = t \; .$$

As the first step in creating the proof, the prover prepares commitments $\{C_m\}_{m=1\ldots M}$ and $\{D_n\}_{n=1\ldots N}$ for all values x_m, z_n in $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. Alternatively, it is possible to reuse commitments from the proof instance. Thus, the instance, known to the prover and verifier, is the pairing product equation (e.g., its coefficients) and a number of commitments while the witness, known only to the prover, consists of the secret values and the *openings* of these commitments.

We now describe how to construct the proof. Let $\mathbb{V}_1$, $\mathbb{V}_2$ be the vector spaces underlying two GS commitment schemes for committing to elements in $\mathbb{G}_1$ and $\mathbb{G}_2$, and let $E\colon \mathbb{V}_1 \times \mathbb{V}_2 \to \mathbb{V}_T$ be a bilinear map that evaluates the bilinear map e in the committed domain. Also let μ_1, μ_2, μ_T be efficiently computable embeddings that map elements of $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ into $\mathbb{V}_1, \mathbb{V}_2, \mathbb{V}_T$, respectively. Note that by the properties of E, $E(\mu_1(a), \mu_2(b)) = \mu_T(e(a,b))$. The public parameters $params_{PK}$ contain a common reference string with elements $u_1, \ldots, u_I \in \mathbb{V}_1$, $v_1, \ldots, v_J \in \mathbb{V}_2$ and values $\eta_{h,i,j}$, $1 \le i \le I$, $1 \le j \le J$, and $1 \le h \le H$ as defined by Groth and Sahai [GS08].

Groth and Sahai show how to efficiently compute proofs $\{\pi_i\}_{i=1}^{I}$, $\{\psi_j\}_{j=1}^{J}$ that prove that values in C_m and D_n satisfy a pairing product equation. To verify such a proof the verifier computes, for all $1 \le q \le Q$, $\hat{C}_q \leftarrow \mu_1(a_q) \cdot \prod_{m=1}^{M} C_m^{\alpha_{q,m}}$ and $\hat{D}_q \leftarrow \mu_2(b_q) \cdot \prod_{n=1}^{N} D_n^{\beta_{q,n}}$. Then the verifier checks that

$$\prod_{q=1}^{Q} E(\hat{C}_q, \hat{D}_q) = \mu_T(t) \cdot \prod_{i=1}^{I} E(u_i, \pi_i) \cdot \prod_{j=1}^{J} E(\psi_j, v_j) \; .$$

The soundness of the proof system follows from the fact that under the perfectly binding parameters, the vectors u_i and v_j can be seen as commitments to 1. Then, by the properties of the bilinear map E, $\prod_{i=1}^{I} E(u_i, \pi_i) \cdot \prod_{j=1}^{J} E(\psi_j, v_j)$ is a commitment to the 1 element of $\mathbb{G}_T$, and based on the homomorphic property of the commitment schemes, $\prod_{q=1}^{Q} E(\hat{C}_q, \hat{D}_q)$ necessarily is a commitment to $t \in \mathbb{G}_T$.

Witness indistinguishability is more difficult to argue, but follows from the fact that under the perfectly hiding parameters, the proofs $\{\pi_i\}_{i=1}^{I}$, $\{\psi_j\}_{j=1}^{J}$ are random vectors of $\mathbb{V}_1$ and $\mathbb{V}_2$, that are only restricted by the constraint that $\prod_{q=1}^{Q} E(\hat{C}_q, \hat{D}_q) = \mu_T(t) \cdot \prod_{i=1}^{I} E(u_i, \pi_i) \cdot \prod_{j=1}^{J} E(\psi_j, v_j)$. As commitments are perfectly hiding, all proofs are drawn acording to the same distribution, no matter which witness was used by the prover. For further details we refer to [GS08].

F.3 Randomizable Non-Interactive Proofs

We consider a proof system with an additional algorithm RandProof. The basic idea is that RandProof takes a proof π for instance y in relation R, and produces a randomized proof of the same statement. The resulting proof must be indistinguishable from a new proof of the same statement. We allow the adversary to choose the instance y, the proof π that is used as input for RandProof, and the witness w that is used to form a new proof of the same statement. More formally:

Definition 5. *We say that* Setup, Prove, VerifyProof, RandProof *constitute a randomizable proof system if the following property holds. For all p.p.t. $\mathcal{A}_1, \mathcal{A}_2$ there exists a negligible function ν*

such that:

$$Pr[params \leftarrow \mathsf{Setup}(1^k); (y, w, \pi, state) \leftarrow \mathcal{A}_1(params); \\ \pi_0 \leftarrow \mathsf{Prove}(params, y, w); \pi_1 \leftarrow \mathsf{RandProof}(params, y, \pi); \\ b \leftarrow \{0,1\}; b' \leftarrow \mathcal{A}_2(state, \pi_b) : \\ R_L(y,w) \wedge \mathsf{VerifyProof}(params, y, \pi) = 1 \wedge b = b'] - \frac{1}{2} = \nu(k) \ .$$

Randomization is perfect if $\nu(k) = 0$.

Remark 2. Note that the existence of RandProof implies witness indistinguishability. However, randomization is a much stronger property. We will create an algorithm RandProof for the Groth-Sahai proof system; this is the only proof system that we know is randomizable.

Instantiating a randomizable proof system.

Lemma 6. *Groth-Sahai proofs are randomizable.*

Proof. RandProof gets as input an instance with the $a_q, b_q, t, \alpha_{q,m}, \beta_{q,n}$ values as well as the proof

$$[(\pi_1, \ldots, \pi_I, \psi_1, \ldots, \psi_J), \mathit{\Pi}] \ .$$

$\mathit{\Pi}$ contains the internal commitments $C_1, \ldots, C_M$ and $D_1, \ldots, D_N$.

The algorithm first chooses randomization exponents $(r_{1,1}, \ldots, r_{M,I})$ and $(s_{1,1}, \ldots, s_{N,J})$ at random from $\mathbb{Z}_p$. It then rerandomizes the commitments C_m and D_n to $C'_m = C_m \cdot \prod_{i=1}^{I} u_i^{r_{m,i}}$ and $D'_n = D_n \cdot \prod_{j=1}^{J} v_j^{s_{n,j}}$. Then it computes $\hat{s}_{q,i} = \sum_{m=1}^{M} r_{m,i} \cdot \alpha_{q,m}$, $\hat{z}_{q,j} = \sum_{n=1}^{N} s_{m,j} \cdot \beta_{q,m}$, $\hat{C}_q \leftarrow \mu_1(a_q) \cdot \prod_{m=1}^{M} C_m^{\alpha_{q,m}}$, and $\hat{D}'_q \leftarrow \mu_2(b_q) \cdot \prod_{n=1}^{N} D_n'^{\beta_{q,n}}$. Next, the prover sets

$$\pi'_i \leftarrow \pi_i \cdot \prod_{q=1}^{Q} (\hat{D}'_q)^{\hat{s}_{q,i}} \quad \text{and} \quad \psi'_j \leftarrow \psi_j \cdot \prod_{q=1}^{Q} (\hat{C}_q)^{\hat{z}_{q,j}} \ .$$

These π'_i and ψ'_j will satisfy the verification equation for the new commitments.

Now the prover must make a certain technical step to fully randomize the proof. Intuitively, for every set of commitments, there are many proofs $(\pi_1, \ldots, \pi_I, \psi_1, \ldots, \psi_J)$ that can satisfy the verification equation. Given one such proof, we can randomly choose another: the prover chooses $t_{i,j}, t_h \leftarrow R$, and multiplies each

$$\pi_i := \pi_i \cdot \prod_{j=1}^{J} v_j^{t_{i,j}} \quad \text{and each} \quad \psi_j := \psi_j \cdot \prod_{i=1}^{I} u_i^{\sum_{h=1}^{H} t_h \eta_{h,i,j}} \prod_{i=1}^{I} u_i^{t_{i,j}} \ .$$

See [GS08] for a detailed explanation of this operation.

The algorithm outputs the new proof $[(\pi'_1, \ldots, \pi'_I, \psi'_1, \ldots, \psi'_J), \mathit{\Pi}']$ where $\mathit{\Pi}'$ contains the internal commitments $C'_1, \ldots, C'_M$ and $D'_1, \ldots, D'_N$.