Shorter Non-Interactive Zero-Knowledge Arguments and ZAPs for Algebraic Languages

Geoffroy Couteau[1], Dominik Hartmann[2]

[1] CNRS, IRIF, Université de Paris, France
couteau@irif.fr
[2] Ruhr-University Bochum, Germany
Dominik.Hartmann@rub.de

Abstract. We put forth a new framework for building pairing-based non-interactive zero-knowledge (NIZK) arguments for a wide class of algebraic languages, which are an extension of linear languages, containing disjunctions of linear languages and more. Our approach differs from the Groth-Sahai methodology, in that we rely on pairings to compile a Σ-protocol into a NIZK. Our framework enjoys a number of interesting features:

- conceptual simplicity, parameters derive from the Σ-protocol;
- proofs as short as resulting from the Fiat-Shamir heuristic applied to the underlying Σ-protocol;
- fully adaptive soundness and perfect zero-knowledge in the common random string model with a single random group element as CRS;
- yields simple and efficient two-round, public coin, publicly-verifiable perfect witness-indistinguishable (WI) arguments(ZAPs) in the plain model. To our knowledge, this is the *first* construction of two-rounds *statistical* witness-indistinguishable arguments from pairing assumptions.

Our proof system relies on a new (static, falsifiable) assumption over pairing groups which generalizes the standard kernel Diffie-Hellman assumption in a natural way and holds in the generic group model (GGM) and in the algebraic group model (AGM).

Replacing Groth-Sahai NIZKs with our new proof system allows to improve several important cryptographic primitives. In particular, we obtain the shortest tightly-secure structure-preserving signature scheme (which are a core component in anonymous credentials), the shortest tightly-secure quasi-adaptive NIZK with unbounded simulation soundness (which in turns implies the shortest tightly-mCCA-secure cryptosystem), and shorter ring signatures.

Keywords: zero-knowledge arguments, non-interactive zero-knowledge arguments, satistical witness-indistinguishability, pairing-based cryptography, tight security, structure-preserving signatures.

1 Introduction

Zero-knowledge proof systems, introduced in the seminal paper of Goldwasser, Micali, and Rackoff [38], allow a prover to convince a verifier of the truth of a statement, without revealing anything beyond this. Zero-knowledge proofs are among the most fundamental cryptographic primitives, and enjoy a tremendous number of applications. A particularly useful kind of zero-knowledge proof systems are *non-interactive zero-knowledge proofs* (NIZKs) [13], which consist of a single flow from the prover to the verifier. NIZKs have found a wide variety of applications in cryptography, ranging from low-interactions secure computation protocols to the design of advanced cryptographic primitives and protocols such as verifiable encryption, group signatures, structure-preserving signatures, anonymous credentials, KDM-CCA2 and identity-based CCA2 encryption, among many others.

Early feasibility results for NIZKs were established in the 90's, under standard assumptions such as factorization, or the existence of (doubly-enhanced) trapdoor permutations [29]. While these results demonstrated the possibility of building NIZKs under standard assumption for all NP languages (in the common reference string model), they were typically built upon a reduction to an NP-complete language such as graph hamiltonicity, and were concretely inefficient.

The Fiat-Shamir (FS) transform [30], which relies on a hash function to compile an interactive ZK proof into a NIZK, provides a practical alternative to the above, leading to efficient NIZK arguments; however, it only offers heuristic security guarantees and any security proof for the FS transform must overcome several barriers [7, 37][1]. Hence, for two decades after their introduction, essentially two types of NIZKs coexisted: inefficient NIZKs provably secure in the standard (common reference string) model, and heuristically secure practical NIZKs.

1.1 Pairing-Based NIZKs

With the advent of pairing-based cryptography, this somewhat unsatisfying situation changed. Starting with the celebrated work of Groth and Sahai [44], a variety of pairing-based NIZK proof systems have been introduced. These proof systems have in common that they handle directly a large class of languages over abelian groups, avoiding the need for expensive reductions to NP-complete problems. Due to its practical significance, the Groth-Sahai proof system (and its follow-ups) initiated a wide variety of cryptographic applications. As of today, all known practically efficient (publicly verifiable) NIZKs in the standard model rely on pairing-based cryptography. Existing pairing-based NIZK proof systems can be divided in two categories:

NIZKs based on the Groth-Sahai (GS) methodology. These NIZKs directly rely on the techniques developed in [44], and enhance the seminal construction in various ways [12, 25, 27, 66]. Unfortunately, in spite of these optimizations, Groth-Sahai proofs remain often unsatisfyingly inefficient, and are in

[1] Alternatively, the Fiat-Shamir transform offers provable security in the random oracle model; we note that there have been recent developments regarding instantiating Fiat-Shamir in the standard model under strong assumptions [16, 55].

particular notably less efficient than (heuristic) NIZKs obtained with the Fiat-Shamir transform. Furthermore, the design and analysis of a suitable NIZK, taking into account all existing optimizations, is often a tedious and error-prone task.

Quasi-Adaptive NIZKs for Linear Languages. In light of the above, an alternative line of research, starting with the work of [51] and culminating with [57], has investigated a different strategy for building pairing-based NIZKs. Roughly, the approach relies on a hash proof system [22] (HPS) for the target language over some abelian group $\mathbb{G}_1$, which can be seen as a kind of designated-verifier NIZK proof, and makes it publicly verifiable by embedding the secret hashing key in the group $\mathbb{G}_2$. Verifying the proof is done with the help of a pairing operation between $\mathbb{G}_1$ and $\mathbb{G}_2$. The HPS-based approach leads to conceptually simple and very efficient proofs (e.g. a membership proof for the DDH language can be made as short as a single group element in [57]). However, this efficiency comes with strong limitations: this approach can only handle linear languages, and only provides a quasi-adaptive type of soundness, where the common reference string is allowed to depend on the language.

1.2 Our Contribution

In this work, we introduce a new approach for building efficient, pairing-based non-interactive zero-knowledge arguments for a large class of languages, where soundness relies on a new (but plausible, static, and falsifiable) assumption, which extends the kernel Diffie-Hellman assumption [63] in a natural way. Our approach is very simple and natural; yet it has to our knowledge never been investigated. It leads to proofs which are shorter and conceptually much simpler than proofs obtained with the GS methodology. At the same time and unlike the HPS-based methodology, our proof system is not limited to linear languages, but handles a more general class of *witness samplable* languages where, roughly, the language parameters can be sampled together with a trapdoor which can be used to decide membership in the language (in particular, this captures the important case of disjunctions of linear languages, from which one can build linear-size NIZKs for circuit satisfiability using the GOS methodology [42]) and achieves fully adaptive soundness with very short common random strings.

Statistical ZAPs and NIWIs. An additional benefit of our NIZK proof system is that it works in the common *random* string model, where the CRS is just a random bit string. Furthermore, we show that if we let the verifier pick the CRS himself, our proof system still satisfies *statistical witness-indistinguishability.* Therefore, we obtain the shortest two-round publicly-verifiable witness-indistinguishable argument system in the plain model (i.e., a ZAP [26]) for witness-samplable algebraic languages. Our ZAPs can be turned into fully non-interactive witness-indistinguishable arguments in the plain model, using the derandomization method of [8]. We emphasize that the ZAPs obtained with our method are *statistically* witness-indistinguishable; to our knowledge, our construction is the first pairing-based statistical ZAP (it is in addition publicly verifiable, and public coin). Existing constructions of statistical ZAPs rely on the quasipolynomial hardness of LWE [6, 49], or rely on subexponential variants of standard assumptions and are not public coin [54]. While our result comes at the cost of basing soundness on a new pairing-based assumption, we believe that it represents a significant contribution to the important and long standing open question of building statistical ZAPs.

High Level Overview. At a high level, our approach consists in compiling a three-move public coin zero-knowledge protocol (so called Σ-protocol) with linear answers over an abelian group $\mathbb{G}_1$ into a non-interactive zero-knowledge argument, by embedding the challenge e into a group $\mathbb{G}_2$ such that there is an asymmetric pairing between $\mathbb{G}_1 \times \mathbb{G}_2$ and a target group $\mathbb{G}_T$, and adding the embedded challenge to the common reference string. Intuitively, correctness is preserved because the pairing can be used to perform the verification procedure, zero-knowledge is perfect, and soundness follows from the fact that a cheating adversary must compute a value in $\mathbb{G}_1$ which has a non-trivial relation to e, which is conjectured to be intractable. An important part of our work is devoted to the analysis of the soundness property of our proof system, and the underlying assumption.

In addition to the efficiency improvements it provides, an important conceptual advantage of our approach over the Groth-Sahai methodology is that it gives a very simple and natural way to construct NIZKs. The construction of optimized Groth-Sahai proofs is generally cumbersome, and a significant amount of expertise is often required for the design of the best-possible GS proof in a given context. In contrast, Σ-protocols are typically straightforward to construct, and require considerably less expertise to optimize. Building a NIZK with our approach requires only to design an algebraic Σ-protocol for the target language distribution, and compiling it into a NIZK (which essentially amounts to adding a single group element to the CRS). Computation, communication and the underlying assumption can be obtained in a straightforward way from the parameters of the underlying Σ-protocol. We believe that this conceptual simplicity is an important feature toward making the use of pairing-based NIZKs accessible to a wider spectrum of researchers and industrials.

1.3 Technical Overview

The starting point of our approach is a (somewhat folklore) Σ-protocol for algebraic languages [10, 17]. A Σ-protocol is a three-move public-coin honest-verifier zero-knowledge proof system (i.e., the message of the verifier is a random string, and the zero-knowledge property holds against verifiers that do not deviate from the specifications of the protocol). In the following, we use the implicit notations introduced in [28]: given a group $\mathbb{G}$ in additive form, we fix a generator g and write $[x]$ for $x \cdot g$. Most, if not all, algebraic languages over abelian groups considered in the literature can be written as $\mathcal{L}_{\mathbf{M},\mathbf{\Theta}} := \{\mathbf{x} \in \mathbb{G}^l | \exists \mathbf{w} \in \mathbb{Z}_p^t : \mathbf{M}(\mathbf{x}) \cdot \mathbf{w} =$

$\mathbf{\Theta}(\mathbf{x})\}$, where $\mathbf{M} : \mathbb{G}^l \mapsto \mathbb{G}^{n \times t}$ and $\mathbf{\Theta} : \mathbb{G}^l \mapsto \mathbb{G}^n$ are linear maps sampled according to a distribution $\mathcal{D}_{par}$. This captures all algebraic languages defined by systems of polynomial equations between secret exponents. Most Σ-protocols for algebraic languages can then be seen as particular instantiations of the generic Σ-protocol represented on Figure 1.

To compile this Σ-protocol into a NIZK, we assume that all computations take place in a group $\mathbb{G}_1$, such that there exists another group $\mathbb{G}_2$ together with an asymmetric pairing $\bullet : \mathbb{G}_1 \times \mathbb{G}_2 \mapsto \mathbb{G}_T$. We use the standard brackets with subscripts $[\cdot]_1, [\cdot]_2, [\cdot]_T$ to extend the implicit notation to the three groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$. The setup algorithm of our proof system picks a random $e \in \mathbb{Z}_p$ and sets the common reference string to $[e]_2$. The prover computes $[\mathbf{a}]_1$ as in the Σ-protocol, and obtains the value $\mathbf{d}$ embedded in $\mathbb{G}_2$ by computing $[\mathbf{d}]_2 := \mathbf{w} \cdot [e]_2 + \mathbf{r} \cdot [1]_2$. Checking the verification equation can still be done, with the help of the pairing: the verifier checks that $[\mathbf{M}(\mathbf{x})]_1 \bullet [\mathbf{d}]_2 \stackrel{?}{=} [\mathbf{\Theta}(\mathbf{x})]_1 \bullet [e]_2 + [\mathbf{a}]_1 \bullet [1]_2$. While this construction is relatively simple, the bulk of our technical contribution is the detailed analysis of the security guarantees it provides.

$$(\mathbf{M}, \mathbf{\Theta}) \xleftarrow{\$} \mathcal{D}_{par}$$

$$\begin{array}{ccc}
\mathcal{P} & & \mathcal{V} \\
[\mathbf{x}], \mathbf{w} & & [\mathbf{x}] \\
\mathbf{r} \xleftarrow{\$} \mathbb{Z}_p^k & & \\
[\mathbf{a}] := [\mathbf{M}(\mathbf{x})]\mathbf{r} & \xrightarrow{[\mathbf{a}]} & \\
 & \xleftarrow{e} & e \xleftarrow{\$} \mathbb{Z}_p \\
\mathbf{d} := e\mathbf{w} + \mathbf{r} & \xrightarrow{\mathbf{d}} & \\
 & & \text{check} \\
 & & [\mathbf{M}(\mathbf{x})]\mathbf{d} \stackrel{?}{=} [\mathbf{\Theta}(\mathbf{x})]e + [\mathbf{a}]
\end{array}$$

Fig. 1. Generic Σ-protocol for algebraic languages $\mathcal{L}_{\mathbf{M},\mathbf{\Theta}}$ from a distribution $\mathcal{D}_{par}$

The Extended-Kernel Matrix Diffie-Hellman Assumption. To prove the soundness of our NIZK, we introduce a new family of assumptions, which we call the *extended-kernel Matrix Diffie-Hellman* assumption (extKerMDH). The regular KerMDH assumption with respect to a distribution `Dist` over an asymmetric pairing group states that, given a matrix $[\mathbf{A}]_2$ sampled from `Dist`, it is infeasible to find a vector $[\mathbf{v}]_1$ where $\mathbf{v}$ is in the kernel of $\mathbf{A}$. It is a natural computational analogue of the decisional Matrix Diffie-Hellman assumption (which it implies), and was introduced in [63]. Our new assumption further generalizes the KerMDH assumption as follows: it states that it should be infeasible, given $[\mathbf{A}]_2$, to find another matrix $[\mathbf{A}']_2$ and a matrix $[\mathbf{B}]_1$ such that $\mathbf{B}$ spans the entire kernel of $\mathbf{A}||\mathbf{A}'$. Intuitively, the adversary is allowed to extend the matrix $[\mathbf{A}]_2$, which facilitates finding $\mathbb{G}_1$-vectors in its kernel; but each time the adversary extends $\mathbf{A}$ by one column, he must provide an additional $\mathbb{G}_1$-vector (linearly independent of the previous vectors) in the kernel of the extended matrix.

The $\mathsf{extKerMDH}$ assumption is a static, non-interactive assumption, which generalizes the KerMDH assumption in a natural way. To provide further evidence for the security of our assumption, we prove that it is unconditionally secure in the generic group model [70] (GGM), and that it reduces to the discrete logarithm assumption in the algebraic group model [31] (AGM). On the downside, the $\mathsf{extKerMDH}$ assumption might not in general be a falsifiable assumption [36,64]: it states that it is infeasible to output $[\mathbf{A}']_2$ and a *basis* $[\mathbf{B}]_1$ of the kernel of $\mathbf{A}||\mathbf{A}'$, but verifying whether the $\mathbb{G}_1$-matrix $[\mathbf{B}]_1$ is full rank is not efficiently feasible in general (indeed, the hardness of deciding whether a matrix given in a group $\mathbb{G}$ is full rank is exactly the decisional matrix Diffie-Hellman assumption). However, we show that for all *witness-sampleable languages*, there is a language trapdoor which does allow to efficiently check whether $\mathbf{B}$ is full rank (intuitively, the trapdoor allows to put $[\mathbf{B}]_1$ in triangular form, from which the rank can be easily checked), turning our new assumption into a falsifiable assumption.

Witness Samplable Languages. We give an intuition of the class of algebraic languages which satisfy our requirements. Intuitively, an algebraic language $\mathcal{L}$ admits a NIZK (using our compiler) where soundness reduces to a falsifiable assumption if the parameters of $\mathcal{L}$ can be sampled together with a trapdoor which allows to efficiently check language membership. For example, this captures the DDH language $\mathcal{L}_{\mathsf{DDH}}$: given language parameters $([1]_1, [s]_1)$, the words in $\mathcal{L}_{\mathsf{DDH}}$ are of the form $([x]_1, [x \cdot s]_1)$, and the trapdoor s allows to verify that a word (c_1, c_2) belongs to $\mathcal{L}_{\mathsf{DDH}}$ by checking whether $s \cdot c_1 - c_2 = [0]_1$. Witness samplable languages need not be linear languages: for example, the language of ElGamal encryptions (in the exponent) of a plaintext $m \in \{0, 1\}$ is not a linear language, yet the ElGamal secret key allows to efficiently check wether a pair of group elements indeed encrypts a bit, hence it is also captured by our methods. More generally, the conjunctions and disjunctions of witness samplable languages are still witness samplable. On the other hand, some natural algebraic languages are not witness-samplable; for example, the language of triples of the form $([1]_1, [x]_1, [x^2]_1)$ does not seem to be witness samplable (since it is not clear how one could generate a word-independent trapdoor allowing to check membership to this language).

Witness-sampleable languages were originally introduced in [51], but were restricted to linear languages. We extend this notion of witness-sampleability to arbitrary algebraic languages, and will show that many languages of interest are actually witness sampleable. For these languages, we therefore obtain shorter NIZKs under a natural, static, *falsifiable* assumption. We note that for the case of linear languages (such as the language of DDH tuple), our generalized notion of witness-samplability is the same as the notion of [51], and applying our compiler to witness-samplable linear languages leads to NIZKs which are actually secure under the standard KerMDH assumption (while still being shorter than GS proofs).

1.4 Applications

Our new NIZKs have several attractive features and can be used to improve the efficiency of many NIZK-based primitives. We provide a non-exhaustive list of some applications below. All applications we describe rely on witness-sampleable algebraic languages, making the underlying extKerMDH assumption falsifiable.

Adaptive NIZKs for Linear Languages. We achieve the shortest and most efficient adaptive NIZKs for (witness-sampleable) linear languages, with perfect zero-knowledge and computational soundness under the kernel Diffie-Hellman assumption: a Groth-Sahai proof for the language of DDH tuples consists of four group elements, while our NIZK requires only three group elements, and considerably less pairings. We note that in the quasi-adaptive setting, where the common reference string is allowed to depend on the language, the work of [57] gives NIZKs with two group elements (for non witness-sampleable languages), or even a single group element (for witness-sampleable languages). Therefore, our work can be seen as filling a remaining gap, providing a more complete picture of the size of NIZKs for linear languages, depending on whether we allow quasi-adaptive soundness, and rely on witness-sampleability. In addition to providing a stronger soundness guarantee, full adaptivity also leads to increased efficiency when many proofs are run in some high level application: it allows to rely on a single CRS (which, in our case, consists of a *single* group element), even when executing many linear subspace proofs for different languages. In contrast, QA-NIZKs have a language-dependent CRS; hence, a different CRS must be generated for each language. The comparison is summarised in Table 1.

Scheme	Assumption	CRS	Proof size	Pairings	WS	Fully Adaptive
GS [44]	SXDH	4	$4(n+2t)$	$24(n(4t+8))$	✗	✓
KW [57]	KerMDH	$6(n+2t+2)$	2(2)	$3(n+1)$	✗	✗
KW [57]	KerMDH	$4(n+t+1)$	1(1)	$2(n)$	✓	✗
Ours	KerMDH	1	$3(n+t)$	$6(n+nt+2)$	✓	✓

Table 1. Comparison of existing NIZKs for the DDH language (linear languages described by an $n \times t$ matrix). CRS/Proof size denotes the number of group elements in the common reference string/a proof. Pairings denotes the number of pairing operations in proof verification. "WS" indicates whether the proof system is restricted to witness sampleable languages.

Adaptive NIZKs for Disjunctions. Since our NIZKs are built by compiling a Σ-protocol, they are compatible with the OR-trick of [21]. The OR-trick provides a general method to construct Σ-protocols of partial satisfiability, such as "k of those n words belong to the language $\mathcal{L}$", from a Σ-protocol for proving membership to $\mathcal{L}$. Building upon this observation, we obtain shorter NIZKs for disjunctions of statements. The state-of-the-art NIZK for partial satisfiability of equations is the one in [66]. For the important case of the disjunction between two (resp. n) DDH languages, it gives proofs of size 10 group elements under the SXDH assumption (resp. $4n + 2$ group elements for 1-out-of-n proofs). For the same language, our approach leads to proofs of size 7 (resp. $3n + 1$ group elements for 1-out-of-n proofs). This is detailed in Table 2. NIZKs for disjunctions of languages are a core component in several applications; we outline some applications below.

Scheme	Assumption	CRS	Proof size	Pairings	WS
[43, 66]	SXDH	4	$10(\sum_{i=1}^{2} n_i + 2t_i + 2)$	$24(\sum_{i=1}^{2} 4n_i + 2n_it_i)$	✗
Ours	$\mathsf{extKerMDH}$	1	$7(\sum_{i=1}^{2} n_i + t_i + 1)$	$12(\sum_{i=1}^{2} n_i + n_it_i + 4)$	✓

Table 2. Comparison of existing NIZKs for the OR of two DDH languages (two linear languages described by $n_i \times t_i$ matrices for $i \in \{1, 2\}$). CRS denotes the number of group elements in the common reference string. "WS" indicates whether the proof system deals only with witness sampleable languages. Note that our scheme can in fact handle non-witness sampleable languages; however, this comes at the cost of making the underlying $\mathsf{extKerMDH}$ assumption non-falsifiable.

Ring Signatures. Ring signatures [67] allow a signer to anonymously sign on behalf of an ad-hoc group to which it belongs. They are a core component in some e-voting and e-cash schemes [71] and anonymous cryptocurrencies such as Monero [65]. A $O(\sqrt{N})$-size proof of membership in a ring of size N was designed by Chandran, Groth and Sahai [18] and subsequently improved in [66]; it relies at its core on a NIZK for $(\ell - 1)$-out-of-ℓ disjunction of DDH languages. Using our improved NIZK for disjunction, we reduce the ring signature size by $\sqrt{N} - 1$ group elements, for rings of size N.

We observe that a $O(\log N)$-size ring signature scheme was recently introduced in [5]. The authors do not provide a concrete efficiency analysis and use generic tools which would likely render concrete instantiations inefficient for reasonable group sizes. We note, though, that our proof system can be used to instantiate the non-interactive witness indistinguishable proof system they rely upon, and would likely lead to efficiency improvements comparable to what we get over the ring signature of [66], for concrete instantiations of their building blocks.

Tightly-Secure QA-NIZKs with Unbounded Simulation Soundness. In several applications in cryptography, the constructions require a NIZK for linear languages which satisfies a stronger soundness guarantee: soundness should hold even if the adversary is allowed to see an arbitrary number of simulated proofs. This stronger notion is known as *unbounded simulation-soundness*. The recent work of [3] introduced the first unbounded simulation-sound quasi-adaptive NIZK

(USS-QA-NIZK) which achieves simultaneously compact CRS, compact proof size, and a tight security reduction. At the core of their construction is the disjunction NIZK of [66], which has 10 group elements; this accounts for most of the size of their USS-QA-NIZK, which has 14 group elements. By replacing the disjunction proof by our new NIZK, we reduce the size of their USS-QA-NIZK to only 11 group elements, and also reduce the CRS size, at the cost of requiring our new assumption. We provide a comparison to existing USS-QA-NIZKs for linear languages on Table 3. In particular, our result allows to further reduce the size of the tightly-secure IND-mCCA-secure public-key encryption scheme of [4] (IND-mCCA refers to indistinguishability against chosen ciphertext attacks in the multi-user, multi-challenge setting), with a security reduction independent of the number of decryption-oracle requests of the CCA2 adversary, from 17 group elements to 14 group elements.

	CRS Size	Proof Size	Pairings	Sec. Loss	Assumption
[58]	$2n+3(t+\lambda)+10$	20	$2n+30$	$O(Q)$	DLIN
[57]	$(2t+6, n+6)$	$(4,0)$	$t(n+t+2)$	$O(Q)$	SXDH
[59]	$2n+3t+24\lambda+55$	42	$2n+10$	$3\lambda+7$	DLIN
[33]	$(t+6\lambda+1, n+2)$	$(3,0)$	$n+4$	$4\lambda+1$	SXDH
[4]	$(3t+14, n+12)$	$(n+16, 2t+5)$	$7n+5t+3nt+121$	$36\log(Q)$	SXDH
[3]	$(4t+4, 2n+8)$	$(8,6)$	$n+30$	$6\log(Q)$	SXDH
Ours	$(4t+8, 2n+3)$	$(8,3)$	$n+18$	$6\log Q$	SXDH, extKerMDH

Table 3. Comparison of existing unbounded simulation-sound NIZKs for linear languages. The notation (x_1, x_2) denotes x_1 elements in $\mathbb{G}_1$ and x_2 elements in $\mathbb{G}_2$. Q denotes the number of simulation queries, λ is the security parameter. (n, t) are the parameters of the underlying linear language, defined by a matrix $\mathbf{M} \in \mathbb{Z}_p^{n\times t}$, with $n > t$.

Tightly-Secure Structure-Preserving Signatures. The notion of structure-preserving cryptography gives a paradigm for building modular protocols designed to be naturally expressed as systems of pairing-product equations, which makes them compatible with the Groth-Sahai methodology. Structure-Preserving Signatures (SPS) are one of the most fundamental primitives in structure-preserving cryptography. They are the core component in a variety of important applications, such as anonymous credentials (see e.g. [9,14,15,19,23,32,46,61], to name just a few), mixnets and voting systems [41], or simulation-sound NIZKs [40,59].

A cryptographic scheme is tightly secure if its security loss is independent of the number of users of the scheme. A tight security reduction gives guarantees that do not degrade with the size of the setting in which the system is used. Tight security is especially important in structure-preserving cryptography, where many components rely on the same cyclic group: if a non-tightly-secure scheme is used and the number of users increases, this might require increasing the group size to compensate for the security loss, degrading the performance of all other schemes relying on the same cyclic group. There has been a long sequence of works that seeked to obtain increasingly shorter structure preserving signatures with tight security reductions; we summarize them in Table 4.

The work of [34] provides a tightly-secure SPS with 14 group elements, which combines an algebraic MAC scheme with the proof of [66] for the disjunction of two DDH languages. The latter has proof size of 10 group elements. Replacing the OR-NIZK in their work by the shorter proof which we introduce leads to a tightly-secure SPS with 11 group elements, matching the size of the best known tightly-secure SPS [3]. The work of [3] improves over [34] by replacing the underlying OR-NIZK by a *designated-prover* OR-NIZK, which suffices in this context. They show that in the designated-prover setting, the size of the OR-NIZK can be reduced to 7 group elements. We observe that their technique is actually compatible with our improved OR-NIZK, and leads to a quasi-adaptive designated-prover OR-NIZK with only 5 group elements (which can be of independent interest). Overall, this leads to a tightly-secure SPS with only 9 group elements under (a falsifiable flavor of) the extKerMDH assumption, significantly improving over the efficiency of the state-of-the-art. Considering a setting with security parameter $\lambda = 80$, a large possible number of signing queries $Q = 2^{30}$, and choosing a group $\mathbb{G}$ of order $p \approx 2^{2(\lambda + \log L)}$ to account for the security loss of $L(Q)$ (assuming that the best attack on the group is the generic $\sqrt{p}$-time attack), our scheme is actually computationally more efficient than the state-of-the-art non-tightly-secure SPS of [52], and produces signatures which are only slightly larger: 241 Bytes versus 201 Bytes.

Scheme	Sig. Size	PK Size	Pairings	Sec. Loss	Assumption
[47]	$10\ell + 6$	13	$81l + 1$	$O(1)$	DLIN
[1]	$(7, 4)$	$(5, n + 12)$	16	Q	SXDH, XDLIN
[60]	$(10, 1)$	$(16, 2n + 5)$	$17 + 2n$	$O(Q)$	SXDH, XDLINX
[56]	$(6, 1)$	$(0, n + 6)$	$3n + 4$	$2Q^2$	SXDH
[52]	$(5, 1)$	$(0, n + 6)$	$n + 3$	$Q \log Q$	SXDH
[2]	$(13, 12)$	$(18, n + 11)$	$n + 16$	80λ	SXDH
[50]	$(11, 6)$	$(7, n + 16)$	$n + 22$	116λ	SXDH
[34]	$(8, 6)$	$(2, n + 9)$	$n + 11$	$6 \log Q$	SXDH
[4]	$(6, 6)$	$(2, n + 5)$	$7n + 5t + 3nt + 121$	$36 \log Q$	SXDH
[3]	$(7, 4)$	$(2, n + 11)$	$n + 31$	$6 \log Q$	SXDH
Ours	$(7, 2)$	$(7, n + 8)$	$n + 23$	$6 \log Q$	SXDH, extKerMDH

Table 4. Comparison of existing structure-preserving signatures for message space $\mathbb{G}_1^n$, in their most efficient variant. For [4], n and t are defined as in Table 3. The notation (x_1, x_2) denotes x_1 elements in $\mathbb{G}_1$ and x_2 elements in $\mathbb{G}_2$. Q denotes the number of signing queries, λ is the security parameter. In the tree-based scheme of [47], ℓ denotes the depth of the tree (which limits the number of signing queries to 2^ℓ).

1.5 Related Work

We already mentioned related works on NIZKs and SPS. Our work was partly inspired by a line of work initiated in [17, 24], which compiles Σ-protocols into *designated-verifier* NIZKs, by encrypting the challenge with a malleable cryptosystem, and putting the ciphertext in the CRS. The idea of hiding the challenge of an interactive protocol in a CRS was also used in different contexts; for example, it bears similarity with methods used in [35, 53].

1.6 Organization

In Section 2, we recall necessary preliminaries. Section 3 introduces our new NIZK argument system. Section 4 is devoted to the security analysis of the new proof system; to this end, it introduces the notion of algebraic witness sampleability and the extKerMDH assumption. Section 5 extends our construction to disjunctions of algebraic languages. We outline several applications of our results in Section 6. The full version of this paper [20] introduces some missing preliminaries for completeness, together with examples to illustrate some of the notions we introduce, and includes a proof of security of our new assumption in the generic group model and in the algebraic group model. It also describes a variant of our compiler which yields (dual-mode) NIZK *proofs* based on the SXDH assumption for arbitrary algebraic languages, shows how disjunctions of languages are in fact directly captured by the framework of algebraic languages without going through the OR-trick of [21], and gives an application of our compiler to the designated-prover QA-NIZK from [3].

2 Preliminaries

Let $\mathbb{P}$ denote the set of all primes and $\lambda \in \mathbb{N}$ denote the security parameter. A probabilistic polynomial time algorithm (PPT, also denoted *efficient* algorithm) runs in time polynomial in the (implicit) security parameter λ. A function f is *negligible* if for any positive polynomial p there exists a bound $B > 0$ such that, for any integer $k \geq B$, $|f(k)| \leq 1/|p(k)|$. We will write $f(\lambda) \approx 0$ to indicate that f is a negligible function of λ; we also write $f(\lambda) \approx g(\lambda)$ for $|f(\lambda) - g(\lambda)| \approx 0$. For sampling an element according to a distribution or selecting it uniformly random from a (finite) set, we write $p \overset{\$}{\leftarrow} S$. We use the same notation for the output of a probabilistic algorithm. For output y of a deterministic algorithm A on input x, we will also use $y := A(x)$. Matrices will always be bold, uppercase letters and vectors will be bold, lower-case letters. For a matrix $\mathbf{A}$ let $\mathrm{span}(\mathbf{A}) := \{\mathbf{x} | \exists \mathbf{r} : \mathbf{x} = \mathbf{A}\mathbf{r}\}$ and $ker(\mathbf{A}) := \{\mathbf{x} | \mathbf{x}^T\mathbf{A} = 0\}$ the *left* kernel of $\mathbf{A}$. All interactive protocols will be performed between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. If one party can deviate from the protocol, we will denote this by $\hat{\mathcal{P}}$ and $\hat{\mathcal{V}}$ respectively. Additionally, a simulator will be called $\mathcal{S}$. For language parameters ρ sampled from a language distribution $\mathcal{D}$, let $\mathcal{L}_\rho$ denote the language defined by ρ and let R_ρ denote its witness relation. Finally, for a distribution $\mathcal{D}$, we write $\mathsf{Supp}(\mathcal{D})$ for the support of the distribution.

2.1 Groups and Pairings

Throughout this work, let $p \in \mathbb{P}$ denote a prime with bit length polynomial in the security parameter λ. Let $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ be finite groups of prime order p with generators g_1, g_2 respectively and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ a bilinear map. We set $g_T := e(g_1, g_2)$, which is a generator of $\mathbb{G}_T$. $\mathcal{PG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ is called a pairing group setting, if the following properties hold: $e(g_1, g_2) \neq 0_T$ *(non-degenerate)*; $e(ag_1, bg_2) = ab \cdot e(g_1, g_2)$ *(bilinearity)*; and e is efficiently computable. Furthermore, we require the existence of a probabilistic algorithm $PGGen$, which on input 1^λ generates pairing parameters as above with a group order close to 2^λ, i.e. $\mathcal{PG} \xleftarrow{\$} PGGen(1^\lambda)$.

Throughout this work, we will write all groups in implicit notation, i.e. for an additive pairing group setting $\mathcal{PG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$, we write $[1]_i := g_i$ and $[x]_i := x \cdot g_i$ for all $x \in \mathbb{Z}_p$ and $i \in \{1, 2, T\}$. If the group is clear from context, we will omit the index. We write $[x]_1 \bullet [y]_2 := e([x]_1, [y]_2) = [xy]_T$ for pairings. The implicit notation also extends to matrices and vectors. For $\mathbf{A} \in \mathbb{Z}_p^{n\times t}, \mathbf{A} = (a_{ij})$, let $[\mathbf{A}]_k = ([a_{ij}]_k) \in \mathbb{G}_k^{n\times t}$ for $k \in \{1, 2, T\}$ and we also extend the pairing notation from above to $[\mathbf{A}]_1 \bullet [\mathbf{B}]_2 := e([\mathbf{A}]_1, [\mathbf{B}]_2) = [\mathbf{AB}]_T$ for matrices $\mathbf{A} \in \mathbb{Z}_p^{n\times t}, \mathbf{B} \in \mathbb{Z}_p^{t\times m}$. Furthermore, we extend the implicit notation to linear (multivariate) polynomials. Let $\mathcal{P}_l := \{[a_0] + \sum_{i=0}^{l} a_i X_i | a_i \in \mathbb{Z}_p \text{ for } i \in \{0, \ldots, l\}\} \subset \mathbb{G}[\mathbf{x} = (X_1, \ldots, X_l)]$ be the set of linear multivariate polynomials over $\mathbb{G}$ in l variables. For $f \in \mathcal{P}_l$ and $\mathbf{y} = (y_1, \ldots, y_l) \in \mathbb{Z}_p^l$, we define the evaluation of f in $\mathbf{y}$ as applying the group operation in the exponent, i.e.

$$f([\mathbf{y}]) := f(\mathbf{y}) = [a_0] + \sum_{i=1}^{l} a_i[y_i] = [a_0] + \sum_{i=0}^{l} [a_i y_i]$$

This allows us (in a slight abuse of notation) to use polynomials from $\mathcal{P}_l$ inside of matrices and equations in implicit notation without changing variable names, i.e. $[a_0]X_0 = [a_0 X_0]$, since the evaluation of the polynomial is defined exactly that way. For a matrix $\mathbf{A} = (a_{i,j}) \in \mathcal{P}_l^{n\times t}$, the evaluation of the matrix (or vector) over $\mathcal{P}_l$ in a vector $\mathbf{y} \in \mathbb{G}^l$ denotes the evaluation of all entries in the given vector, i.e. $\mathbf{A}(\mathbf{y}) := (a_{i,j}(\mathbf{y})) \in \mathbb{G}^{n\times t}$.

The assumptions used in this work are parametrised over matrix distributions. These are defined as follows.

Definition 1 (Matrix Distribution). *Let $k, l \in \mathbb{N}$ with $k < l$. We call $\mathcal{D}_{k,l}$ a matrix distribution, if it outputs matrices over $\mathbb{G}^{l\times k}$ of full rank k in polynomial time. If $l = k + 1$, we write $\mathcal{D}_k$ instead. Without loss of generality, we assume that the first k rows of a matrix $\mathbf{A} \in \mathsf{Supp}(\mathcal{D}_{k,l})$ form an invertible matrix.*

An example for a matrix distribution for which the KerMDH and MDDH assumptions hold in the AGM is the following:

$$\mathcal{L}_k : \mathbf{M} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ e_1 & 0 & 0 & \cdots & 0 \\ 0 & e_2 & 0 & & 0 \\ 0 & 0 & e_3 & \ddots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & & 0 & e_k \end{bmatrix}$$

For $k = 1$, this distribution generates Diffie-Hellman matrices and for $k \geq 2$ these matrices correspond to the k-Lin assumption [48]. We will only consider the distribution $\mathcal{L}_k$ in this work as it is sufficient for all of our applications.

2.2 Σ-Protocols

A Σ-protocol for an NP language $= \{x : \exists w, |w| = \mathsf{poly}(|x|) \wedge (x, w) \in R\}$ (where R is a polytime checkable relation) is a public-coin, three-move interactive proof between a prover $\mathcal{P}$ with witness w and a verifier $\mathcal{V}$, where the prover sends an initial message $a = P_1(x, w)$, the verifier responds with a random $e \xleftarrow{\$} \{0,1\}^\lambda$ and the prover concludes with a message $d = P_2(x, w, a, e)$. Lastly, the verifier outputs 1, if it accepts and 0 otherwise.

Three properties are required for a Σ-protocol: completeness, special soundness and special honest-verifier zero-knowledge.

Definition 2 (Completeness). *A three-move protocol Π_R for a relation R with prover $\mathcal{P}$ and verifier $\mathcal{V}$ is* complete*, if*

$$Pr\left[\mathsf{out}(V(x,a,e,d)) = 1 \,\middle|\, \begin{matrix} (x,w) \in R, a \xleftarrow{\$} P_1(x,w), \\ e \xleftarrow{\$} \{0,1\}^\lambda, d \xleftarrow{\$} P_2(x,w,a,e) \end{matrix}\right] = 1$$

Definition 3 (Special soundness). *A three-move protocol Π_R for a relation R has the* special soundness *property, if a polynomial time algorithm E exists, which for a statement x and two accepting transcripts $(a, e, d), (a, e', d')$ of Π_R with $e \neq e'$ outputs a witness w, s.t. $(x, w) \in R$ with overwhelming probability.*

Definition 4 (Special honest-verifier zero-knowledge). *A three-move protocol Π_R for a relation R is* special honest-verifier zero-knowledge*, if there exists a polynomial-time simulator $\mathcal{S}$ such that the distributions of $\mathcal{S}(x, e)$ and the transcript of an honest protocol execution between $\mathcal{P}$ and $\mathcal{V}$ are identical for $(x, w) \in R$, $e \in \{0,1\}^\lambda$.*

2.3 Non-interactive Zero-Knowledge Arguments

An adaptive NIZK Π for a family of language distribution $\{\mathcal{D}_{par}\}_{par}$ consists of four probabilistic algorithms:

– $\mathsf{CRSGen}(1^\lambda)$. On input 1^λ generates public parameters par (such as group parameters), a CRS and a trapdoor $\mathcal{T}$. For simplicity of notation, we assume that any group parameters are implicitly included in the CRS.
– $\mathsf{Prove}(\mathsf{CRS}, \rho, x, w)$. On input of a CRS, a language description $\rho \in \mathcal{D}_{par}$ and a statement x with witness w, outputs a proof π for $x \in \mathcal{L}_\rho$.
– $\mathsf{Verify}(\mathsf{CRS}, \rho, x, \pi)$. On input of a CRS, a language description $\rho \in \mathcal{D}_{par}$, a statement and a proof, accepts or rejects the proof.
– $\mathsf{SimProve}(\mathsf{CRS}, \mathcal{T}, \rho, x)$. Given a CRS, the trapdoor $\mathcal{T}$, a language description $\rho \in \mathcal{D}_{par}$ and a statement x, outputs a simulated proof for the statement $x \in \mathcal{L}_\rho$.

Note that the CRS does not depend on the language distribution or language parameters, i.e. we define fully adaptive NIZKs for language distributions.

The following properties need to hold for a NIZK argument (see e.g. [44]).

Definition 5 (Perfect Completeness:). *A proof system Π for a family of language distributions $\{\mathcal{D}_{par}\}_{par}$ is* perfectly complete*, if*

$$Pr\left[\mathsf{Verify}(\mathsf{CRS}, \rho, x, \pi) = 1 \,\middle|\, \begin{array}{c}(par, \mathsf{CRS}, \mathcal{T}) \xleftarrow{\$} \mathsf{CRSGen}(1^\lambda); \rho \in \mathsf{Supp}(\mathcal{D}_{par});\\ (x, w) \in R_\rho; \pi \xleftarrow{\$} \mathsf{Prove}(\mathsf{CRS}, \rho, x, w)\end{array}\right] = 1$$

A proof system is sound, if it is hard to find proofs of incorrect statements. This is captured in the following definition.

Definition 6 (Computational Soundness). *A* NIZK *proof system Π for a family of language distributions $\{\mathcal{D}_{par}\}_{par}$ is* computationally sound*, if for every efficient adversary $\mathcal{A}$*

$$\Pr\left[\begin{array}{c}\mathsf{Verify}(\mathsf{CRS}, \rho, x, \pi) = 1\\ \wedge x \notin \mathcal{L}_\rho\end{array} \,\middle|\, \begin{array}{c}(par, \mathsf{CRS}, \mathcal{T}) \xleftarrow{\$} \mathsf{CRSGen}(1^\lambda);\\ \rho \in \mathsf{Supp}(\mathcal{D}_{par}); (\pi, x) \xleftarrow{\$} \mathcal{A}(\mathsf{CRS}, \rho)\end{array}\right] \approx 0$$

with the probability taken over CRSGen*.*

A proof system is zero knowledge, if it is impossible to distinguish between the output of $\mathsf{SimProve}$ and Prove. This is formalised as follows.

Definition 7 (Perfect Zero Knowledge). *A* NIZK *proof system Π for a family of language distributions $\{\mathcal{D}_{par}\}_{par}$ is called* perfectly zero-knowledge*, if for all λ, all $(par, \mathsf{CRS}, \mathcal{T}) \in \mathsf{Supp}(\mathsf{CRSGen}(1^\lambda))$, all $\rho \in \mathsf{Supp}(\mathcal{D}_{par})$ and all $(x, w) \in R_\rho$, the distributions*

$$\mathsf{Prove}(\mathsf{CRS}, \rho, x, w) \textit{ and } \mathsf{SimProve}(\mathsf{CRS}, \mathcal{T}, \rho, x)$$

are identical.

We can relax the security of a NIZK argument to a Non-Interactive Witness Indistinguishable (NIWI) argument by replacing the zero-knowledge property with the following witness indistinguishability property. Note that unlike NIZKs, which can only exist in the CRS model, NIWIs are possible in the plain model.

Definition 8 (Statistical Witness Indistinguishability). *A proof system* $\Pi = (\mathsf{CRSGen}, \mathsf{Prove}, \mathsf{SimProve}, \mathsf{Verify})$ *for a family of language distributions* $\{\mathcal{D}_{par}\}_{par}$ *is* statistically witness indistinguishable*, if for every adversary* $\mathcal{A}$*, every* λ*, every* $(par, \mathsf{CRS}, \mathcal{T}) \in \mathsf{Supp}(\mathsf{CRSGen}(1^\lambda))$*, all* $\rho \in \mathsf{Supp}(\mathcal{D}_{par})$ *and all* $x \in \mathcal{L}_\rho$ *with witnesses* w_1, w_2*, we have*

$$\begin{aligned}|\Pr[\mathcal{A}(\mathsf{CRS}, \rho, x, \pi) = 1 | \pi \xleftarrow{\$} \mathsf{Prove}(\mathsf{CRS}, \rho, x, w_1)] \\ - \Pr[\mathcal{A}(\mathsf{CRS}, \rho, x, \pi) = 1 | \pi \xleftarrow{\$} \mathsf{Prove}(\mathsf{CRS}, \rho, x, w_2)]| \approx 0\end{aligned}$$

The property adapts to interactive protocols in a natural way.

3 A Pairing-Based Compiler for NIZKs from Σ-Protocols

In this section, we will describe our new approach to pairing-based non-interactive zero-knowledge arguments. Our starting point is a natural Σ-protocol for algebraic languages over abelian groups, which was used (implicitly or explicitly) in previous works [11, 17, 45]. Before describing the protocol and our NIZK construction, we formally introduce algebraic languages.

3.1 Algebraic Languages

We focus on languages that can be described by a set of algebraic equations over an abelian group. More precisely, we will consider languages of the form $\{\mathbf{x} \in \mathbb{G}^l | \exists \mathbf{w} \in \mathbb{Z}_p^t : \mathbf{M}(\mathbf{x}) \cdot \mathbf{w} = \mathbf{\Theta}(\mathbf{x})\}$, where $\mathbf{M} : \mathbb{G}^l \mapsto \mathbb{G}^{n\times t}$ and $\mathbf{\Theta} : \mathbb{G}^l \mapsto \mathbb{G}^n$ are linear maps, which can be sampled efficiently according to a language distribution $\mathcal{D}_{par}$. These languages have been used previously in several works on zero-knowledge proofs and hash proof systems over abelian groups [11, 17, 45], and are quite expressive: they capture a wide variety of languages, including but not limited to, linear and polynomial relations between committed values and the plaintexts of ElGamal-style ciphertexts, or polynomial relations between exponents. We call these languages *algebraic languages*.

It will prove convenient in this work to view the linear maps $\mathbf{M}$ and $\mathbf{\Theta}$ as matrices and vectors over $\mathcal{P}_l$, where $\mathcal{P}_l$ is the set of linear multivariate polynomial in l variables, via the natural extension.

Definition 9 (Algebraic Languages). *Let* $t, l, n \in \mathbb{N}, n > t$ *and* $\mathcal{P}_l := \{[a_0] + \sum_{i=1}^{l} a_i X_i\} \subset \mathbb{G}[\mathbf{X} = (X_1, \ldots, X_l)]$ *the set of linear multivariate polynomials of degree at most 1. Let* $\mathcal{D}_{par}$ *be a distribution that outputs pairs* $(\mathbf{M}, \mathbf{\Theta}) \in \mathcal{P}_l^{n\times t} \times \mathcal{P}_l^n$*. We define the algebraic language* $\mathcal{L}_{\mathbf{M},\mathbf{\Theta}} \subset \mathbb{G}^n$*:*

$$\mathcal{L}_{\mathbf{M},\mathbf{\Theta}} := \{\mathbf{x} \in \mathbb{G}^l | \exists \mathbf{w} \in \mathbb{Z}_p^t : \mathbf{M}(\mathbf{x}) \cdot \mathbf{w} = \mathbf{\Theta}(\mathbf{x})\}$$

where $\mathbf{M}(\mathbf{x})$ *(resp.* $\mathbf{\Theta}(\mathbf{x})$*) denotes the matrix(resp. vector) received by evaluating every entry of* $\mathbf{M}$*(resp.* $\mathbf{\Theta}$*) in the points of* $\mathbf{x}$*.*

Example: Linear Languages. Linear languages, capturing e.g. DDH relations, are obtained as a special case of algebraic languages by restricting $\mathbf{M}(\mathbf{x})$ to be a constant matrix, independent of $\mathbf{x}$ and $\mathbf{\Theta}$ to being the identity. NIZKs for linear languages have been widely studied, see e.g. [51, 57].

Definition 10 (Linear subspace languages). *Let $\mathcal{D}_{par}$ be a parameter distribution that outputs matrices from $\mathbb{G}^{n \times t}$. For $\mathbf{A} \in \mathsf{Supp}(\mathcal{D}_{par})$, we define the language $\mathcal{L}_{\mathbf{A}} := \{\mathbf{x} | \exists \mathbf{w} : \mathbf{A}\mathbf{w} = \mathbf{x}\}$. Specifically, the relation $R_{\mathbf{A}}$ is defined such that $(\mathbf{x}, \mathbf{w}) \in R_{\mathbf{A}} \Leftrightarrow \mathbf{x} = \mathbf{A}\mathbf{w}$. We call $\mathcal{D}_{par}$* witness samplable*, if there is a distribution $\mathcal{D}'_{par}$ which outputs matrices from $\mathbb{Z}_p^{n \times t}$ s.t. the distributions of $\mathbf{A} \overset{\$}{\leftarrow} \mathcal{D}_{par}$ and $[\mathbf{B}] \overset{\$}{\leftarrow} \mathcal{D}'_{par}$ are indistinguishable.*

Effectively, witness-samplability states that the language parameters can be sampled together with a trapdoor matrix $\mathbf{T}$ which allows to check whether $\mathbf{x} \in L$. For linear languages, this trapdoor matrix is simply the exponents of all matrix entries, so the original matrix can be computed from the trapdoor, hence we only sample the latter in the distribution $\mathcal{D}'_{par}$.

***Σ*-Protocol for Algebraic Languages.** We introduce a generic Σ-protocol Π_Σ for algebraic languages on Figure 2.

$$
\begin{array}{ccc}
 & (\mathbf{M}, \mathbf{\Theta}) \in \mathsf{Supp}(\mathcal{D}_{par}) & \\
\mathcal{P} & & \mathcal{V} \\
\mathbf{w}, [\mathbf{x}] & & [\mathbf{x}] \\
\mathbf{r} \overset{\$}{\leftarrow} \mathbb{Z}_p^k & & \\
[\mathbf{a}] := [\mathbf{M}(\mathbf{x})]\mathbf{r} & \xrightarrow{[\mathbf{a}]} & \\
 & \xleftarrow{e} & e \overset{\$}{\leftarrow} \mathbb{Z}_p \\
\mathbf{d} := e\mathbf{w} + \mathbf{r} & \xrightarrow{\mathbf{d}} & \\
 & & \text{check} \\
 & & [\mathbf{M}(\mathbf{x})]\mathbf{d} \overset{?}{=} [\mathbf{\Theta}(\mathbf{x})]e + [\mathbf{a}]
\end{array}
$$

Fig. 2. Σ-protocol Π_Σ for the generic language $\mathcal{L}_{\mathbf{M},\mathbf{\Theta}}$

Theorem 11. *The Σ-protocol Π_Σ is complete, special honest-verifier zero-knowledge and special sound.*

For the proof of Theorem 11 refer to e.g. [62]. We will however recall the special honest-verifier zero-knowledge simulation algorithm $\mathcal{S}_\Pi$, since we need it in our construction. The simulator receives as input $([\mathbf{x}], e)$ and samples $\mathbf{d} \overset{\$}{\leftarrow} \mathbb{Z}_p^t$. Then it sets $[\mathbf{a}] := \mathbf{M}(\mathbf{x})\mathbf{d} - e[\mathbf{\Theta}(\mathbf{x})]$ and returns $([\mathbf{a}], \mathbf{d})$.

3.2 Compiling Π_Σ into a NIZK

The main idea of our construction is to keep the Σ-protocol in group $\mathbb{G}_1$ while moving the challenge e to a group $\mathbb{G}_2$, which admits a bilinear pairing $e : \mathbb{G}_1 \times$

$\mathbb{G}_2 \to \mathbb{G}_T$. This keeps the challenge hidden while allowing verification due to the pairing. For protocol Π_Σ, the compiled NIZK Π_Σ^C is described in Figure 3. We present a detailed security analysis in Section 4.

$\underline{\mathsf{CRSGen}\ (1^\lambda)}$:	$\underline{\mathsf{Prove}\ (\mathsf{CRS}, ([\mathbf{M}]_1, [\mathbf{\Theta}]_1), [\mathbf{x}]_1 \in \mathbb{G}_1^l, \mathbf{w} \in \mathbb{Z}_p^t)}$:
$par := \mathcal{PG} \xleftarrow{\$} PGGen(1^\lambda)$	$\mathbf{r} \xleftarrow{\$} \mathbb{Z}_p^t$
$e \xleftarrow{\$} \mathbb{Z}_p$	$[\mathbf{a}]_1 := [\mathbf{M}(\mathbf{x})]_1 \mathbf{r}$
$\mathsf{CRS} := (\mathcal{PG}, [e]_2)$, $\mathcal{T} := e$	$[\mathbf{d}]_2 := [e]_2 \mathbf{w} + [\mathbf{r}]_2$
return $(par, \mathsf{CRS}, \mathcal{T})$	return $\sigma := ([\mathbf{a}]_1, [\mathbf{d}]_2)$
$\underline{\mathsf{SimProve}\ (\mathsf{CRS}, e, ([\mathbf{M}]_1, [\mathbf{\Theta}]_1), [\mathbf{x}]_1)}$:	$\underline{\mathsf{Verify}\ (\mathsf{CRS}, ([\mathbf{M}]_1, [\mathbf{\Theta}]_1), [\mathbf{x}]_1, \sigma = ([\mathbf{a}]_1, [\mathbf{d}]_2))}$:
$([\mathbf{a}]_1, \mathbf{d}) := \mathcal{S}_\Pi([\mathbf{x}]_1, e)$	check
return $\sigma := ([\mathbf{a}]_1, [\mathbf{d}]_2)$	$[\mathbf{M}(\mathbf{x})]_1 \bullet [\mathbf{d}]_2 \stackrel{?}{=} [\mathbf{\Theta}(\mathbf{x})]_1 \bullet [e]_2 + [\mathbf{a}]_1 \bullet [1]_2$

Fig. 3. Compiled protocol Π_Σ^C, where $\mathcal{S}_\Pi$ denotes the special honest-verifier simulator of Π_Σ and $([\mathbf{M}]_1, [\mathbf{\Theta}]_1) \in \mathcal{P}_l^{n\times t} \times \mathcal{P}_l^n$ is sampled from $\mathcal{D}_{par}$.

3.3 Compiled NIZK as a ZAP

The CRS in our compiled NIZK consists of just one (random) group element from $\mathbb{G}_2$; therefore, our protocol actually works in the common *random* string model. Furthermore, we observe that by allowing the verifier to choose the CRS himself and send it as its first flow, we can transform the NIZK into a statistical ZAP in the *plain model* (*i.e.*, a two-round publicly-verifiable statistical witness-indistinguishable argument system, where the first flow can be reused for an arbitrary (polynomial) number of proofs). We stress that this provides the first known construction of statistical ZAPs from pairing-based assumptions; to our knowledge, the only existing constructions rely on the quasipolynomial hardness of LWE [6, 49]. We can apply the derandomisation technique from [8] to obtain a NIWI argument in the plain model. Since correctness and soundness carry over directly from the NIZK case, it remains to show that our 2-round proof system is witness-indistinguishable. This is shown in Lemma 12.

Lemma 12. *The ZAP resulting from the protocol Π_Σ^C for a family of language distributions $\{\mathcal{D}_{par}\}_{par}$ as described above is perfectly witness indistinguishable.*

Proof. Let $\rho := (\mathbf{M}, \mathbf{\Theta}) \in \mathsf{Supp}(\mathcal{D}_{par})$ and $\mathbf{x} \in \mathcal{L}_\rho$ with two witnesses $\mathbf{w}_1, \mathbf{w}_2$ and let $\hat{\mathcal{V}}$ be a (potentially misbehaving) verifier. Let $[e]_2$ be the CRS (*i.e.*, first flow) chosen by $\hat{\mathcal{V}}$. We have to show that the distributions $\mathsf{Prove}([e]_2, \rho, \mathbf{x}, \mathbf{w}_1)$ and $\mathsf{Prove}([e]_2, \rho, \mathbf{x}, \mathbf{w}_2)$ are indistinguishable. A proof consists of the two vectors $[\mathbf{a}_i]_1 = [\mathbf{M}(\mathbf{x})]_1 \mathbf{r}_i$ and $[\mathbf{d}_i]_2 = [e]_2 \mathbf{w}_i + [\mathbf{r}_i]_2$ for random vectors $\mathbf{r}_i$, witnesses $\mathbf{w}_i$ and e chosen by the verifier. Let $\mathbf{w} := \mathbf{w}_1 - \mathbf{w}_2$. Note that $\mathbf{M}(\mathbf{x})\mathbf{w} = \mathbf{0}$, since $\mathbf{M}(\mathbf{x})\mathbf{w} = \mathbf{M}(\mathbf{x})(\mathbf{w}_1 - \mathbf{w}_2) = \mathbf{\Theta}(\mathbf{x}) - \mathbf{\Theta}(\mathbf{x}) = \mathbf{0}$. For $i = 1$, we have $\pi_1 = ([\mathbf{a}_1]_1 = [\mathbf{M}(\mathbf{x})\mathbf{r}_1]_1, [\mathbf{d}_1]_2 = [e]_2 \mathbf{w}_1 + [\mathbf{r}_1]_2)$. For $i = 2$ and by replacing $\mathbf{w}_2$

with $\mathbf{w}_1 - \mathbf{w}$, we get $\pi_2 = ([\mathbf{a}_2]_1 = [\mathbf{M}(\mathbf{x})]_1 \mathbf{r}_2, [\mathbf{d}_2]_2 = [e]_2 \mathbf{w}_1 + ([\mathbf{r}_2 - e\mathbf{w}]_2))$. Let $\mathbf{r}' := -e\mathbf{w} + \mathbf{r}_2$ and consider a proof using witness $\mathbf{w}_1$ and random vector $\mathbf{r}'$. We get $[\mathbf{a}']_1 = [\mathbf{M}(\mathbf{x})]_1 \mathbf{r}' = [\mathbf{M}(\mathbf{x})]_1 (-e\mathbf{w} + \mathbf{r}_2) = -e [\mathbf{M}(\mathbf{x})]_1 \mathbf{w} + [\mathbf{M}(\mathbf{x})]_1 \mathbf{r}_2 = [\mathbf{M}(\mathbf{x})]_1 \mathbf{r}_2 = [\mathbf{a}_2]_1$ and $[\mathbf{d}']_2 = [e]_2 \mathbf{w}_1 + [\mathbf{r}']_2 = [\mathbf{d}_2]_2$. This is identical to the proof using $\mathbf{w}_2$ and randomness $\mathbf{r}_2$. $\mathbf{r}_1, \mathbf{r}_2$, and $\mathbf{r}'$ are distributed identically (i.e. uniformly random), hence the proof distributions for witness $\mathbf{w}_1$ and $\mathbf{w}_2$ are identical.

4 Security Analysis

4.1 Generalised Witness Samplablility

The definition of witness samplability for linear languages does not carry over to the case of algebraic languages, since only linear languages can be in the span of the kernel of their language trapdoor. To handle this issue, we adapt the witness samplability by requiring the samplability of a language trapdoor $\mathbf{T}$, sampled together with the parameters of the language, which allows to efficiently check the *rank* of $(\mathbf{M}||\mathbf{\Theta})(\mathbf{x})$, which will be full for words *not* in the language, and lower otherwise. We formally define our new notion of algebraic witness samplablility in Definition 13.

Definition 13 (Algebraic Witness Samplability). *Let $t, l, n \in \mathbb{N}$ with $n > t$. An algebraic language distribution $\mathcal{D}_{par}$, outputting pairs $\rho = (\mathbf{M}, \mathbf{\Theta}) \in \mathcal{P}_l^{n \times t} \times \mathcal{P}_l^n$ is called witness samplable, if there exists a second distribution $\mathcal{D}'_{par}$ outputting pairs $(\rho' = (\mathbf{M}', \mathbf{\Theta}'), \mathbf{T}_{\rho'} \in \mathbb{Z}_p^{n \times n})$, with $\mathcal{D}'_{par}(1)$ denoting the distribution of $\mathcal{D}'_{par}$ restricted to the first component, such that the following properties hold.*

1. *The distributions $(\mathcal{D}_{par})$ and $(\mathcal{D}'_{par}(1))$ are identical.*
2. $\mathsf{rank}(\mathbf{T}_{\rho'} \cdot (\mathbf{M}'||\mathbf{\Theta}')(\mathbf{x})) = \begin{cases} t+1 & \mathbf{x} \notin \mathcal{L}_{\rho'} \\ l' < t+1 & \mathbf{x} \in \mathcal{L}_{\rho'} \end{cases}$
3. *$\exists \mathbf{R}, \mathbf{S}$ permutation matrices such that $(\mathbf{R} \cdot \mathbf{T}_{\rho'} \cdot (\mathbf{M}'||\mathbf{\Theta}') \cdot \mathbf{S})(\mathbf{x})$ is an upper triangular matrix*

A family of language distributions $\{\mathcal{D}_{par}\}_{par}$ is witness samplable, if $\mathcal{D}_{par}$ is witness samplable for all possible par.

Note that $\mathbf{R}, \mathbf{S}$ are efficiently computable from $\mathbf{T}_{\rho'} \cdot (\mathbf{M}||\mathbf{\Theta})(\mathbf{x})$ (even without knowledge of $\mathbf{T}_{\rho'}$), as they only rearrange the rows and columns of $\mathbf{T}_{\rho'} \cdot (\mathbf{M}'||\mathbf{\Theta}')(\mathbf{x})$ to a specific form.

The first property states that we can sample a distribution with or without a trapdoor without altering the distribution. The second property is the rank condition itself, which shows language membership. The last property guarantees that the second condition can always be verified in polynomial time. To provide a better intuition of this property, we illustrate it on the language of ElGamal encryptions of a bit (which is a special case of the OR-language for DDH tuples) in the full version of this paper [20].

Definition 14 (Trapdoor Reducibility). *Let $t, l, m, n \in \mathbb{N}$ with $n > t$ and $\mathcal{D}_{par}$ be an algebraic language distribution which outputs pairs $\rho = (\mathbf{M}, \mathbf{\Theta}) \in \mathcal{P}_l^{n\times t} \times \mathcal{P}_l^n$.*

*$\mathcal{D}_{par}$ is m-**trapdoor reducible**, if it is witness samplable with trapdoor distribution $\mathcal{D}'_{par}$ and for every language $(\rho', \mathbf{T}_{\rho'}) \in \mathsf{Supp}(\mathcal{D}'_{par})$, we can instead sample a reducibility trapdoor $\mathbf{T}'_{\rho'} \in \mathbb{Z}_p^{(n-m)\times n}$ such that the following properties hold.*

- *$\mathbf{T}'_{\rho'} \subset \mathbf{T}_{\rho'}$, i.e. the rows of $\mathbf{T}'_{\rho'}$ are a subset of the rows of $\mathbf{T}_{\rho'}$.*
- $\mathsf{rank}(\mathbf{T}'_{\rho'} \cdot (\mathbf{M}||\mathbf{\Theta})(\mathbf{x})) = \begin{cases} n-m & \mathbf{x} \notin \mathcal{L}_{\rho'} \\ m' < n-m & \mathbf{x} \in \mathcal{L}_{\rho'} \end{cases}$
- *m columns of $\mathbf{T}'_{\rho'} \cdot (\mathbf{M}||\mathbf{\Theta})$ are zero-columns and the last column is a non-zero column.*

A family of language distributions $\{\mathcal{D}_{par}\}_{par}$ is trapdoor reducible, if $\mathcal{D}_{par}$ is trapdoor reducible for all possible par.

Trapdoor reducibility captures a stronger notion of witness samplability where in addition to checking the rank of the matrix, we can also reduce the size of the check. Although this is not a necessary property, it allows us to perform reductions to weaker-parametrised assumptions and therefore to strengthen the security guarantees of our constructions for specific language distributions. We illustrate it as well in the full version of this paper [20].

4.2 Extended-Kernel Matrix Diffie-Hellman Assumption

For the linear case, the security of our compiled NIZKs can be reduced to the KerMDH assumption. However for OR-proofs or general algebraic languages, it seems to be insufficient. Hence we propose a generalisation of the KerMDH assumption, which we will call the extKerMDH assumption, and to which we can reduce the soundness of our compiler for all algebraic languages.

Inadequacy of the KerMDH. Before we introduce our new assumption, we want to argue why the existing KerMDH assumption is not sufficient for our application. To do so we give an (informal) example.

For a linear language (described by matrix $\mathbf{A}$), we can reduce soundness to the $\mathcal{L}_1$-KerMDH assumption for matrix distribution $\mathcal{L}_1$ as follows. Suppose that a verifier in the Σ-protocol for a linear language(Figure 2) sends e as its challenge. Then the verification equation is $[\mathbf{Ad}] = [\mathbf{x}]e + [\mathbf{a}]$. If $\mathbf{A}$ is from a witness samplable distribution, we can use the trapdoor to find a vector $\mathbf{t}$ in the kernel of $\mathbf{A}$, i.e. $\mathbf{t}\cdot\mathbf{A} = 0$. Multiplying the above equation with $\mathbf{t}$ then yields $0 = [\mathbf{tx}]e + [\mathbf{ta}]$ and if $\mathbf{x}$ and $\mathbf{a}$ are not in the span of $\mathbf{A}$, we have a non-zero vector in the kernel of $\left[\begin{smallmatrix} 1 \\ e \end{smallmatrix}\right]_2$, namely $\left(\begin{smallmatrix} \mathbf{ta} \\ \mathbf{tx} \end{smallmatrix}\right)^T$ and therefore a solution to the KerMDH problem for $\left[\begin{smallmatrix} 1 \\ e \end{smallmatrix}\right]_2 \in \mathsf{Supp}(\mathcal{L}_1)$. However for the simple binary OR proof from [21] (see Figure 4 for more details), this approach already fails. Instead of one such equation, we get two equations of the form $0 = [\mathbf{t}_i\mathbf{x}_i]e_i + [\mathbf{t}_i\mathbf{a}_i]$ for $i \in \{0, 1\}$ and with $e = e_0 + e_1$.

Since the two vectors consist of group elements, we can't combine them to a single solution for the matrix $\left[\begin{smallmatrix}1\\e\end{smallmatrix}\right]_2$. However, what we obtain are two linearly independent vectors in the kernel of $[1, e, e_0]_2^\intercal$, namely $v_1 = [\mathbf{t}_0\mathbf{a}_0, 0, \mathbf{t}_0\mathbf{x}_0]^\intercal$ and $v_2 = [\mathbf{t}_1\mathbf{a}_1, \mathbf{t}_1\mathbf{x}_1, -\mathbf{t}_1\mathbf{x}_1]^\intercal$. We assume that such a relation is also hard to compute and we formalise it as the extKerMDH assumption.

The extKerMDH Assumption.

Definition 15 ($\mathcal{D}_k$-l-extended Kernel Diffie-Hellman Assumption ($\mathcal{D}_k$-l-extKerMDH)). *Let $l, k \in \mathbb{N}$, $\mathcal{PG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \xleftarrow{\$} \mathrm{PGGen}(1^\lambda)$ and $\mathcal{D}_k$ be a matrix distribution. The $\mathcal{D}_k$-l-extKerMDH assumption holds in $\mathbb{G}_s$ relative to* PGGen*, if for all efficient adversaries $\mathcal{A}$, the following probability is negligible.*

$$\Pr\left[\begin{array}{c|c} [\mathbf{C}]_{3-s} \in \mathbb{G}_{3-s}^{l+1\times k+l+1} \wedge [\mathbf{B}]_s \in \mathbb{G}_s^{l\times k} & \mathcal{PG} \xleftarrow{\$} \mathrm{PGGen}(1^\lambda), \mathbf{D} \xleftarrow{\$} D_k \\ \wedge [\mathbf{C}]_{3-s} \bullet [\mathbf{D}']_s = 0 & ([\mathbf{C}]_{3-s}, [\mathbf{B}]_s) \xleftarrow{\$} \mathcal{A}(\mathcal{PG}, [\mathbf{D}]_s) \\ \wedge\ \mathsf{rank}(\mathbf{C}) \geq l+1 & [\mathbf{D}']_s := \left[\begin{smallmatrix}\mathbf{D}\\ \mathbf{B}\end{smallmatrix}\right]_s \end{array}\right]$$

The probability is taken over then randomness of $\mathcal{A}$, $\mathcal{D}_k$ and PGGen*.*

If in addition to the rank condition, $\mathbf{C}$ is also required to be an upper triangular matrix (in which case the bound on the rank can be verified in polynomial time), the assumption is called falsifiable *$\mathcal{D}_k$-l-extKerMDH.*

This assumption is to the best of our knowledge new and so we want to give an intuition on why we deem it reasonable. First, it is a natural extension of the KerMDH assumption. We give the adversary more freedom by allowing it to extend the given matrix but require it to output multiple, linearly independent vectors in the kernel. As long as the number of linearly independent vectors is strictly larger than the number of vectors the adversary gets to add, breaking the assumption requires finding vectors in $\mathbb{G}_1$ which depend on $[\mathbf{M}]_2$ in a non-trivial way. Second, it is a static family of assumptions (as opposed to Q-type assumptions; once $[\mathbf{M}]_2$ is fixed, our proof system will rely on an extKerMDH assumption with fixed parameters). Third, we consider the issue of falsifiability. It turns out that the extKerMDH assumption is not always falsifiable: to check the given matrix $\mathbf{C}$ for being a basis, one must break a DDH-like problem. However in many concrete cases of interest (formally, each time we will consider *witness samplable* languages), the matrix $\mathbf{C}$ can be brought in an upper triangular form where the rank will be visible and we can instead reduce the security to the *falsifiable* variant. Eventually, the assumption is unconditionally secure in the Generic Group Model (GGM) and can be reduced to the discrete logarithm problem in the Algebraic Group Model (AGM). For the proofs, refer to the full version of this paper [20].

4.3 Security Proof

With the two definitions and the new extKerMDH assumption, we can now finally prove the security of our construction.

Theorem 16. *1. The protocol Π_{Σ}^{C} described in Figure 3 is a* NIZK *argument for any algebraic language distribution $\mathcal{D}_{par}$ outputting pairs $\rho = (\mathbf{M}, \mathbf{\Theta}) \in \mathcal{P}_l^{n \times t} \times \mathcal{P}_l^{n}$, if the $\mathcal{L}_1$-t-extKerMDH assumption holds in $\mathbb{G}_2$ relative to* PGGen.

2. If the language distribution is witness samplable with trapdoors $\mathbf{T}_\rho \in \mathbb{Z}_p^{n \times n}$, then it is a NIZK *argument if the* falsifiable *$\mathcal{L}_1$-t-extKerMDH holds in $\mathbb{G}_2$ relative to* PGGen.

3. If the language distribution is m-trapdoor reducible, then it is a NIZK *argument if the* falsifiable *$\mathcal{L}_1$-$(t-m)$-extKerMDH holds in $\mathbb{G}_2$ relative to* PGGen.

Proof. To prove theorem 16, we have to show *completeness*, *perfect zero knowledge* and *computational soundness*. The first two properties are identical for all parts of the theorem. For the second and third part, the witness samplability and the trapdoor reducibility directly imply the soundness statements, if soundness holds in the first part.

Perfect Completeness: Let $\rho = (\mathbf{M}, \mathbf{\Theta}) \in \mathsf{Supp}(\mathcal{D}_{par})$. If $\mathbf{\Theta}(\mathbf{x}) = \mathbf{M}(\mathbf{x}) \cdot \mathbf{w}$ and $\mathbf{a} = \mathbf{M}(\mathbf{x}) \cdot \mathbf{r}$, we get

$$\begin{aligned}
[\mathbf{M}(\mathbf{x})]_1 \bullet [\mathbf{d}]_2 = &\quad [\mathbf{M}(\mathbf{x}) \cdot \mathbf{d}]_T \\
= &\quad [\mathbf{M}(\mathbf{x}) \cdot (e \cdot \mathbf{w} + \mathbf{r})]_T \\
= &\quad [\mathbf{M}(\mathbf{x}) \cdot \mathbf{w} \cdot e]_T + [\mathbf{M}(\mathbf{x})\mathbf{r} \cdot 1]_T \\
= &\quad [\mathbf{\Theta}(\mathbf{x}) \cdot e]_T + [\mathbf{a} \cdot 1]_T \ \text{ (since } \mathbf{\Theta}(\mathbf{x}) = \mathbf{M}(\mathbf{x}) \cdot \mathbf{w}) \\
= &\quad [\mathbf{\Theta}(\mathbf{x})]_1 \bullet [e]_2 + [\mathbf{a}]_1 \bullet [1]_2
\end{aligned}$$

Perfect Zero Knowledge: We have to show that the distributions Prove and $\mathsf{SimProve}$ are identical. This directly follows from the perfect honest-verifier zero-knowledge property of the Σ-protocol, since we use its simulator in $\mathsf{SimProve}$.

Computational Soundness: We will show that Π_{Σ}^{C} is computationally sound, if the $\mathcal{L}_1$-t-extKerMDH holds in $\mathbb{G}_2$ relative to *PGGen*. Assume an adversary $\mathcal{A}$ which forges a proof for Π_{Σ}^{C} with non-negligible probability. We will construct an adversary $\mathcal{B}$ against the $\mathcal{L}_1$-t-extKerMDH assumption, that uses adversary $\mathcal{A}$ and has the same success probability. $\mathcal{B}$ receives a challenge $\left[\begin{smallmatrix} 1 \\ e \end{smallmatrix}\right]_2$ from its challenger. $\mathcal{B}$ then sets $[e]_2$ as the CRS and samples language parameters $\rho \xleftarrow{\$} \mathcal{D}_{par}$. Now $\mathcal{B}$ runs $\mathcal{A}(\mathsf{CRS}, \rho)$ and receives a statement $\mathbf{x}$ and a proof $\pi = ([\mathbf{a}]_1, [\mathbf{d}]_2)$ which are accepting with non-negligible probability, i.e.

$$\begin{aligned}
[\mathbf{M}(\mathbf{x})]_1 \bullet [\mathbf{d}]_2 &= [\mathbf{\Theta}(\mathbf{x})]_1 \bullet [e]_2 + [\mathbf{a}]_1 \bullet [1]_2 \\
0 &= [\mathbf{a}]_1 \bullet [1]_2 + [\mathbf{\Theta}(\mathbf{x})]_1 \bullet [e]_2 - [\mathbf{M}(\mathbf{x})]_1 \bullet [\mathbf{d}]_2 \\
0 &= [\mathbf{a} || \mathbf{\Theta}(\mathbf{x}) || - \mathbf{M}(\mathbf{x})]_1 \bullet \left[\, 1 \ e \ \mathbf{d} \,\right]_2^{\intercal}
\end{aligned}$$

If $\mathbf{C} := (\mathbf{a} || \mathbf{\Theta}(\mathbf{x}) || - \mathbf{M}(\mathbf{x}))$ has at least $\mathsf{rank}(\mathbf{C}) = t + 1$, then $([\mathbf{C}]_1, [\mathbf{d}]_2)$ is a solution for the assumption, since $\mathbf{d}$ has length t. This can be seen with simple linear algebra.

We know that $\mathbf{M}(\mathbf{x})$ has full rank t. By adding the two columns $\mathbf{a}$ and $\mathbf{\Theta}(\mathbf{x})$, the rank cannot decrease. Assume $\mathbf{\Theta}(\mathbf{x})$ and $\mathbf{a}$ are not in the span of $\mathbf{M}$, i.e. $\mathcal{A}$ did produce a forgery. Then $\mathbf{a}$ and $\mathbf{\Theta}(\mathbf{x})$ are completely independent of $\mathbf{M}$

and therefore the rank of the matrix will be increased by at least 1 and $\mathcal{B}$ has a solution. For a regular proof however, $\mathbf{a}$ and $\mathbf{\Theta}(\mathbf{x})$ are in the span of $\mathbf{M}(\mathbf{x})$ and therefore linearly dependant on the columns of $\mathbf{M}(\mathbf{x})$, therefore the rank can not increase. This shows that $\mathbf{C}$ is full rank if and only if $\mathcal{A}$ outputs a valid forgery and $\mathcal{B}$ wins in this exact case.

For the second part of the theorem, $\mathcal{B}$ samples $(\rho, \mathbf{T}_\rho)$ from the trapdoor distribution $\mathcal{D}'_{par}$, which is by definition indistinguishable from sampling ρ regularly. The statement is seen exactly as the first one except for a multiplication with the trapdoor matrix, which yields a full rank matrix in upper triangular form if and only if the given word is not in the language and we get a falsifiable solution. For the third part, $\mathcal{B}$ samples $(\rho, \mathbf{T}'_\rho)$ from the trapdoor reducibility distribution (which is again indistinguishable from regular sampling) and $\mathcal{B}$ takes the matrix received by the multiplication with the reducibility trapdoor. By removing the zero columns and removing the corresponding elements from $\mathbf{d}$, $\mathcal{B}$ can reduce $\mathbf{d}$'s size by exactly m and therefore gets a solution to the $\mathcal{L}_1$-$(t-m)$-$\mathsf{extKerMDH}$.

5 Extension to Disjunctions of Languages

In this section, we will show how to obtain efficient OR-proofs by applying our compiler to the generic Σ-protocols for k-out-of-n disjunctions of [21].

We briefly recall the method of [21] (for concreteness, we focus on 1-out-of-2 proofs; the general case is similar). It starts from two Σ-protocols for memberships into languages $\mathcal{L}_{\rho_0}, \mathcal{L}_{\rho_1}$, and produces a Σ-protocol for the language $\mathcal{L}_{\rho_0 \vee \rho_1}$. Consider a prover knowing a witness w for $\mathbf{x}_i \in \mathcal{L}_{\rho_i}$ but not for $\mathbf{x}_{1-i} \in \mathcal{L}_{\rho_{(1-i)}}$. The prover chooses a random $e_{1-i} \overset{\$}{\leftarrow} \mathbb{Z}_p$ and uses the special honest-verifier zero-knowledge simulation algorithm to generate $[a_{1-i}], d_{1-i}$ which form an accepting proof for $\mathbf{x}_{1-i} \in \mathcal{L}_{\rho_{(1-i)}}$. Additionally it computes an honest commitment $[a_i]$ for the Σ-protocol for $\mathcal{L}_{\rho_i}$ and sends $[a_0], [a_1]$ to the verifier, which returns a challenge e. The prover now sets $e_i := e - e_{1-i}$ and continues the honest protocol for $\mathbf{x}_i \in \mathcal{L}_{\rho_i}$, calculating d_i and concludes the protocol by sending d_0, d_1 and e_0. The verifier can then calculate $e_1 := e - e_0$ and check both proofs. This protocol can be seen in Figure 4. While this does not immediately fit into the framework of Section 3, our approach is still applicable: The prover again chooses a challenge $e_{1-i} \overset{\$}{\leftarrow} \mathbb{Z}_p$ and simulates the first proof as in the interactive variant and gets the second challenge for the honest proof part only in $\mathbb{G}_2$ as $[e_i]_2 = [e]_2 - ([1]_2\, e_{1-i})$. In addition to the two regular proofs, we have to include $[e_0]_2$ in the proof. This is illustrated in Figure 5. We get the following new, efficient OR-proof.

Theorem 17. *Let $\mathcal{D}^{(0)}_{par}, \mathcal{D}^{(1)}_{par}$ be two algebraic language distributions outputting matrices of dimension $n_0 \times t_0$ and $n_1 \times t_1$ respectively. Applying the construction from Figure 5 yields a fully adaptive* NIZK *argument for the OR-language of $\rho_0 \in \mathsf{Supp}(\mathcal{D}^{(0)}_{par}), \rho_1 \in \mathsf{Supp}(\mathcal{D}^{(1)}_{par})$ of size $n_0 + n_1 + t_0 + t_1 + 1$, if the $\mathcal{L}_1 - (n_0 + n_1 + 1) -$ extKerMDH assumption holds in $\mathbb{G}_2$.*

If both language distributions are witness samplable, the above holds for the falsifiable *$\mathcal{L}_1 - (n_0 + n_1 + 1) -$ extKerMDH assumption.*

If $\mathcal{D}^{(0)}_{par}$ resp. $\mathcal{D}^{(1)}_{par}$ is m_0- resp. m_1-trapdoor reducible, the above holds for the $\mathcal{L}_1 - (n_0 - m_0 + n_1 - m_1 + 1) -$ extKerMDH assumption.

The proof for Theorem 17 is almost identical to one for Theorem 16. The only difference lies in the soundness proof, where we apply the witness samplability (trapdoor reducibility) trapdoors of each language to the respective proofs separately and then combine the results by expressing e_1 as $e - e_0$.

$$\mathbf{M}_0 \xleftarrow{\$} \mathcal{D}^{(0)}_{par} \qquad \mathbf{M}_1 \xleftarrow{\$} \mathcal{D}^{(1)}_{par}$$

$$\begin{array}{ccc}
\mathcal{P} & & \mathcal{V} \\
([\mathbf{x}_0], [\mathbf{x}_1], \mathbf{w}) & & ([\mathbf{x}_0], [\mathbf{x}_1]) \\
\mathbf{r}_i \xleftarrow{\$} \mathbb{Z}_p^t & & \\
[\mathbf{a}_i] := \mathbf{M}_i \mathbf{r}_i & & \\
e_{1-i} \xleftarrow{\$} \mathbb{Z}_p & & \\
([\mathbf{a}_{1-i}], \mathbf{d}_{1-i}) \xleftarrow{\$} \mathcal{S}_{\mathbf{M}_{1-i}}([\mathbf{x}_{1-i}], e_{1-i}) & \xrightarrow{[\mathbf{a}_0], [\mathbf{a}_1]} & \\
 & \xleftarrow{e} & e \xleftarrow{\$} \mathbb{Z}_p \\
e_i = e - e_{1-i} & & \\
\mathbf{d}_i := e_i \mathbf{w} + \mathbf{r}_i & \xrightarrow{\mathbf{d}_0, \mathbf{d}_1, e_0} & e_1 = e - e_0 \\
 & & \text{check} \\
 & & [\mathbf{M}_i]\mathbf{d}_i \stackrel{?}{=} [\mathbf{x}_i] e_i + [\mathbf{a}_i] \\
 & & i \in \{0, 1\}
\end{array}$$

Fig. 4. Sigma protocol $\Pi_{\mathbf{M}_0 \vee \mathbf{M}_1}$ for the or language $\mathcal{L}_{\mathbf{M}_0 \vee \mathbf{M}_1}$ from [21]

<u>CRSGen (1^λ):</u>

$par := \mathcal{PG} \xleftarrow{\$} PGGen(1^\lambda)$
$e \xleftarrow{\$} \mathbb{Z}_p$
$\mathsf{CRS} := (\mathcal{PG}, [e]_2)$, $\mathcal{T} := e$
return $(par, \mathsf{CRS}, \mathcal{T})$

<u>Prove $(\mathsf{CRS}, ([\mathbf{M}_i]_1)_{i \in \{0,1\}}, [\mathbf{x}_0]_1, [\mathbf{x}_1]_1, \mathbf{w})$:</u>

$// \ \mathbf{x}_i = \mathbf{M}_i \mathbf{w}$
$\mathbf{r}_i \xleftarrow{\$} \mathbb{Z}_p^{t_i}$, $e_{1-i} \xleftarrow{\$} \mathbb{Z}_p$
$[e_i]_2 = [e]_2 - [e_{1-i}]_2$
$([\mathbf{a}_i]_1, [\mathbf{d}_i]_2) := ([\mathbf{M}_i]_1 \mathbf{r}_i, [e_i]_2 \mathbf{w} + [\mathbf{r}_i]_2)$
$([\mathbf{a}_{1-i}]_1, \mathbf{d}_{1-i}) \xleftarrow{\$} \mathcal{S}_{\mathbf{M_{1-i}}}([\mathbf{x}_{1-i}]_1, e_{1-i})$
return $\pi := ([\mathbf{a}_0]_1, [\mathbf{a}_1]_1, [\mathbf{d}_0]_2, [\mathbf{d}_1]_2, [\mathbf{e}_0]_2)$

<u>SimProve $(\mathsf{CRS}, \mathcal{T} = e, ([\mathbf{M}_i]_1)_{i \in \{0,1\}}, [\mathbf{x}_0]_1, [\mathbf{x}_1]_1)$:</u>

$e_0 \xleftarrow{\$} \mathbb{Z}_p$
$e_1 = e - e_0$
$([\mathbf{a}_j]_1, \mathbf{d}_j) \xleftarrow{\$} \mathcal{S}_{\mathbf{M}_j}([\mathbf{x}_j]_1, e_j)$ for $j \in \{0, 1\}$
return $\pi := ([\mathbf{a}_0]_1, [\mathbf{a}_1]_1, [\mathbf{d}_0]_2, [\mathbf{d}_1]_2, [e_0]_2)$

<u>Verify $(\mathsf{CRS}, ([\mathbf{M}_i]_1)_{i \in \{0,1\}}, [\mathbf{x}_0]_1, [\mathbf{x}_1]_1, \pi)$:</u>

parse $\pi = ([e_0]_2, ([\mathbf{a}_i]_1, [\mathbf{d}_i]_2)_{i \in \{0,1\}}))$
$[e_1]_2 = [e]_2 - [e_0]_2$
for $j \in \{0, 1\}$ check
$\quad [\mathbf{M}_j]_1 \bullet [\mathbf{d}_j]_2 \stackrel{?}{=} [\mathbf{x}_j]_1 \bullet [e_j]_2 + [\mathbf{a_j}]_1 \bullet [1]_2$

Fig. 5. Compiled protocol $\Pi^c_{\mathbf{M}_0 \vee \mathbf{M}_1}$. $\mathcal{S}_{(\cdot)}$ denotes the SHVZK-simulator for the respective language.

The construction naturally extends to the 1 out of n setting by letting the prover choose $n-1$ challenges itself and setting the last as the difference of e and the sum of all chosen challenges. With n matrices $\mathbf{M}_i$, we get the following size, as the prover has to send $n-1$ challenges to uniquely determine the last challenge: $\sum_{i=1}^{n}(n_i+t_i)+n-1$. In the special case of the disjunction of two DDH languages (as needed in e.g [34]), the compiled OR-trick yields a proof with 7 group elements. The construction can easily be adapted to the setting of k-out-of-n disjunctions, by using a threshold secret sharing (e.g. [69]) to force the adversary to choose at most $n-k$ challenges by itself. Our compiler can be applied in the same way as for the 1 out of n setting and yields NIZK arguments of size $\sum_{i=1}^{n}(n_i+t_i)+n-k$.

6 Applications

6.1 NIZK for linear languages

Let $\mathbb{G}$ be a finite group of prime order p and $\mathcal{D}_{k,n}$ be a matrix distribution. We apply our compiler to the standard Σ-protocol for membership to the linear language generated by $\mathbf{A}$, for $\mathbf{A} \in \mathsf{Supp}(\mathcal{D}_{k,n})$. This protocol was formally analyzed in [62]. Applying our compiler yields the protocol $\Pi^{C}_{\mathbf{A}}$ shown in Figure 6.

$\underline{\mathsf{CRSGen}\ (1^\lambda):}$	$\underline{\text{Prove}(\mathsf{CRS}, [\mathbf{A}]_1, [\mathbf{x}]_1, \mathbf{w}):}$
$par := \mathcal{PG} \xleftarrow{\$} PGGen(1^\lambda)$	$\mathbf{r} \xleftarrow{\$} \mathbb{Z}_p^k$
$[e]_2 \xleftarrow{\$} \mathbb{Z}_p$	$[\mathbf{a}]_1 := [\mathbf{A}]_1\,\mathbf{r}$
$\mathsf{CRS} := (\mathcal{PG}, [e]_2),\ \mathcal{T} := e$	$[\mathbf{d}]_2 := [e]_2\,\mathbf{w} + [\mathbf{r}]_2$
return $(par,\ \mathsf{CRS},\ \mathcal{T})$	return $\pi := ([\mathbf{a}]_1, [\mathbf{d}]_2)$
$\underline{\text{SimProve}(\mathsf{CRS}, \mathcal{T}=e, [\mathbf{A}]_1, [\mathbf{x}]_1):}$	$\underline{\text{Verify}(\mathsf{CRS}, [\mathbf{A}]_1, [\mathbf{x}]_1, \pi = ([\mathbf{a}]_1, [\mathbf{d}]_2)):}$
$\mathbf{d} \xleftarrow{\$} \mathbb{Z}_p^k$	check $[\mathbf{A}]_1 \bullet [\mathbf{d}]_2 \stackrel{?}{=} [\mathbf{x}]_1 \bullet [e]_2 + [\mathbf{a}]_1 \bullet [1]_2$
$[\mathbf{a}]_1 := [\mathbf{A}]_1\,\mathbf{d} - [\mathbf{x}]_1\,e$	
return $\pi := ([\mathbf{a}]_1, [\mathbf{d}]_2)$	

Fig. 6. Compiled protocol $\Pi^{c}_{\mathbf{A}}$.

Theorem 18. *Protocol $\Pi^{c}_{\mathbf{A}}$ is a* NIZK *for the language $\mathcal{L}_{\mathbf{A}}$, if $\mathcal{D}_{k,n}$ is witness samplable and the $\mathcal{L}_1$-kerMDH (= $\mathcal{L}_1$-0-$\mathsf{extKerMDH}$) assumption holds in $\mathbb{G}_2$ relative to $\mathcal{PG}$.*

Proof. If we show that $\mathbf{A}$ is k-trapdoor reducible, then the proof follows from Theorem 16. It is easy to see that all witness samplable matrix distributions $\mathcal{D}_{k,n}$ are k-trapdoor reducible. We sample $\mathbf{A} \in \mathbb{Z}_p^{n\times k}$ and compute an element in the kernel of $\mathbf{A}$, which is exactly the reducibility trapdoor.

The construction above includes proofs for DDH tuples, the Schnorr protocol [68], and general linear subspace membership. We compare our construction instantiated for the DDH language (and asymptotic) with the Groth-Sahai framework [44] and the Kiltz-Wee proofs [57] on Table 1 from Section 1. Our construction is more efficient than Groth-Sahai both in terms of proof size as well as CRS size. For the verification, we also need less pairings (6 versus 24 for Groth-Sahai). Of course this comes at the (mild) cost of assuming witness sampleability of the language (Note that the security is based on the standard kerMDH assumption). On the other hand, our proofs are longer than the proofs from [57] (linear versus constant size). However our construction yields fully adaptive zero-knowledge arguments, while theirs yields quasi-adaptive zero-knowledge arguments and our CRS size is constant while theirs is linear. Our construction closes a gap in characterizing the efficiency of NIZKs for linear languages (with or without witness sampleability, with or without full adaptivity).

6.2 Disjunction of DDH Languages and Tight USS-QA-NIZKs

Using the construction of Section 5, we obtain a NIZK for the disjunction of two DDH languages with only 7 group elements. This is three group elements less than the best previously known NIZK for this language [66]. We provide a self-contained description of the resulting proof system in Figure 5. As discussed in the introduction, combining this proof with the result of [3], we obtain shorter tightly-secure QA-NIZKs with unbounded simulation-soundness (11 versus 14 group elements) and shorter IND-mCCA-secure PKE with tight security reduction (14 versus 17 group elements).

6.3 Tightly-Secure Structure-Preserving Signatures

NIZKs arguments are an important building block in structure-preserving signatures (SPS). Since our constructions yield shorter NIZK arguments for OR-proofs and (in the fully adaptive case) for linear subspaces, substituting existing proofs with our constructions directly improves various SPS schemes. For example, Gay *et.al* [34] use an Or-proof for two DDH languages in their construction; using our OR-proof reduces the size of their tightly-secure SPS from 14 group elements to 11.

The same size was achieved recently by Abe et.al. [3]. They use a new approach in describing the used OR-language as a conjunction statement and build a designated-prover quasi-adaptive NIZK from this formalization, which is shorter than the (publicly-verifiable) OR-proof of [66] (7 versus 10 group elements). We notice that their OR-proof is compatible with our compiler, which allows us to reduce its size down to 5 group elements. This in turn reduces the size of the SPS by 2, resulting in a size of 9 group elements per signature. The exact construction is shown in the full version of this paper [20] and yields the following lemma:

Lemma 19. *There exists a structure-preserving signature scheme which reduces to the SXDH assumption and the $\mathcal{L}_1$-1-extKerMDH assumption with security loss $6\log Q$, where Q is the number of signing queries, with a signature size is 9 group elements and a public key size of $n+15$ elements (for length-n messages).*

A comparison of the resulting SPS to existing schemes can be found in Table 4. We note that our tight SPS can be converted into a bilateral tight SPS (where messages can be from both $\mathbb{G}_1$ and $\mathbb{G}_2$) using the generic transform of [56], leading to a bilateral tightly-secure SPS of size 12 group elements (versus 14 for the best known bilateral tight SPS [3]).

6.4 Ring Signatures

An example of the use of k-out-of-n OR-proofs is the construction of sublinear ring signatures in the standard model [18, 66]. The two constructions produce signatures of size $\mathcal{O}(\sqrt{N})$, where N is the size of the ring. The previous works reduce the size of the signature by rearranging the list of potential signers in the ring into a square matrix, and commit to two bit vectors of weight one, where one denotes the row and the other the column of the used key. Then, a NIZK is used to show that the signature was produce with the key at the corresponding (committed) coordinates in the matrix. This NIZK requires at its core a proof that the two vectors are actually bit vectors and sum to 1. This can be rephrased as an $(n-1)$-out-of-n proof of opening of the commitments to 0, together with a proof that their sum opens to one. The proof given in [66] requires $4 \cdot \sqrt{N}$ group elements, while our proof from Section 5 only requires $3 \cdot \sqrt{N} + 1$ group elements.

We note that there are also constructions achieving sizes of $\sqrt[3]{N}$ [39], however these constructions are not compatible with our NIZK arguments, as they require proofs of knowledge, which our construction does not provide. Furthermore, the constant factors of the construction are quite large, so improving $\sqrt{N}$ constructions might still be useful.

Acknowledgements. We would like to thank Dennis Hofheinz for discussions and contributions to early stages of this work, Eike Kiltz for helpful comments and Carla Ràfols for discussion and pointers.

The first author was supported by ERC Project PREP-CRYPTO (724307), and the second author was supported by the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy (EXC 2092 CASA), and the German Federal Ministry of Education and Research (BMBF) iBlockchain project.

References

1. Abe, M., Chase, M., David, B., Kohlweiss, M., Nishimaki, R., Ohkubo, M.: Constant-size structure-preserving signatures: Generic constructions and simple assumptions. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 4–24. Springer, Heidelberg (Dec 2012)
2. Abe, M., Hofheinz, D., Nishimaki, R., Ohkubo, M., Pan, J.: Compact structure-preserving signatures with almost tight security. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 548–580. Springer, Heidelberg (Aug 2017)

3. Abe, M., Jutla, C.S., Ohkubo, M., Pan, J., Roy, A., Wang, Y.: Shorter QA-NIZK and SPS with tighter security. Asiacrypt 2019 (2019)
4. Abe, M., Jutla, C.S., Ohkubo, M., Roy, A.: Improved (almost) tightly-secure simulation-sound QA-NIZK with applications. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part I. LNCS, vol. 11272, pp. 627–656. Springer, Heidelberg (Dec 2018)
5. Backes, M., Döttling, N., Hanzlik, L., Kluczniak, K., Schneider, J.: Ring signatures: Logarithmic-size, no setup - from standard assumptions. In: Rijmen, V., Ishai, Y. (eds.) EUROCRYPT 2019, Part III. pp. 281–311. LNCS, Springer, Heidelberg (May 2019)
6. Badrinarayanan, S., Fernando, R., Jain, A., Khurana, D., Sahai, A.: Statistical zap arguments
7. Barak, B.: How to go beyond the black-box simulation barrier. In: 42nd FOCS. pp. 106–115. IEEE Computer Society Press (Oct 2001)
8. Barak, B., Ong, S.J., Vadhan, S.P.: Derandomization in cryptography. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 299–315. Springer, Heidelberg (Aug 2003)
9. Belenkiy, M., Chase, M., Kohlweiss, M., Lysyanskaya, A.: P-signatures and noninteractive anonymous credentials. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 356–374. Springer, Heidelberg (Mar 2008)
10. Ben Hamouda, F., Blazy, O., Chevalier, C., Pointcheval, D., Vergnaud, D.: Efficient UC-secure authenticated key-exchange for algebraic languages. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 272–291. Springer, Heidelberg (Feb / Mar 2013)
11. Benhamouda, F., Couteau, G., Pointcheval, D., Wee, H.: Implicit zero-knowledge arguments and applications to the malicious setting. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 107–129. Springer, Heidelberg (Aug 2015)
12. Blazy, O., Fuchsbauer, G., Izabachène, M., Jambert, A., Sibert, H., Vergnaud, D.: Batch Groth-Sahai. In: Zhou, J., Yung, M. (eds.) ACNS 10. LNCS, vol. 6123, pp. 218–235. Springer, Heidelberg (Jun 2010)
13. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications (extended abstract). In: 20th ACM STOC. pp. 103–112. ACM Press (May 1988)
14. Camenisch, J., Kohlweiss, M., Soriente, C.: An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 481–500. Springer, Heidelberg (Mar 2009)
15. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (Aug 2004)
16. Canetti, R., Chen, Y., Reyzin, L., Rothblum, R.D.: Fiat-Shamir and correlation intractability from strong KDM-secure encryption. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part I. LNCS, vol. 10820, pp. 91–122. Springer, Heidelberg (Apr / May 2018)
17. Chaidos, P., Couteau, G.: Efficient designated-verifier non-interactive zero-knowledge proofs of knowledge. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part III. LNCS, vol. 10822, pp. 193–221. Springer, Heidelberg (Apr / May 2018)
18. Chandran, N., Groth, J., Sahai, A.: Ring signatures of sub-linear size without random oracles. In: Arge, L., Cachin, C., Jurdzinski, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 423–434. Springer, Heidelberg (Jul 2007)

19. Chase, M., Meiklejohn, S., Zaverucha, G.: Algebraic MACs and keyed-verification anonymous credentials. In: Ahn, G.J., Yung, M., Li, N. (eds.) ACM CCS 14. pp. 1205–1216. ACM Press (Nov 2014)
20. Couteau, G., Hartmann, D.: Shorter non-interactive zero-knowledge arguments and zaps for algebraic languages. IACR Cryptol. ePrint Arch. 2020, 286 (2020)
21. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y. (ed.) CRYPTO'94. LNCS, vol. 839, pp. 174–187. Springer, Heidelberg (Aug 1994)
22. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (Apr / May 2002)
23. Damgård, I.: Payment systems and credential mechanisms with provable security against abuse by individuals. In: Goldwasser, S. (ed.) CRYPTO'88. LNCS, vol. 403, pp. 328–335. Springer, Heidelberg (Aug 1990)
24. Damgård, I., Fazio, N., Nicolosi, A.: Non-interactive zero-knowledge from homomorphic encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 41–59. Springer, Heidelberg (Mar 2006)
25. Daza, V., González, A., Pindado, Z., Ràfols, C., Silva, J.: Shorter quadratic QA-NIZK proofs. In: PKC 2019, Part I. pp. 314–343. LNCS, Springer, Heidelberg (2019)
26. Dwork, C., Naor, M.: Zaps and their applications. In: 41st FOCS. pp. 283–293. IEEE Computer Society Press (Nov 2000)
27. Escala, A., Groth, J.: Fine-tuning Groth-Sahai proofs. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 630–649. Springer, Heidelberg (Mar 2014)
28. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.L.: An algebraic framework for Diffie-Hellman assumptions. Journal of Cryptology 30(1), 242–288 (Jan 2017)
29. Feige, U., Lapidot, D., Shamir, A.: Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In: 31st FOCS. pp. 308–317. IEEE Computer Society Press (Oct 1990)
30. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO'86. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (Aug 1987)
31. Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 33–62. Springer, Heidelberg (Aug 2018)
32. Garman, C., Green, M., Miers, I.: Decentralized anonymous credentials. In: NDSS 2014. The Internet Society (Feb 2014)
33. Gay, R., Hofheinz, D., Kiltz, E., Wee, H.: Tightly CCA-secure encryption without pairings. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 1–27. Springer, Heidelberg (May 2016)
34. Gay, R., Hofheinz, D., Kohl, L., Pan, J.: More efficient (almost) tightly secure structure-preserving signatures. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 230–258. Springer, Heidelberg (Apr / May 2018)
35. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct NIZKs without PCPs. Cryptology ePrint Archive, Report 2012/215 (2012), `http://eprint.iacr.org/2012/215`
36. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC. pp. 99–108. ACM Press (Jun 2011)

37. Goldwasser, S., Kalai, Y.T.: On the (in)security of the Fiat-Shamir paradigm. In: 44th FOCS. pp. 102–115. IEEE Computer Society Press (Oct 2003)
38. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM Journal on Computing 18(1), 186–208 (1989)
39. González, A.: Shorter ring signatures from standard assumptions. In: PKC 2019, Part I. pp. 99–126. LNCS, Springer, Heidelberg (2019)
40. Groth, J.: Simulation-sound NIZK proofs for a practical language and constant size group signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (Dec 2006)
41. Groth, J., Lu, S.: A non-interactive shuffle with pairing based verifiability. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 51–67. Springer, Heidelberg (Dec 2007)
42. Groth, J., Ostrovsky, R., Sahai, A.: Non-interactive zaps and new techniques for NIZK. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 97–111. Springer, Heidelberg (Aug 2006)
43. Groth, J., Ostrovsky, R., Sahai, A.: New techniques for noninteractive zero-knowledge. Journal of the ACM (JACM) 59(3), 11 (2012)
44. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (Apr 2008)
45. Hamouda, F.B., Blazy, O., Chevalier, C., Pointcheval, D., Vergnaud, D.: Efficient UC-Secure authenticated key-exchange for algebraic languages. Cryptology ePrint Archive, Report 2012/284 (2012), `http://eprint.iacr.org/2012/284`
46. Hanser, C., Slamanig, D.: Structure-preserving signatures on equivalence classes and their application to anonymous credentials. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part I. LNCS, vol. 8873, pp. 491–511. Springer, Heidelberg (Dec 2014)
47. Hofheinz, D., Jager, T.: Tightly secure signatures and public-key encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 590–607. Springer, Heidelberg (Aug 2012)
48. Hofheinz, D., Kiltz, E.: Secure hybrid encryption from weakened key encapsulation. Cryptology ePrint Archive, Report 2007/288 (2007), `http://eprint.iacr.org/2007/288`
49. Jain, A., Jin, Z.: Statistical zap arguments from quasi-polynomial lwe. Tech. rep., Cryptology ePrint Archive, Report 2019/839, 2019. https://eprint. iacr. org ...
50. Jutla, C.S., Ohkubo, M., Roy, A.: Improved (almost) tightly-secure structure-preserving signatures. In: Abdalla, M., Dahab, R. (eds.) PKC 2018, Part II. LNCS, vol. 10770, pp. 123–152. Springer, Heidelberg (Mar 2018)
51. Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 1–20. Springer, Heidelberg (Dec 2013)
52. Jutla, C.S., Roy, A.: Improved structure preserving signatures under standard bilinear assumptions. In: Fehr, S. (ed.) PKC 2017, Part II. LNCS, vol. 10175, pp. 183–209. Springer, Heidelberg (Mar 2017)
53. Kalai, Y., Paneth, O., Yang, L.: On publicly verifiable delegation from standard assumptions. Cryptology ePrint Archive, Report 2018/776 (2018), `https://eprint.iacr.org/2018/776`
54. Kalai, Y.T., Khurana, D., Sahai, A.: Statistical witness indistinguishability (and more) in two messages. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part III. LNCS, vol. 10822, pp. 34–65. Springer, Heidelberg (Apr / May 2018)

55. Kalai, Y.T., Rothblum, G.N., Rothblum, R.D.: From obfuscation to the security of Fiat-Shamir for proofs. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 224–251. Springer, Heidelberg (Aug 2017)
56. Kiltz, E., Pan, J., Wee, H.: Structure-preserving signatures from standard assumptions, revisited. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 275–295. Springer, Heidelberg (Aug 2015)
57. Kiltz, E., Wee, H.: Quasi-adaptive NIZK for linear subspaces revisited. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 101–128. Springer, Heidelberg (Apr 2015)
58. Libert, B., Peters, T., Joye, M., Yung, M.: Non-malleability from malleability: Simulation-sound quasi-adaptive NIZK proofs and CCA2-secure encryption from homomorphic signatures. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 514–532. Springer, Heidelberg (May 2014)
59. Libert, B., Peters, T., Joye, M., Yung, M.: Compactly hiding linear spans - tightly secure constant-size simulation-sound QA-NIZK proofs and applications. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part I. LNCS, vol. 9452, pp. 681–707. Springer, Heidelberg (Nov / Dec 2015)
60. Libert, B., Peters, T., Yung, M.: Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 296–316. Springer, Heidelberg (Aug 2015)
61. Lysyanskaya, A., Rivest, R.L., Sahai, A., Wolf, S.: Pseudonym systems. In: Heys, H.M., Adams, C.M. (eds.) SAC 1999. LNCS, vol. 1758, pp. 184–199. Springer, Heidelberg (Aug 1999)
62. Maurer, U.M.: Unifying zero-knowledge proofs of knowledge. In: Preneel, B. (ed.) AFRICACRYPT 09. LNCS, vol. 5580, pp. 272–286. Springer, Heidelberg (Jun 2009)
63. Morillo, P., Ràfols, C., Villar, J.L.: The kernel matrix Diffie-Hellman assumption. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 729–758. Springer, Heidelberg (Dec 2016)
64. Naor, M.: On cryptographic assumptions and challenges (invited talk). In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 96–109. Springer, Heidelberg (Aug 2003)
65. Noether, S.: Ring signature confidential transactions for monero. Cryptology ePrint Archive, Report 2015/1098 (2015), http://eprint.iacr.org/2015/1098
66. Ràfols, C.: Stretching groth-sahai: NIZK proofs of partial satisfiability. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part II. LNCS, vol. 9015, pp. 247–276. Springer, Heidelberg (Mar 2015)
67. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (Dec 2001)
68. Schnorr, C.P.: Efficient identification and signatures for smart cards (abstract) (rump session). In: Quisquater, J.J., Vandewalle, J. (eds.) EUROCRYPT'89. LNCS, vol. 434, pp. 688–689. Springer, Heidelberg (Apr 1990)
69. Shamir, A.: How to share a secret. Communications of the Association for Computing Machinery 22(11), 612–613 (Nov 1979)
70. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT'97. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (May 1997)
71. Tsang, P.P., Wei, V.K.: Short linkable ring signatures for e-voting, e-cash and attestation. In: International Conference on Information Security Practice and Experience. pp. 48–60. Springer (2005)