A New Hash-and-Sign Approach and Structure-Preserving Signatures from DLIN

Melissa Chase and Markulf Kohlweiss

Microsoft Research
{melissac,markulf}@microsoft.com

Abstract. Suppose we have a signature scheme for signing elements of message space $\mathcal{M}_1$, but we need to sign messages from $\mathcal{M}_2$. The traditional approach of applying a collision resistant hash function from $\mathcal{M}_1$ to $\mathcal{M}_2$ can be inconvenient when the signature scheme is used within more complex protocols, for example if we want to prove knowledge of a signature. Here, we present an alternative approach in which we can combine a signature for $\mathcal{M}_1$, a pairwise independent hash function with key space $\mathcal{M}_1$ and message space $\mathcal{M}_2$, and a non-interactive zero knowledge proof system to obtain a signature scheme for message space $\mathcal{M}_2$. This transform also removes any dependence on state in the signature for $\mathcal{M}_1$.

As a result of our transformation we obtain a new signature scheme for signing a vector of group elements that is based only on the 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, so the result is a structure-preserving signature. In combination with the Groth-Sahai proof system, such a signature scheme is an ideal building block for many privacy-enhancing protocols.

1 Introduction

The hash and sign approach. 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. However, in some applications there is also a disadvantage to this approach. In particular, it seems to be much more di cult to build e cient 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).

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]. One of the key elements in all of these protocols is the ability to prove that certain hidden values have been

signed without revealing the signature nor all of the certified values. Similarly, one might want to jointly compute a signature without revealing the key or all the certified 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, two party computation, 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.

On the other hand, pairwise-independent hash functions often have very simple, algebraic constructions that make them much better suited for proofs and multi-party computation. (For example, for a group G of prime order p, the simple function $H_{a,b}(x) = g^a x^b$ for key $(a, b) \in \mathbb{Z}_p^2$ can be shown to be a family of pairwise independent functions from G to G.) Thus, we consider an alternative approach, in which we can use pairwise-independent hash functions (together with NIZK proofs) to change the message spaces allowed by a given signature scheme.

Structure preserving signatures. 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 one can directly construct efficient proof systems or multi-party protocols. However, these schemes do have significant limitations. First, the resulting proof systems must be either interactive or in the random oracle model, which 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.

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 consists of the elements of the bilinear group. Furthermore, if the signatures are made up of group elements and the signature veri cation is done using the bilinear pairing, then the proof system of Groth and Sahai [GS08] allows for simple, e cient proofs. Abe et al. [AHO10] formalized these requirements (that messages, signatures, and public keys be group elements and veri cation proceed via a product of pairings) as *structure-preserving signatures* (SPS).[1]

Even before the term was coined, several early protocols made use of ad-hoc structure-preserving signature schemes but relied on very strong assumptions [AWSM07,ASM08,GH08]. Recently there have been a series of constructions for structure-prserving signature schemes [CLY09,AHO10,AGHO11]. However, all known e cient schemes are based on so-called \q-type" or interactive assumptions that are primarily justi ed based on the Generic Group model.[2] 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 satis es 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 e cient, a signature in Groth's scheme requires as con rmed 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 reasonably *efficient* constructions based on the DLIN assumption. Protocols based on our primitives are within an order of magnitude or two of the e ciency of the e cient protocols mentioned above.

Our results. First, we give a general approach for constructing a signature scheme for a message space $\mathcal{M}_1$ from a signature scheme for message space $\mathcal{M}_2$, a NIZK proof system, and a pairwise independent hash function with message space $\mathcal{M}_2$, key space $\mathcal{M}_1$, and any exponential sized range.

Then, as an application, we construct the rst practical structure preserving signature scheme secure under the DLIN assumption. To do this, we use the above transformation to transform a signature for signing elements of $\mathbb{Z}_p$ (with certain additional properties) into a structure preserving signature scheme.

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 hardness assumptions [BCKL08,BCKL09,Fuc09]. Thus, this already generates a range of structure preserving signatures schemes. However, all of these possible underly-

[1] For details on applications of SPS, we refer to [AFG+10] and to the full version [CK11].

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

ing 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.[3]

2 Preliminaries

In this section, we first describe the building blocks that we will use in our generic construction, and then summarize the assumptions that we will need for our application to structure-preserving signatures.

2.1 Weak F-unforgeable signature schemes

Our construction will require a signatures scheme unforgeable under a *weak chosen message attack* (Weak CMA) for signing elements of some messages space $\mathcal{K}$. In a weak chosen message attack, the adversary is required to make all of his signature queries at once, before seeing the public key or any signatures. In fact, we will see later that this signature scheme will only be used to sign random messages, thus security under weak chosen message attacks will suffice. In our SPS application, we will also require that the signature scheme be F-unforgeable for an appropriate bijection F. 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. In our SPS application this is important because when the message space is $\mathbb{Z}_p$, known pairing based proof systems only allow one to efficiently prove knowledge of some function of the message (e.g. g^m). We now formally define these notions:

Definition 1 (Unforgeability under Weak Chosen Message Attacks). *A weak chosen message attack (Weak CMA) [BB04,HW09b] requires that the adversary submits 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*

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

[3] 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.

For a bijection F, the Weak CMA F-unforgeability game is the same with the exception that instead of $\tilde{m}$, $\mathcal{A}_1$ only has to output $\tilde{f}$, such that $F^{-1}(\tilde{f}) \notin \{m_1, \ldots, m_Q\} \wedge \mathsf{SigVerify}(pk, F^{-1}(\tilde{f}), \tilde{\sigma}) = \mathsf{accept}$.

2.2 Pairwise independent hash functions.

The second ingredient will be a family of pairwise independent hash functions. This will be a family of functions parameterized by a "key" $k \in \mathcal{K}$. 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 2. *A family of hash-functions $\{H_k\}_{k\in\mathcal{K}}$, where $H_k : \mathcal{M} \to \mathcal{R}$ is called pairwise independent if $\forall x \neq y \in \mathcal{M}$ and $\forall a, b \in \mathcal{R}$, the probability*

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

2.3 Non-interactive zero-knowledge proofs

The final tool we need is a non-interactive zero-knowledge (NIZK) proof of knowledge system. A NIZK proof system consists of three algorithms PKSetup, PKProve, and PKVerify. $\mathsf{PKSetup}(1^k)$ is run by a trusted party and generates parameters crs (sometimes refered to as a common reference string) which are given to both the prover and the verifier. The prover runs $\mathsf{PKProve}(crs, x, w)$ to prove statement x with witness w which generates a proof π. The verifier runs $\mathsf{PKVerify}(crs, x, \pi)$ to verify the proof. Informally, zero knowledge means that there should exist a simulator (PKSimSetup, PKSimProve) that generates simulated parameters and simulated proofs that are indistinguishable from those produced by the prover (PKSetup, PKProve); a proof system is a proof of knowledge if there exists an extractor algorithm PKExtract that can extract a valid witness from any adversarially generated proof that is accepted by PKVerify.

We use the notation $\pi \leftarrow \mathsf{NIZKPK}\{(f(w)) : R_L(x, w)\}$ to indicate that π is a proof for statement x with witness w satisfying relation R_L and that from π we can extract $f(w)$.

2.4 Assumptions

Our concrete constructions will use bilinear groups groups $\mathbb{G}, \mathbb{G}_T$ of prime order p with a map e such that for any $g \in \mathbb{G}$, and any $a, b \in Z_p$, it must hold that $e(g^a, g^b) = e(g, g)^{ab}$, and if g is a generator for $\mathbb{G}$, then $e(g, g)$ must be a generator for $\mathbb{G}_T$. We rely on the following assumptions:

Definition 3 (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 4 (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.*

We also introduce a new assumption which we show is implied by DLIN:

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:

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

Theorem 1. *In groups with a symmetric bilinear pairing RCDH is implied by DLIN.* The proof can be found in the full version [CK11].

3 A New Hash-and-Sign Approach

Our main result is to show how to construct a signature scheme for signing elements of a message space $\mathcal{M}$ based on an efficient NIZK proof of knowledge system, a signature scheme for signing message space $\mathcal{K}$ and a family of pairwise independent hash functions $\{H_k\} : \mathcal{M} \to \mathcal{R}$ with key space $\mathcal{K}$ and exponential sized range.

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

3.1 A stateless signature scheme for message space $\mathcal{M}$

Let $\mathsf{Sig}_{\mathcal{K}} = (\mathsf{Kg}_{\mathcal{K}}, \mathsf{Sign}_{\mathcal{K}}, \mathsf{Verify}_{\mathcal{K}})$ be a (potentially stateful) Weak CMA F-unforgeable signature scheme on message space $\mathcal{K}$ for some bijection F. (Note that a stateless signature scheme would suffice - the construction would then simply not use the state s.) Let $\{h_k\}_{\mathcal{K}(\lambda)} : \mathcal{M} \to \mathcal{R}$ be a pairwise independent hash function.[4] Let $\mathsf{Setup}, \mathsf{Prove}, \mathsf{VerifyProof}$ be a non-interactive zero knowledge

[4] We will omit the security parameter λ and simply write $\mathcal{K}$ when it is clear from context.

proof of knowledge system. We construct a signature scheme with message space $\mathcal{M}$ as follows:

SigKg(1^λ)**:** Run $\mathsf{Kg}_\mathcal{K}(1^\lambda)$ to generate a key pair $(pk_\mathcal{K}, sk_\mathcal{K})$. Generate the common reference string crs for a NIZKPK proof system. Output $pk = (pk_\mathcal{K}, crs)$ and $sk = (sk_\mathcal{K}, crs)$.

Sign(sk, M)**:** Parse $sk = (sk_\mathcal{K}, crs)$. Choose random key $k \leftarrow \mathcal{K}$. Compute the signature $\sigma_\mathcal{K} \leftarrow \mathsf{Sign}_\mathcal{K}{}^{s=0}(sk_\mathcal{K}, k)$ and the hash value $h = H_k(M)$. Finally, construct a proof of knowledge of $F(k)$ and the corresponding signature, i.e.:

$$\pi \in \mathsf{NIZKPK}\{(f, \sigma_\mathcal{K}) : \{\exists k \in \mathcal{K} \text{ s.t. } f = F(k) \wedge \mathsf{Verify}_\mathcal{K}(pk_\mathcal{K}, k, \sigma_\mathcal{K}) = 1 \wedge h = H_k(M)\}\}$$

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

Note that we write $\mathsf{Sign}_\mathcal{K}{}^{s=0}$ 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.

SigVerify(pk, M, σ)**:** Parse $pk = (pk_\mathcal{K}, crs)$ and $\sigma = (h, \pi)$. Verify the proof π w.r.t. crs and $pk_\mathcal{K}, h, M$.

3.2 Unforgeability of the signature scheme

We now prove our main result:

Theorem 2. *Given a (potentially stateful) Weak CMA F-unforgeable signature scheme* $(\mathsf{Kg}_\mathcal{K}, \mathsf{Sign}_\mathcal{K}{}^s, \mathsf{Verify}_\mathcal{K})$*, a secure NIZKPK proof system* (Setup, Prove, VerifyProof)*, and a pairwise independent hash function family $\{H_k\}_{k \in \mathcal{K}(\lambda)}$ whose range is exponential in λ, the resulting construction* (SigKg, Sign, SigVerify) *is a stateless CMA unforgeable signature scheme.*

Proof. We formally prove the security of the transformation using a sequence of games. For simplicity, we will assume that the proof system has perfect soundness and perfect extraction, but this can be relaxed to allow for a negligible error. Let $p_i(\lambda)$ 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 the success probability in this game 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 SigVerify accept for a message vector that was never signed before.

The adversary succeeds with probability $p_1(\lambda)$.

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

Lemma 1. $\Delta_1(\lambda) = |p_2(\lambda) - p_1(\lambda)|$ *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 $\mathsf{Sign}_{\mathcal{K}}{}^{s=0}$ and by $\mathsf{Sign}_{\mathcal{K}}{}^{s}$ 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 $\mathsf{Sign}_{\mathcal{K}}{}^{s=0}$ is replaced by $\mathsf{Sign}_{\mathcal{K}}{}^{s}$. 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 $\mathsf{Sign}_{\mathcal{K}}{}^{s=0}$ and $\mathsf{Sign}_{\mathcal{K}}{}^{s}$ 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 since $\mathcal{A}$ can make at most a polynomial number of queries, $\Delta_1(\lambda)$ is negligible □

Game 3: reusing k. This game will proceed just as **Game 2** except that once the adversary outputs his forgery, $M, \sigma = (h, \pi)$, we will extracts f' from π, and compare it against the values used to answer the adversary's queries. The adversary succeeds in this game if and only if the signature verifies, the message is new, and the value f' corresponds to $F(k)$ for some k used to answer a previous query. Let $p_3(\lambda)$ be the probability that the adversary succeeds in this game.

Lemma 2. $\Delta_2(\lambda) = |p_3(\lambda) - p_2(\lambda)|$ *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 a value f' can be extracted that does 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 values $k_1 \ldots k_Q \in \mathcal{K}$ 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 $\mathcal{A}$. $\mathcal{B}$ extracts $\sigma_{\mathcal{K}}$ and $f' \notin \{F(k_1), \ldots, F(k_Q)\}$ from π and outputs it as a forgery. By perfect extraction, we are guaranteed that $\sigma_{\mathcal{K}}$ is a valid signature on $F^{-1}(f)$, so if $\mathcal{A}$ is successful in producing event **Bad**, then $f', \sigma_{\mathcal{K}}$ exactly matches the definition of a valid Weak CMA F-forgery. Consequently we conclude that $\Delta_2(\lambda) \leq Pr[\mathbf{Bad}]$. □

Game 4: check h. This game will proceed as in **Game 2** except that once the adversary outputs his forgery, $M, \sigma = (h, \pi)$, we let $K = (k_1, \ldots, k_Q)$ be

the set of hash keys used to answer the adversary's queries. Then we verify whether $h = H_{k_i}(M)$ for any $i \in 1 \ldots Q$. The adversary succeeds if and only if the signature verifies, the message is new, and this check succeeds (i.e. there is such a value). Let $p_4(\lambda)$ be the probability that the adversary succeeds in this game.

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

Proof. If h is computed correctly with the hash key k corresponding to the value $f = F(k)$ extracted from the proof, **Game 4** will be successful in all cases in which **Game 3** is successful. Thus, this follows directly from the perfect extraction of the proof system. □

Game 5: simulate proofs. In this game, when the public parameters are generated, the challenger will run SimSetup to generate parameters crs, and trapdoor sim. When responding to signature queries, the challenger chooses random $k \leftarrow \mathcal{K}$ and forms h as in the real signing protocol, but generates the proof using SimProve. As above, we judge the adversary's success by verifying the proof and checking the h component of the signature against the set of hash keys $\{k_1, \ldots, k_Q\}$ used in previous queries. Let $p_5(\lambda)$ be the probability that the adversary succeeds in this game.

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

Proof. An attacker with non-negligible $\Delta_4(\lambda)$ 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 $\Delta_4(\lambda)$, 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 public key using the parameters received from the challenge game; to generate each signature, it chooses random $k \leftarrow \mathcal{K}$, generates $h, \sigma_{\mathcal{K}}$ as in the signing algorithm, and generate the zero-knowledge proof using an oracle query. If $\mathcal{A}$ succeeds in producing h which does not correspond to any of the hash keys $k_1, \ldots, k_Q$ together with a proof π that verifies, then $\mathcal{B}$ outputs 1. If $|p_5(\lambda) - p_4(\lambda)|$ is non-negligible, then $\mathcal{B}$ will succeed in the zero knowledge game with non-negligible advantage.

□

Lemma 5. *$p_5(\lambda)$ is negligible when h is computed by a pairwise-independent hash function whose range $\mathcal{R}$ is exponential in λ.*

Proof. Suppose we know h and M for some unknown hash key k. Then for any other $h' \in \mathcal{R}$, $M' \in \mathcal{M}$, the probability (taken over possible values of $k \in \mathcal{K}$) that $h' = H_k(M')$ is $1/|\mathcal{R}|$ by pairwise independence. Thus, for any key k used by the signer, the probability of $\mathcal{A}$ producing a correct pair h', M' for that tuple is at most $1/|\mathcal{R}|$. Taking a union bound over all tuples used gives $q/|\mathcal{R}|$ where q is the total number of queries made by $\mathcal{A}$. This will be negligible since q is polynomial and $|\mathcal{R}|$ is exponential in λ.

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

4 Structure-Preserving Signatures from DLIN

Here we show that we can instantiate the building blocks described in the previous section based on DLIN, to construct a structure-preserving signature scheme. (In fact, we will describe a structure preserving scheme which allows us to sign vectors of ℓ group elements at once.)

First, we will review the Groth-Sahai NIZK proof system [GS08], which gives efficient proofs that are compatible with many pairing based schemes. Then we briefly present the pairwise-independent hash function we use, and how it can be used with Groth-Sahai. Finally, we will construct a new signature scheme for elements in $\mathbb{Z}_p^{\ell+1}$ which is both secure under DLIN and compatible with the Groth-Sahai proof system. Putting all of these together using the generic construction in Section 3 gives a secure signature scheme. Finally, since the hash function produces elements in the bilinear group $\mathbb{G}$, and Groth-Sahai proofs are composed of elements in $\mathbb{G}$ and can be verified with pairing product equations, the result is a structure preserving signature scheme.

4.1 NIZK proofs based on DLIN: the Groth-Sahai proof system.

Groth and Sahai [GS08] (in an extension of the results of [GOS06b] and [GOS06a]) showed how to construct non-interactive 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.

Groth-Sahai proofs. The Groth-Sahai proof system allows to 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 $open_i$. This

means that given a Groth-Sahai proof for a pairing product equation we can extract all the elements of $\mathbb{G}$ that make up the witness.

4.2 Pairwise independent hash functions.

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

$$H_k(M_1, \ldots, M_\ell) = g^{k_0} \prod_{i=1..\ell} M_i^{k_i},$$

where $k = (k_0, \ldots, k_\ell)$. We show that this function family is indeed pairwise independent:

Theorem 3. *The above function family is pairwise independent.*

Proof. Let us express the probability

$$Pr[k \leftarrow \mathcal{K} : H_k(x) = a \wedge H_k(y) = b] = \frac{|\{k_0, \ldots, k_\ell \mid g^{k_0} \prod_{i=1..\ell} x_i^{k_i} = a \wedge g^{k_0} \prod_{i=1..\ell} y_i^{k_i} = b\}}{|\mathbb{Z}_p|^{\ell+1}}.$$

We have to show that the numerator equals $|\mathbb{Z}_p|^{\ell-1}$. This can be seen by looking at $g^{k_0} \prod_{i=1..\ell} x_i^{k_i} = a$ and $g^{k_0} \prod_{i=1..\ell} y_i^{k_i} = b$ as independent linear equations over the variables $k_0, \ldots, k_\ell$ (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 $|\mathbb{Z}_p|^{\ell-1}$.

Finally, we observe that, given $g^{k_0}, \ldots, g^{k_\ell}$, we can easily use a pairing product equation to verify that h is correctly computed: for key $k = k_0, \ldots, k_\ell$ and message $M = M_1, \ldots, M_\ell$, it will be the case that $h = H_k(M)$ i $e(h, g) = e(g, g^{k_0}) \prod_{i=1}^{\ell} e(M_i, g^{k_i})$. Thus, we can use the Groth-Sahai proof system to prove knowledge of $g^{k_0}, \ldots, g^{k_1}$ and $M_1, \ldots, M_k$ such that h is correct.

4.3 A signature scheme for elements of Z_p

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 Di e-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, the result will be a fully secure, stateless signature scheme for signing group elements.

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 e ciently 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]. 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 $\mathbb{Z}_p$.[5] 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 note 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.[6]

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:

$\mathsf{SigKg}_{exp}(1^\lambda)$ runs the Waters key generation to generate (pk_w, sk_w), chooses random $a \leftarrow \mathbb{Z}_p$ and $u, d, z, h \leftarrow \mathbb{G}$, and 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.)

$\mathsf{Sign}^s_{exp}(sk, m)$ is a stateful signature algorithm which first 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 $\sigma = (\sigma_1, \sigma_2, \sigma_3, s)$.

[5] 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.

[6] 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. Weak CMA F-unforgeable signatures are, however, sufficient to obtain a CMA secure signature scheme for signing group elements via our transform.

$\mathsf{SigVerify}_{exp}(pk, m, \sigma)$ 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 $e(\sigma_1, g) = e(u^m d, g^a)e(\sigma_2, z^i h)$.

Note: We write $\mathsf{Sign}^s_{exp}(sk, m)$ to indicate that we run the signing algorithm on state s.

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 to 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 4. *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.

Theorem 5. *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. Since RCDH is implied by DLIN, this means the signature is secure under DLIN.*

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 $m_1, \ldots, m_Q$, $Q \leq poly(\lambda)$. 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 $m_1, \ldots, m_Q$, $Q \leq poly(\lambda)$, 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 random t' and implicitly let $t = t' - a(m_s - m_{i^*})/(s - i^*)$. Form $\sigma_1 = (g^a)^{y_d} T^{x_z s + x_h} (g^b)^{t'(s-i^*)}$ and $\sigma_2 = T$ for $T = g^{t'}/(g^a)^{(m_s - m_{i^*})/(s-i^*)}$.

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 $\tilde{\sigma} = (\tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{\sigma}_3, \tilde{i})$, $g^{\tilde{m}}$, $\mathcal{G}^{\tilde{m}}$, such that $\tilde{m} \notin \{m_1, \ldots, m_Q\}$. If $\tilde{i} \neq i^*$ the reduction aborts. Otherwise it outputs $g^{\tilde{m}}/g^{m_{i^*}}$, $\mathcal{G}^{\tilde{m}}/\mathcal{G}^{m_{i^*}}$ and $\tilde{\sigma}_1 / g^{a y_d} \tilde{\sigma}_2^{(x_z \tilde{i} - x_h)}$ 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 $n > 1$. 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:

$\mathsf{SigKg}_{exp}(1^\lambda)$ 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. If we combine this with the pairing-product equations described in Section 4.2, we can generate an efficient GS proof for the relation needed for our generic construction.

Instantiation	stateless signature
DLIN	$100 + 24\ell + 9x$
q-BB-HSDH + q-TDH + DLIN	$79 + 7\ell$
RCDH + SXDH	$77 + 18\ell + 6x$
q-BB-HSDH + q-TDH + SXDH	$61 + 6\ell$

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.

4.4 Performance analysis

For the performance analysis we instantiate our signatures and proofs with two signature schemes { the scheme based on RCDH described in Section 4.3 and one based on q-BB-HSDH and q-TDH described in [BCKL09].[7] 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 the full version [CK11]. 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).

5 Conclusion and Open Problems

We construct a reasonably e cient 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 tradeo between e ciency and security, and we argue that in many cases sacri cing an order of magnitude in e ciency for a signi cantly 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 satis ed with schemes based on the generic group model, but rather that we should continue looking for schemes which are *both* e cient *and* based on weak assumptions.

[7] For a discussion of other possible instantiations for the exponent signature scheme, see the full version [CK11].

References

[ACJT00] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik, *A practical and provably secure coalition-resistant group signature scheme*, CRYPTO(Mihir Bellare, ed.), LNCS, vol. 1880, Springer-Verlag, 2000, pp. 255–270.

[AFG+10] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo, *Structure-preserving signatures and commitments to group elements*, CRYPTO (Tal Rabin, ed.), LNCS, vol. 6223, Springer-Verlag, 2010, pp. 209–236.

[AGHO11] Masayuki Abe, Jens Gorth, Kristiyan Haralambiev, and Miyako Ohkubo, *Optimal structure-preserving signatures in asymmetric bilinear groups*, CRYPTO (Phillip Rogaway, ed.), LNCS, vol. 6841, Springer-Verlag, 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*, Financial Cryptography (Gene Tsudik, ed.), LNCS, vol. 5143, Springer-Verlag, 2008, pp. 287–301.

[AWSM07] Man Ho Au, Qianhong Wu, Willy Susilo, and Yi Mu, *Compact e-cash from bounded accumulator*, CT-RSA (Masayuki Abe, ed.), LNCS, vol. 4377, Springer-Verlag, 2007, pp. 178–195.

[BB04] Dan Boneh and Xavier Boyen, *Short signatures without random oracles*, EUROCRYPT(Christian Cachin and Jan Camenisch, eds.), LNCS, vol. 3027, Springer-Verlag, 2004, pp. 54–73.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham, *Short group signatures using strong Diffie-Hellman*, CRYPTO(Matthew K. Franklin, ed.), LNCS, vol. 3152, Springer-Verlag, 2004, pp. 41–55.

[BCDvdG87] Ernest F. Brickell, David Chaum, Ivan Damgård, and Jeroen van de Graaf, *Gradual and verifiable release of a secret*, CRYPTO (Carl Pomerance, ed.), LNCS, vol. 293, Springer-Verlag, 1987, pp. 156–166.

[BCKL08] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya, *P-signatures and noninteractive anonymous credentials*, TCC (Ran Canetti, ed.), LNCS, vol. 4948, Springer-Verlag, 2008, pp. 356–374.

[BCKL09] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya, *Compact e-cash and simulatable VRFs revisited*, Pairing (Hovav Shacham and Brent Waters, eds.), LNCS, vol. 5671, Springer-Verlag, 2009, pp. 114–131.

[BCL04] Endre Bangerter, Jan Camenisch, and Anna Lysyanskaya, *A cryptographic framework for the controlled release of certified data*, Workshop on Security Protocols, LNCS, Springer-Verlag, 2004.

[Bou00] Fabrice Boudot, *Efficient proofs that a committed number lies in an interval*, EUROCRYPT(Bart Preneel, ed.), LNCS, vol. 1807, Springer-Verlag, 2000, pp. 431–444.

[CCS08] Jan Camenisch, Rafik Chaabouni, and Abhi Shelat, *Efficient protocols for set membership and range proofs*, ASIACRYPT (Josef Pieprzyk, ed.), LNCS, vol. 5350, Springer-Verlag, 2008, pp. 234–252.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers, *Proofs of partial knowledge and simplified design of witness hiding protocols*, CRYPTO (Yvo Desmedt, ed.), LNCS, vol. 839, 1994, pp. 174–187.

[CG08] Sébastien Canard and Aline Gouget, *Anonymity in transferable e-cash*, ACNS (Steven M. Bellovin, Rosario Gennaro, Angelos D. Keromytis, and Moti Yung, eds.), LNCS, vol. 5037, Springer-Verlag, 2008, pp. 207–223.

[CGH06] Sébastien Canard, Aline Gouget, and Emeline Hufschmitt, *A handy multi-coupon system.*, ACNS (Jianying Zhou, Moti Yung, and Feng Bao, eds.), LNCS, vol. 3989, Springer-Verlag, 2006, pp. 66–81.

[CGH09] Scott E. Coull, Matthew Green, and Susan Hohenberger, *Controlling access to an oblivious database using stateful anonymous credentials*, PKC (Stanislaw Jarecki and Gene Tsudik, eds.), LNCS, vol. 5443, Springer-Verlag, 2009, pp. 501–520.

[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*, ACM CCS (Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, eds.), ACM, 2006, pp. 201–210.

[CHL05] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya, *Compact E-Cash*, EUROCRYPT (Ronald Cramer, ed.), LNCS, vol. 3494, Springer-Verlag, 2005, pp. 302–321.

[CHL06] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya, *Balancing accountability and privacy using e-cash*, SCN (Roberto De Prisco and Moti Yung, eds.), LNCS, vol. 4116, Springer-Verlag, 2006, pp. 141–155.

[CK11] Melissa Chase and Markulf Kohlweiss, *A domain transformation for structure-preserving signatures on group elements*, IACR Cryptology ePrint Archive (2011), 342.

[CL01] Jan Camenisch and Anna Lysyanskaya, *Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation*, EUROCRYPT(Birgit Pfitzmann, ed.), LNCS, vol. 2045, Springer-Verlag, 2001, pp. 93–118.

[CL02] Jan Camenisch and Anna Lysyanskaya, *A signature scheme with efficient protocols*, SCN (Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, eds.), LNCS, vol. 2576, Springer-Verlag, 2002, pp. 268–289.

[CL06] Melissa Chase and Anna Lysyanskaya, *On signatures of knowledge*, CRYPTO (Cynthia Dwork, ed.), LNCS, vol. 4117, 2006, pp. 78–96.

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

[CLY09] Julien Cathalo, Benoît Libert, and Moti Yung, *Group encryption: Non-interactive realization in the standard model*, ASIACRYPT (Mitsuru Matsui, ed.), LNCS, vol. 5912, Springer-Verlag, 2009, pp. 179–196.

[Cra97] Ronald Cramer, *Modular design of secure yet practical cryptographic protocols*, Ph.D. thesis, University of Amsterdam, Amsterdam, 1997.

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

[EG85] Taher El Gamal, *A public key cryptosystem and a signature scheme based on discrete logarithms*, CRYPTO (G. R. Blakley and David Chaum, eds.), LNCS, vol. 196, Springer-Verlag, 1985, pp. 10–18.

[FP08] Georg Fuchsbauer and David Pointcheval, *Anonymous proxy signatures*, SCN (Rafail Ostrovsky, Roberto De Prisco, and Ivan Visconti, eds.), LNCS, vol. 5229, Springer-Verlag, 2008, pp. 201–217.

[FP09] Georg Fuchsbauer and David Pointcheval, *Proofs on encrypted values in bilinear groups and an application to anonymity of signatures*, Pairing (Hovav Shacham and Brent Waters, eds.), LNCS, vol. 5671, Springer-Verlag, 2009, pp. 132–149.

[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*, ASIACRYPT, LNCS, vol. 5350, Springer-Verlag, 2008, pp. 179–197.

[GOS06a] Jens Groth, Rafail Ostrovsky, and Amit Sahai, *Non-interactive Zaps and new techniques for NIZK*, CRYPTO (Cynthia Dwork, ed.), LNCS, vol. 4117, Springer-Verlag, 2006, pp. 97–111.

[GOS06b] Jens Groth, Rafail Ostrovsky, and Amit Sahai, *Perfect non-interactive zero knowledge for NP*, EUROCRYPT (Serge Vaudenay, ed.), LNCS, vol. 4004, Springer-Verlag, 2006, pp. 339–358.

[Gro06] Jens Groth, *Simulation-sound nizk proofs for a practical language and constant size group signatures*, ASIACRYPT (Xuejia Lai and Kefei Chen, eds.), LNCS, vol. 4284, Springer-Verlag, 2006, pp. 444–459.

[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*, EUROCRYPT (Nigel Smart, ed.), LNCS, vol. 4965, Springer-Verlag, 2008.

[HW09a] Susan Hohenberger and Brent Waters, *Realizing hash-and-sign signatures under standard assumptions*, EUROCRYPT (Antoine Joux, ed.), LNCS, vol. 5479, Springer-Verlag, 2009, pp. 333–350.

[HW09b] Susan Hohenberger and Brent Waters, *Short and stateless signatures from the rsa assumption*, CRYPTO (Shai Halevi, ed.), LNCS, vol. 5677, Springer-Verlag, 2009, pp. 654–670.

[TS06] Isamu Teranishi and Kazue Sako, *k-times anonymous authentication with a constant proving cost.*, PKC (Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, eds.), LNCS, Springer-Verlag, 2006, pp. 525–542.

[Wat05] B. Waters, *Efficient identity-based encryption without random oracles*, EUROCRYPT(Ronald Cramer, ed.), LNCS, vol. 3494, Springer-Verlag, 2005.