Set (Non-)Membership NIZKs from Determinantal Accumulators

14:31, Wednesday 2nd November, 2022

Helger Lipmaa and Roberto Parisella

Simula UiB, Bergen, Norway

Abstract. We construct the most efficient (in the argument size and the verifier's computation) known falsifiable set (non-)membership NIZK $\mathbf{\Pi}^*$, where the membership (resp., non-membership) argument consists of only 9 (resp., 15) group elements. It also has a universal CRS. $\mathbf{\Pi}^*$ is based on the novel concept of determinantal accumulators. Determinantal primitives have a similar relation to recent pairing-based (non-succinct) NIZKs of Couteau and Hartmann (Crypto 2020) and Couteau et al. (CLPØ, Asiacrypt 2021) that structure-preserving primitives have to the NIZKs of Groth and Sahai. $\mathbf{\Pi}^*$ is considerably more efficient than known falsifiable based set (non-)membership NIZKs. We also extend CLPØ by proposing efficient (non-succinct) set *non*-membership arguments for a large class of languages.

Keywords: Commit-and-prove · non-interactive zero-knowledge · set (non-)membership argument · universal accumulator

1 Introduction

In a set (non-)membership NIZK, the prover aims to convince the verifier that an encrypted element χ belongs (does not belong) to a public set $\mathcal{S}$. Fully succinct (constant size and constant-time verifiable) set (non-)membership NIZKs have many applications. Classical applications include anonymous credentials (one has to prove that one has a valid credit card), governmental whitelist (to prevent money laundering), and e-voting (one has to prove that one is an eligible voter). A non-membership NIZK can be used to prove that a key is *not* blacklisted. Set membership NIZKs are instrumental in ring signatures. Recently, set (non-)membership NIZKs have gained popularity in cryptocurrencies. For example, in Zcash, to validate a transaction that intends to spend a coin χ requires one to check that χ is in the set UTXO (unspent transaction outputs).

When χ is public, one can use an efficient (universal) accumulator [BdM93] for this task. A universal accumulator can be reframed as a set (non-)membership *non-zk* non-interactive argument system. Accumulator's completeness and collision-resistance (see Section 2) correspond directly to the completeness and soundness of the set (non-)membership argument system but with public input. To construct a set (non-)membership NIZK, one only needs to add a zero-knowledge (ZK) compiler to the accumulator. Unfortunately, the ZK compiler is

quite complicated in existing constructions, resulting in set (non-)membership NIZKs that are either not falsifiable or not sufficiently efficient.

Related Work. Many set membership NIZKs use either signature schemes or accumulators. In a signature-based set membership NIZKs, the CRS includes signatures of all set elements. The prover proves it knows an (encrypted) signature on the (encrypted) χ. Such NIZKs have several undesirable properties. First, their CRS is non-universal[1] (i.e., it depends on the set). A universal CRS is important in practice since it allows one to rely on a single CRS to construct set (non-)membership NIZKs for different sets. Second, assuming that $|\mathcal{S}|$ is polynomial (and the complement of $\mathcal{S}$ has exponential size), it seems to disallow the construction of set non-membership arguments explicitly.

We will concentrate on accumulator-based constructions since they do not have these two problems. Recall briefly that a (CRS-model) universal accumulator enables one, given a CRS $\mathtt{crs}$, to construct a succinct (non-hiding) commitment $\mathsf{C}_{\mathcal{S}}$ of the set $\mathcal{S}$, such that one can efficiently verify whether $\chi \in \mathcal{S}$, given $\mathtt{crs}$, $\mathsf{C}_{\mathcal{S}}$, χ, and a succinct accumulator argument ψ of (non-)membership.

In a typical accumulator-based set membership NIZK, the CRS contains *set-independent* elements that are sufficient to compute the accumulator arguments of (non-)membership. (This depends on the underlying accumulator, but importantly, the efficient Nguyen accumulator [Ngu05] allows for that.) Hence, their CRS is universal. Moreover, since there is no need to add all accumulator arguments to the CRS, one can at least hope to construct efficient accumulator-based set *non-membership* NIZKs.

Next, we will summarize the published falsifiable set-membership NIZKs.[2] In all cases $\mathcal{S} \subset \mathbb{Z}_p$ and hence $\chi \in \mathbb{Z}_p$. Since the previous papers have not written down all efficiency numbers, our efficiency comparison (see Table 1) is not completely precise.

Belenkiy et al. (BCKL, [BCKL08]) construct a set-membership NIZK by first building a P-signature scheme [BCKL08]. They prove that a commitment opens to an element for which the prover knows a signature, using a Groth-Sahai NIZK [GS08]. Daza et al. (DGPRS-GS, [DGP+19]) use the more efficient weak Boneh-Boyen (WBB) signature scheme instead of the P-signature scheme. Since the WBB signature scheme is not F-unforgeable [BCKL08], Daza et al. modify it slightly. However, using signature schemes means that the CRS of BCKL and DGPRS-GS is non-universal. In addition, Daza et al. [DGP+19] propose a succinct set membership QA-NIZK. However, their verifier's computation is $O(|\mathcal{S}|)$; thus, it is not suitable in our applications.

Acar and Nguyen (AN, [AN11]) replace the signature scheme with the Nguyen accumulator [Ngu05] and then use Groth-Sahai to prove that the prover knows

[1] We follow the previous literature by using "universal " in the definition of universal accumulators (that have a non-membership argument) and universal CRS (that does not depend on the language).

[2] There are many non-falsifiable or random-oracle-based NIZKs (see, e.g., [CCs08,BCF+21]); we do not compete with them, and thus we omit any discussion.

Table 1. Comparison of known fully succinct falsifiable set (non-)membership arguments for univariate sets of size $|\mathcal{S}| \leq N$. Here, $\mathfrak{g}_\iota$ denotes the bit-length of an element of $\mathbb{G}_\iota$, $\mathfrak{m}_\iota$ denotes the cost of a scalar multiplication in $\mathbb{G}_\iota$, $\mathfrak{m}$ denotes the cost of a scalar multiplication in either $\mathbb{G}_1$ or $\mathbb{G}_2$, and $\mathfrak{p}$ denotes the costs of a pairing. The numbers with $*$ are based on our estimation when the original paper did not give enough data. We give online prover's computation, i.e., assuming precomputation.

Paper	Belenkiy et al. [BCKL08]	Acar-Nguyen [AN11]	Daza et al. [DGP+19]	This work (Fig. 9)
Building blocks				
Primitive	P-signature	Nguyen acc.	WBB signature	determinantal acc.
NIZK	Groth-Sahai	Groth-Sahai	Groth-Sahai	CLPØ
Structural properties				
Universal CRS?	✗	✓	✗	✓
Updatable CRS?	✗	✗	✗	✓
Non-membership?	✗	✓	✗	✓
Membership argument efficiency				
$\lvert\mathtt{crs}\rvert$	$(2N+1)\mathfrak{g}_1 + (N+1)\mathfrak{g}_2$	$(N+5)\mathfrak{g}_1 + 4\mathfrak{g}_2{}^*$	$5\mathfrak{g}_1 + (N+5)\mathfrak{g}_2{}^*$	$(N+1)\mathfrak{g}_1 + 4\mathfrak{g}_2$
$\lvert\pi\rvert$	$18\mathfrak{g}_1 + 16\mathfrak{g}_2$	$8\mathfrak{g}_1 + 10\mathfrak{g}_2{}^*$	$10\mathfrak{g}_1 + 8\mathfrak{g}_2{}^*$	$6\mathfrak{g}_1 + 3\mathfrak{g}_2$
P computation	$34\mathfrak{m}$	$16\mathfrak{m}_1 + 16\mathfrak{m}_2{}^*$	$17\mathfrak{m}_1 + 18\mathfrak{m}_2{}^*$	$8\mathfrak{m}_1 + 6\mathfrak{m}_2$
V computation	$68\mathfrak{p}$	$30\mathfrak{p}^*$	$30\mathfrak{p}^*$	$13\mathfrak{p}$
Non-membership argument efficiency				
$\lvert\mathtt{crs}\rvert$	✗	$(N+5)\mathfrak{g}_1 + 4\mathfrak{g}_2{}^*$	✗	$(N+1)\mathfrak{g}_1 + 4\mathfrak{g}_2$
$\lvert\pi\rvert$	✗	$11\mathfrak{g}_1 + 16\mathfrak{g}_2{}^*$	✗	$10\mathfrak{g}_1 + 5\mathfrak{g}_2$
P computation	✗	$26\mathfrak{m}_1 + 28\mathfrak{m}_2{}^*$	✗	$14\mathfrak{m}_1 + 10\mathfrak{m}_2$
V computation	✗	$46\mathfrak{p}^*$	✗	$20\mathfrak{p}$

an accumulator argument. Due to the use of an accumulator, the AN NIZK has a universal CRS; they also propose a set non-membership argument.

BCKL, AN, and DGPRS-GS, and all rely on new (though falsifiable) security assumptions. The central intuition here is that the underlying signature schemes and accumulators are proven to be only secure when the adversary returns χ as an integer. In these NIZKs, χ is essentially encrypted, and the soundness reduction can only recover a group version (say[3], $[\chi]_1$) of χ. The new assumptions (that differ from work to work, see Table 1) guarantee that the adversary cannot break the underlying primitives even if it is allowed only to output $[\chi]_1$.

Structural properties. Another drawback of the signature-based solutions is that it is unclear how to define a universal argument that efficiently allows for non-membership proofs. From the above solutions, only [AN11] (that does not rely on signatures) proposes a set non-membership NIZK.

Efficiency. According to [BCKL08], BCKL's prover performs 34 multi-scalar-multiplications ([BCKL08] does not give separately the number of scalar-multiplications in $\mathbb{G}_1$ and $\mathbb{G}_2$) and the verifer 68 pairings. Neither AN [AN11] Daza et al. [DGP+19] give any efficiency numbers. Hence, the corresponding entries (marked with an asterisk) in Table 1 are based on our estimations.

[3] We use the standard additive bracket notation for pairing-based setting.

Recent NIZKs of Couteau et al. Most of the prior falsifiable set membership NIZKs are based on the Groth-Sahai NIZK [GS08]. Recently, Couteau and Hartmann (CH, [CH20]) proposed a methodology to transform a specific class of Σ-protocols to NIZKs. Intuitively, starting with a Σ-protocol with transcript (a, e, z), CH puts $[\mathsf{e}]_2$ to the CRS and then modifies the computation of z and the verifier's algorithm to work on $[\mathsf{e}]_2$ instead of e. The resulting NIZKs have a CRS consisting of a single group element.

Couteau et al. (CLPØ [CLPØ21]) significantly extended the CH methodology. They constructed efficient commit-and-prove NIZKs for many languages, including (Boolean and arithmetic) Circuit-SAT. Importantly, [CLPØ21] constructed efficient NIZKs for languages that can be described by small algebraic branching programs. The CLPØ NIZK is secure under a new assumption CED (*Computational Extended Determinant*). Depending on the parameters, CED can be either falsifiable or non-falsifiable. For many natural problems like Boolean Circuit-SAT and set membership for poly-sized sets, CED is falsifiable.

Both [CH20,CLPØ21] compare their work to the Groth-Sahai NIZK, showing that in several important use cases, their (falsifiable) NIZKs are more efficient than the Groth-Sahai NIZK. In particular, an important difference between Groth-Sahai and CH/CLPØ is that in the latter, all secret values are only encrypted in $\mathbb{G}_1$. Because of this, the encrypted witness is often three times shorter in CLPØ than in Groth-Sahai; see [CH20,CLPØ21] for examples.

Our first main question is whether one can construct CLPØ-based set (non-)membership NIZKs that are more efficient than the known falsifiable NIZKs [BCKL08,AN11,DGP+19]. Moreover, Groth-Sahai-based NIZKs use specialized primitives (structure-preserving signatures [AFG+16]) that are designed to allow for efficient Groth-Sahai NIZKs. Our second main question is whether one can define a similar class of primitives that allow for efficient CLPØ NIZKs.

1.1 Our Contributions

Summary. Recall that a universal accumulator is a non-zk (non-)membership non-interactive argument system. Thus, one can construct efficient set (non-)membership NIZKs by creating an efficient universal accumulator and then using an efficient ZK compiler to build a NIZK. Our approach is to make the latter part (ZK compiler) as efficient as possible without sacrificing the former part (accumulator) too much.

Differently from the previous work, we will ZK-compile the accumulator to a CLPØ NIZK. We define a *determinantal accumulator* as a universal accumulator with a structure that supports efficient ZK compilation to CLPØ. Determinantal accumulators are related to but different from structure-preserving signatures [AFG+16] that support efficient Groth-Sahai NIZKs. After that, we construct AC^*, an updatable determinantal accumulator with efficient (non-)membership arguments. For this, we follow CLPØ's technique of using algebraic branching programs. Based on AC^*, we then construct $\mathbf{\Pi}^*$, a commit-and-prove, updatable set (non-)membership NIZK with a universal CRS.

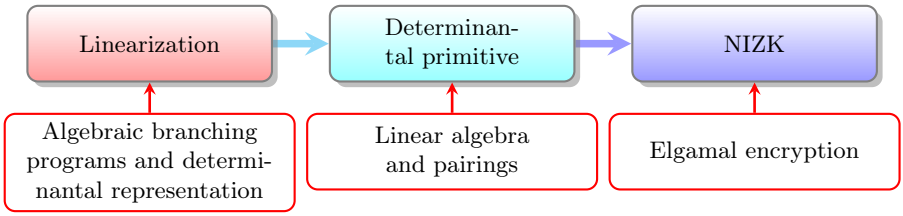


Fig. 1. Our general blueprint for constructing efficient falsifiable NIZKs.

We emphasize that this results in a clear, modular framework for constructing efficient falsifiable NIZKs: first, construct an efficient algebraic branching program for the task at hand. Second, construct a determinantal accumulator (or, in general, a non-zk non-interactive argument system). Third, use the efficient CLPØ-inspired ZK compiler to achieve zero knowledge. See Fig. 1 for a high-level diagram of the new approach.

Moreover, we develop a general efficient technique that allows one to construct non-membership NIZKs for a large class of languages where CLPØ only supported membership NIZKs. We use this technique in the case of AC^* and $\mathbf{\Pi}^*$, but it potentially has many more applications.

The pairing-based setting is ubiquitous in contemporary public-key cryptography. Any advancement in concrete efficiency in simple problems like set-membership proofs is challenging to come by. Our work demonstrates that in this case, the CH/CLPØ framework gives concretely better results than the seminal Groth-Sahai framework.

Determinantal Accumulators. We assume the standard pairing-based setting (see Section 2). We follow [CLPØ21], but we reinterpret their constructions. First, the verifier has access to input (namely, χ), auxiliary (for example, commitment to $\mathcal{S}$), and output (the accumulator's argument) only in $\mathbb{G}_1$, that is, not as integers. The availability of all private values in $\mathbb{G}_1$ enables us to use an efficient ZK compiler, where only elements of $\mathbb{G}_1$ will be encrypted. (In many pairing-based settings, elements of $\mathbb{G}_2$ are twice longer.) On the other hand, they are not available as integers since the ZK compiler encrypts these values by using Elgamal, and the decryption only returns group elements and not integers.

Second, a determinantal accumulator's verifier checks that the determinants (a potentially high-degree polynomial) of some fixed matrices, whose entries are affine maps, are zero. (On the other hand, in prior falsifiable pairing-based accumulators, the verification equations were pairing-product equations.) This can be seen as a linearization of a polynomial $F(\boldsymbol{X})$ by using affine maps. More precisely, the determinantal accumulator's verifier accepts iff $\det \boldsymbol{C}_i(\boldsymbol{\chi}) = 0$ for DRs $\boldsymbol{C}_i(\boldsymbol{X})$ of some well-chosen polynomials $F_i(\boldsymbol{X})$. Here, a DR (determinantal representation) $\boldsymbol{C}(\boldsymbol{X})$ of $F(\boldsymbol{X})$ is a matrix, where each entry of $\boldsymbol{C}(\boldsymbol{X})$ is an affine map of $\boldsymbol{X}$, and the determinant of $\boldsymbol{C}(\boldsymbol{X})$ is $F(\boldsymbol{X})$.

Since we only need to test that the determinant is zero, we follow the underlying ideas of [CH20,CLPØ21] to make the accumulator efficiently and publicly verifiable. Namely, we use the undergraduate linear algebraic fact that

$\det \boldsymbol{C}(\boldsymbol{X}) = 0$ iff there exists a non-zero vector $\boldsymbol{d}$, such that $\boldsymbol{C}(\boldsymbol{X}) \cdot \boldsymbol{d} = \boldsymbol{0}$. To simplify the construction of accumulators and NIZKs, we follow [CLPØ21] and require that the first coordinate of $\boldsymbol{d}$ is non-zero. Moreover, to achieve both soundness and zero-knowledge in the case of NIZKs, we define $\boldsymbol{d} = \left(\begin{smallmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{smallmatrix}\right)$ for a new trapdoor $\mathsf{e} \leftarrow_{\$} \mathbb{Z}_p$. (For such e to exist, the matrices $\boldsymbol{C}(\boldsymbol{X})$ need to satisfy an additional requirement, see [CLPØ21].) To achieve zero knowledge, we mask $\boldsymbol{\delta}$ additively with well-chosen randomness. To balance the randomness, we introduce an additional (e-independent) vector $\boldsymbol{\gamma}$ and prove that $\boldsymbol{C}(\boldsymbol{X}) \cdot \left(\begin{smallmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{smallmatrix}\right) = \boldsymbol{\gamma}$.

Hence, in the implementation of a determinantal accumulator, the prover outputs $[\boldsymbol{\chi}]_1$ (this includes $[\chi]_1$, the candidate element for $\chi \in \mathcal{S}$) and hints $[\boldsymbol{\delta}]_2$ and $[\boldsymbol{\gamma}]_1$. The verifier checks that $[\boldsymbol{C}(\boldsymbol{\chi})]_1 \bullet \left[\begin{smallmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 = [\boldsymbol{\gamma}]_1 \bullet [1]_2$. (Here, $\boldsymbol{\chi}$ is the vector of concrete values of the indeterminates $\boldsymbol{X}$.) Assuming $\boldsymbol{C}(\boldsymbol{X})$ is small, the verification is constant time.

The definition of determinantal accumulators is an important independent contribution of the current paper. In particular, it is easy to take another primitive (for example, a signature scheme) and define its determinantal variant. This may result in other efficient CLPØ-style NIZKs, but we leave any such generalizations to future work.

New Determinantal Accumulator AC^*. AC^* uses a DR $\boldsymbol{C}(\boldsymbol{X})$ that is motivated by Ngyuen's accumulator [Ngu05]. Define

$$\boldsymbol{C}_\Sigma(\mathsf{X}, \mathsf{Q}) := \begin{bmatrix} \Sigma - \mathsf{X} & -1 \\ -\mathbf{Z}_{\mathcal{S}}(\Sigma) & \mathsf{Q} \end{bmatrix}_1 \quad \text{and} \quad \boldsymbol{C}_\sigma(\chi, \mathsf{q}) = \begin{bmatrix} \sigma - \chi & -1 \\ -\mathbf{Z}_{\mathcal{S}}(\sigma) & \mathsf{q} \end{bmatrix}_1 .$$

The AC^* verifier accepts a membership argument if $\det \boldsymbol{C}_\sigma(\chi, \mathsf{q}) = 0$ (that is, $(\sigma - \chi)\mathsf{q} = \mathbf{Z}_{\mathcal{S}}(\sigma)$). Here, χ is the statement (a candidate member of $\mathcal{S}$), $[\mathsf{q}]_1$ is given in the membership argument, σ is a CRS trapdoor, and $\mathbf{Z}_{\mathcal{S}}(\Sigma) := \prod_{s \in \mathcal{S}} (\Sigma - s)$ is the vanishing polynomial of $\mathcal{S}$.

In Nguyen's accumulator, given the membership argument $[\mathsf{q}]_2 \in \mathbb{G}_2$, the verifier checks that $[\sigma - \chi]_1 \bullet [\mathsf{q}]_2 \bullet = [\mathbf{Z}_{\mathcal{S}}(\sigma)]_1 \bullet [1]_2$. In all known Groth-Sahai based solutions, to verify that $\det \boldsymbol{C}_\sigma(\chi, \mathsf{q}) = 0$, either the encryption of χ or q has to be given in $\mathbb{G}_2$. In AC^*, however, all elements are given as members of $\mathbb{G}_1$. Using the approach from above, AC^*'s membership argument is equal to $([\mathsf{q}, \boldsymbol{\gamma}]_1, [\delta]_2)$, where $\boldsymbol{\gamma} \in \mathbb{Z}_p^2$ and $\delta \in \mathbb{Z}_p$. (We will define $\boldsymbol{\gamma}$ and δ in Section 5.)

Complications. Unfortunately, the described solution is not yet sufficient. The main reason why not is that the implication $(\Sigma - \chi) \mid (\mathbf{Z}_{\mathcal{S}}(\Sigma) - \mathsf{r}) \implies \mathbf{Z}_{\mathcal{S}}(\chi) = \mathsf{r}$ (where $\mathsf{r} = 0$ in the membership case and $\mathsf{r} = 1/\mathsf{s}$ in the non-membership case) holds only if χ and r are integers, that is, they do not depend on the trapdoor σ. Since the verifier only has access to $[\chi]_1$ (and $[\mathsf{s}]_1$ in the non-membership case) as group elements, there is no guarantee that χ (and s) does not depend on σ.

Previous works [BCKL08,AN11,DGP+19] solve this problem from scratch, each using a new assumption. We approach it systematically. We define a new security property, F-collision-resistancy. An accumulator is collision-resistant if it is hard for an efficient adversary to return a set $\mathcal{S}$, a candidate element χ, and an accumulator argument ψ, such that the verifier accepts χ as a member of $\mathcal{S}$ iff $\chi \notin \mathcal{S}$. An accumulator is F-collision-resistant if the same holds even if

the adversary, instead of χ, outputs $F(\chi)$. (We always have $F(\chi) = [\chi]_1$.) This notion is related to that of F-unforgeable signatures [BCKL08].

We observed that Nguyen's accumulator (and thus the described version of AC^*) is not F-collision-resistant. We solve this issue by introducing another trapdoor τ. The goal of τ is to guarantee that if the verifier accepts, then χ and r do not depend on σ. We also carefully change AC^*'s verification equations. Crucially, we do it without increasing communication complexity. On the other hand, previous work [BCKL08,AN11,DGP+19] introduced a new equation to prove the knowledge relation and thus added new group elements to the argument.

We prove the F-collision-resistance of AC^* under new, essentially tautological, security assumptions DETACM and DETACNM (determinantal accumulator membership/non-membership). We rely on DETACM (resp., DETACNM) to prove that it is intractable to construct fake accepting membership (resp., non-membership) arguments. Crucially, DETACM and DETACNM are falsifiable. We prove the security of DETACM and DETACNM in the AGM. The AGM security proofs are far from trivial and profoundly rely on which elements of AC^*'s argument are or are not multiplied by τ. Note that also the most efficient structure-preserving signatures are proven secure in the generic group model or AGM, the main difference being that the collision-resistance of accumulators is a simpler assumption than the unforgeability of signature schemes.

General Non-Membership CLPØ NIZK. As a result of independent importance, in Section 3, we develop a generic technique for constructing efficient non-membership CLPØ NIZKs. This results, for example, in a very efficient falsifiable NIZKs that the Elgamal-encrypted value χ is non-zero or that two Elgamal-encrypted values are unequal, see Section 3. Both are more efficient than known alternatives [BCV15,BDSS16] based on Groth-Sahai. Such NIZKs have independent applications in, say, anonymous credential systems and privacy-preserving authenticated identification and key exchange protocols [BCV15,BDSS16] and controllable linkability of group signatures, [BDSS16].

New Succinct Set (Non-)Membership NIZK $\mathbf{\Pi}^*$. We are now ready to describe an efficient commit-and-prove NIZK $\mathbf{\Pi}^*$ for showing that an Elgamal-encrypted χ belongs (or does not belong) to the set $\mathcal{S}$. $\mathbf{\Pi}^*$ is just a simple ZK compilation of AC^*. On top of the work done in AC^*, the prover additionally (1) encrypts the data (including the accumulator input χ) one wants to hide, and (2) creates an additional randomizer $[\boldsymbol{z}]_2$ that balances off the randomizers used in such encryptions. The NIZK verifier performs the accumulator verification on the ciphertexts, taking $[\boldsymbol{z}]_2$ into account.

$\mathbf{\Pi}^*$ is computationally sound, assuming that AC^* is F-collision-resistant. Knowing the Elgamal secret key, the reduction decrypts the encrypted data and returns it together with the hint $[\boldsymbol{\delta}]_2$. We emphasize that $\mathbf{\Pi}^*$ is falsifiable. We prove that $\mathbf{\Pi}^*$ is computationally zero-knowledge, assuming that Elgamal is IND-CPA secure (that is, XDH holds).

Efficiency. In Table 1, we provide an efficiency comparison with some previously proposed set (non-)membership NIZKs. In the case of prover's computation, we have taken the standard approach and assumed that the accumulator

argument ($[\mathsf{q}]_1$ in our case) is precomputed. This always makes sense if $\mathcal{S}$ is small (then all accumulator arguments can be precomputed), but it is also common in case $\mathcal{S}$ can be large. For example, in an anonymous credential system, one only needs to compute the accumulator argument for its own credential. Moreover, all signature-based solutions have precomputation built-in since the signatures are in the CRS. We hence assume precomputation in all cases.

Updatability. Notably, AC^* and $\mathbf{\Pi}^*$ have an updatable [GKM+18] CRS. That is, it is possible to update the CRS sequentially so that the soundness relies only on the honesty of at least one of the updaters (or the original CRS creator). This partially eliminates the undesirable need to trust the CRS creator. None of the previous falsifiable set membership NIZKs (see Table 1) is updatable: this is caused by the use of (non-updatable) signature schemes and Groth-Sahai NIZK. See [BLL00,Lip12] for work on "transparent" accumulators that do not need a trusted CRS at all. We leave it as another open problem to construct a transparent, efficient, falsifiable set (non-)membership NIZK.

Note that one can build set-membership arguments more efficiently by using (non-falsifiable) zk-SNARKs, but the most efficient zk-SNARKs are not updatable. On the other hand, $\mathbf{\Pi}^*$'s efficiency is comparable to that of most efficient updatable and universal zk-SNARKs like Vampire [LSZ22]. However, the latter are only known to be secure in the ROM.

We end the paper with some general discussion and generalization.

2 Preliminaries

Algebraic Branching Programs. An algebraic branching program (ABP) over a finite field $\mathbb{F}_p$ is defined by a directed acyclic graph (V, E), two special vertices $s, t \in V$, and a labeling function ϕ. It computes a function $F : \mathbb{F}_p^\nu \to \mathbb{F}_p$. Here, ϕ assigns to each edge in E a fixed affine (possibly, constant) function in input variables, and $F(\boldsymbol{X})$ is the sum over all $s-t$ paths (that is, paths from s to t) of the product of all the values along the path.

Ishai and Kushilevitz [IK00,IK02] related ABPs to matrix determinants. Given an ABP $\textsc{abp} = (V, E, s, t, \phi)$ computing $F : \mathbb{F}_p^\nu \to \mathbb{F}_p$, we can efficiently (and deterministically) compute a function $\mathsf{IK}_F(\boldsymbol{\chi})$ mapping an input $\boldsymbol{\chi} \in \mathbb{F}_p^\nu$ to a matrix from $\mathbb{F}_p^{\ell\times\ell}$, where $\ell = |V| - 1$, such that: (1) $\det \mathsf{IK}_F(\boldsymbol{\chi}) = F(\boldsymbol{\chi})$, (2) each entry of $\mathsf{IK}_F(\boldsymbol{\chi})$ is an affine map in a single variable χ_i, (3) $\mathsf{IK}_F(\boldsymbol{\chi})$ contains only -1's in the upper 1-diagonal (the diagonal above the main diagonal) and 0's above the upper 1-diagonal.

IK_F is obtained by transposing the matrix you get by removing the column corresponding to s and the row corresponding to t in the matrix $\mathsf{adj}(\boldsymbol{X}) - \boldsymbol{I}$. Here, $\mathsf{adj}(\boldsymbol{X})$ is the adjacency matrix for $\textsc{abp}$ with $\mathsf{adj}(\boldsymbol{X})_{ij} = x$ iff $\phi(i \to j) = x$ and $\mathsf{adj}(\boldsymbol{X})_{ij} = 0$ if there is no edge $i \to j$.

For example, assuming $F(X) = X^2 - X$, one can define an ABP with

$$\mathsf{adj}(X) = \begin{pmatrix} 0 & X & 0 \\ 0 & 0 & X-1 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathsf{IK}_F(X) = \begin{pmatrix} X & -1 \\ 0 & X-1 \end{pmatrix} .$$

Cryptography. A bilinear group generator $\mathsf{Pgen}(1^\lambda)$ returns $\mathsf{p} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, [1]_1, [1]_2)$, where $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are three additive cyclic groups of prime order p, $\mathcal{P}_\iota = [1]_\iota$ is a generator of $\mathbb{G}_\iota$ for $\iota \in \{1, 2, T\}$ with $\mathcal{P}_T = [1]_T := \hat{e}([1]_1, [1]_2)$, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerate efficiently computable bilinear pairing. We require the bilinear pairing to be Type-3; that is, we assume that there is no efficient isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. We use the standard implicit additive "bracket" notation, writing $[a]_\iota$ to denote $a\mathcal{P}_\iota = a[1]_\iota$ for $\iota \in \{1, 2, T\}$. We denote $\hat{e}([a]_1, [b]_2)$ by $[a]_1 \bullet [b]_2$. Thus, $[a]_1 \bullet [b]_2 = [ab]_T$. We freely use the bracket notation together with matrix notation; for example, if $\boldsymbol{AB} = \boldsymbol{C}$ then $[\boldsymbol{A}]_1 \bullet [\boldsymbol{B}]_2 = [\boldsymbol{C}]_T$. We also define $[\boldsymbol{A}]_2 \bullet [\boldsymbol{B}]_1 := ([\boldsymbol{B}]_1^\top \bullet [\boldsymbol{A}]_2^\top)^\top = [\boldsymbol{AB}]_T$.

We write $A \approx_c B$ if the distributions A and B are computationally indistinguishable. Let $\ell, \mathsf{k} \in \mathbb{N}$, with $\ell \geq \mathsf{k}$, be small constants. In the case of asymmetric pairings, usually $\mathsf{k} = 1$. Let p be a large prime. A PPT-sampleable distribution $\mathcal{D}_{\ell,\mathsf{k}}$ is a *matrix distribution* if it samples matrices $\boldsymbol{A} \in \mathbb{Z}_p^{\ell \times \mathsf{k}}$ of full rank k. $\mathcal{L}_1$ is the matrix distribution over matrices $\left(\begin{smallmatrix} 1 \\ a \end{smallmatrix}\right)$, where $a \leftarrow_\$ \mathbb{Z}_p$.

The XDH assumption in $\mathbb{G}_\iota$ holds relative to Pgen if for every PPT $\mathcal{A}$,

$$\Pr\left[b' = b \;\middle|\; \begin{array}{c} \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \sigma, \tau, \zeta \leftarrow_\$ \mathbb{Z}_p; b \leftarrow_\$ \{0,1\}; \\ b' \leftarrow \mathcal{A}([1, \sigma, \tau, \sigma\tau + b\zeta]_\iota) \end{array} \right] \approx_c 0 \ .$$

Let $d_1, d_2 \in \mathsf{poly}(\lambda)$. The (d_1, d_2)-PDL *(Power Discrete Logarithm)* assumption holds relative to Pgen, if for any PPT $\mathcal{A}$,

$$\Pr\left[\sigma' = \sigma \;\middle|\; \begin{array}{c} \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \sigma \leftarrow_\$ \mathbb{Z}_p; \\ \sigma' \leftarrow \mathcal{A}(\mathsf{p}; [(\sigma^i)_{i=0}^{d_1}]_1, [(\sigma^i)_{i=0}^{d_2}]_2) \end{array} \right] \approx_c 0 \ .$$

Let $\ell, \mathsf{k} \in \mathbb{N}$, and $\mathcal{D}_\mathsf{k}$ be a matrix distribution. The $\mathcal{D}_\mathsf{k}$-$(\ell-1)$-CED *assumption* [CLPØ21] holds in $\mathbb{G}_\iota$ relative to Pgen, if for all PPT $\mathcal{A}$,

$$\Pr\left[\begin{array}{c} \boldsymbol{\delta} \in \mathbb{Z}_p^{(\ell-1)\times\mathsf{k}} \wedge \boldsymbol{\gamma} \in \mathbb{Z}_p^{\ell\times\mathsf{k}} \wedge \\ \boldsymbol{C} \in \mathbb{Z}_p^{\ell\times\ell} \wedge (\boldsymbol{\gamma} \| \boldsymbol{C}) \left(\begin{smallmatrix} \boldsymbol{D} \\ \boldsymbol{\delta} \end{smallmatrix}\right) = \boldsymbol{0} \wedge \\ \mathrm{rk}(\boldsymbol{C}) = \ell \end{array} \;\middle|\; \begin{array}{c} \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda), [\boldsymbol{D}]_\iota \leftarrow_\$ \mathcal{D}_\mathsf{k}, \\ ([\boldsymbol{\gamma}, \boldsymbol{C}]_{3-\iota}, [\boldsymbol{\delta}]_\iota) \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{D}]_\iota) \end{array} \right] \approx_c 0 \ .$$

CED may or may not be falsifiable, see [CLPØ21] for a discussion.

Following [CH20,CLPØ21], we will be only concerned with the case $\mathsf{k} = 1$ and $\mathcal{D}_\mathsf{k} = \mathcal{L}_1$. Then, $(\boldsymbol{\gamma} \| \boldsymbol{C}) \left(\begin{smallmatrix} \boldsymbol{D} \\ \boldsymbol{\delta} \end{smallmatrix}\right) = \boldsymbol{0}$ iff, after changing the sign of $\boldsymbol{\gamma}$, $\boldsymbol{C} \left(\begin{smallmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{smallmatrix}\right) = \boldsymbol{\gamma}$.

Elgamal encryption. In Elgamal, the public key is $\mathsf{pk} = [1 \| \mathsf{sk}]_1$, and $\mathsf{Enc}_\mathsf{pk}(\chi; \varrho) \leftarrow (\varrho[1]_1 \| \chi[1]_1 + \varrho[\mathsf{sk}]_1)$, where $\varrho \leftarrow_\$ \mathbb{Z}_p$. We also denote the encryption of $[\chi]_1$ by $\mathsf{Enc}_\mathsf{pk}([\chi]_1; \varrho) = (\varrho[1]_1 \| [\chi]_1 + \varrho[\mathsf{sk}]_1)$. To decrypt, one computes $[\chi]_1 = \mathsf{Dec}_\mathsf{sk}([\boldsymbol{c}]_1) \leftarrow -\mathsf{sk}[c_1]_1 + [c_2]_1$; clearly, the result $[\chi]_1$ of the decryption is a group element and not an integer. Note that $\mathsf{pk} = \mathsf{Enc}_\mathsf{pk}(0; 1)$ and $[0 \| \chi]_1 = \mathsf{Enc}_\mathsf{pk}(\chi; 0)$. As always, we denote $\mathsf{Enc}_\mathsf{pk}([\boldsymbol{a}]_1; \boldsymbol{\varrho}) := (\mathsf{Enc}_\mathsf{pk}([a_i]_1; \varrho_i))_i$. Elgamal is IND-CPA secure under the XDH assumption.

Algebraic Group Model. AGM [FKL18] is an idealized model for security proofs. In the AGM, adversaries are restricted to be *algebraic* in the following sense: if $\mathcal{A}$ inputs some group elements and outputs a group element, it can provide an algebraic representation of the latter in terms of the former.

More precisely, let $\mathbb{G}$ be a cyclic group of prime order p. Let $\mathcal{A}_{\mathrm{alg}}$ be a PPT algorithm, run on initial inputs including description p with oracles or other parties and receive further inputs including obliviously sampled group elements (which it cannot sample directly). Let $\boldsymbol{L} \in \mathbb{G}^n$ be the list of all group elements $\mathcal{A}$ has been given so far such that all other inputs it has received do not depend in any way on group elements. $\mathcal{A}$ is *algebraic* if whenever it outputs a group element $G \in \mathbb{G}$ it also outputs a vector $\boldsymbol{a} = (a_i)_{i=1}^n \in \mathbb{Z}_p^n$, such that $G = \sum_{i=1}^n a_i L_i = \langle \boldsymbol{a}, \boldsymbol{L} \rangle$.

2.1 Universal NIZK Arguments

Let $\{\mathcal{D}_{\mathsf{p}}\}_{\mathsf{p}}$ be a family of distributions, s.t. each $\mathtt{lpar} \in \mathcal{D}_{\mathsf{p}}$ defines a language $\mathcal{L}_{\mathtt{lpar}}$. A universal NIZK Π for $\{\mathcal{D}_{\mathsf{p}}\}_{\mathsf{p}}$ consists of six probabilistic algorithms:

Parameter generation $\mathsf{Pgen}(1^\lambda)$**:** generates public parameters p that fix a distribution $\mathcal{D}_{\mathsf{p}}$.

Key generation $\mathsf{Kgen}(\mathsf{p}, N)$**:** generates a CRS $\mathtt{crs}$ and a trapdoor $\mathtt{td}$. Here, N is a public size parameter (an upper bound of the size $\mathcal{S}$ in our case); we assume N is implicitly in the CRS. We omit N if the CRS does not depend on it. For simplicity of notation, we assume that any group parameters are implicitly included in the CRS. We often denote the sequence "$\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda)$; $(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{Kgen}(\mathsf{p}, N)$" by $(\mathsf{p}, \mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{Kgen}(1^\lambda, N)$.

Computation commitment $\mathsf{Com}(\mathtt{crs}, \mathtt{lpar})$**:** Given a CRS $\mathtt{crs}$ and a language description $\mathtt{lpar} \in \mathcal{D}_{\mathsf{p}}$, outputs a specialized CRS $\mathtt{crs}_{\mathtt{lpar}}$. We assume that $\mathtt{crs}_{\mathtt{lpar}}$ implicitly contains $\mathtt{lpar}$. Com is a deterministic algorithm that can hence be run by both the prover and the verifier. (This algorithm is also known as CRS specialization algorithm, indexer, or derive.)

Prover $\mathsf{P}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x}, \mathbb{w})$**:** Given a specialized CRS $\mathtt{crs}_{\mathtt{lpar}}$ and a statement $\mathbb{x}$ with witness $\mathbb{w}$, outputs an argument π for $\mathbb{x} \in \mathcal{L}_{\mathtt{lpar}}$.

Verifier $\mathsf{V}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x}, \pi)$**:** Given a specialized CRS $\mathtt{crs}_{\mathtt{lpar}}$, a statement, and an argument, either accepts or rejects the argument.

Simulator $\mathsf{Sim}(\mathtt{crs}_{\mathtt{lpar}}, \mathtt{td}, \mathbb{x})$**:** Given a specialized CRS $\mathtt{crs}_{\mathtt{lpar}}$, a trapdoor $\mathtt{td}$, and a statement $\mathbb{x}$, outputs a simulated argument for $\mathbb{x} \in \mathcal{L}_{\mathtt{lpar}}$.

The CRS does not depend on the language distribution or language parameters. However, Com (applied on public arguments) allows one to derive a specialized CRS such that the verifier's operation is efficient given $\mathtt{crs}_{\mathtt{lpar}}$.

The following properties need to hold for a NIZK argument.

Π for $\{\mathcal{D}_{\mathsf{p}}\}_{\mathsf{p}}$ is *perfectly complete*, if

$$\Pr\left[\mathsf{V}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x}, \pi) = 1 \;\middle|\; \begin{array}{c}(\mathsf{p}, \mathtt{crs}, \mathtt{td}) \leftarrow_{\$} \mathsf{K}_{\mathtt{crs}}(1^\lambda); \mathtt{lpar} \in \mathrm{Supp}(\mathcal{D}_{\mathsf{p}}); \\ \mathtt{crs}_{\mathtt{lpar}} \leftarrow \mathsf{Com}(\mathtt{crs}, \mathtt{lpar}); \\ (\mathbb{x}, \mathbb{w}) \in \mathcal{R}_{\mathtt{lpar}}; \pi \leftarrow_{\$} \mathsf{P}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x}, \mathbb{w})\end{array}\right] = 1 \ .$$

Π for $\{\mathcal{D}_{\mathsf{p}}\}_{\mathsf{p}}$ is *computationally sound*, if for every efficient $\mathcal{A}$,

$$\Pr\left[\begin{array}{c}\mathsf{V}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x}, \pi) = 1 \wedge \\ \mathbb{x} \notin \mathcal{L}_{\mathtt{lpar}}\end{array} \;\middle|\; \begin{array}{c}(\mathsf{p}, \mathtt{crs}, \mathtt{td}) \leftarrow_{\$} \mathsf{K}_{\mathtt{crs}}(1^\lambda); \mathtt{lpar} \in \mathrm{Supp}(\mathcal{D}_{\mathsf{p}}); \\ \mathtt{crs}_{\mathtt{lpar}} \leftarrow \mathsf{Com}(\mathtt{crs}, \mathtt{lpar}); \\ (\mathbb{x}, \pi) \leftarrow \mathcal{A}(\mathtt{crs}, \mathtt{lpar})\end{array}\right] \approx 0 \ .$$

Π for $\{\mathcal{D}_{\mathsf{p}}\}_{\mathsf{p}}$ is *perfectly zero-knowledge*, if for all λ, all $(\mathsf{p}, \mathtt{crs}, \mathtt{td}) \in \mathrm{Supp}(\mathsf{K}_{\mathtt{crs}}(1^\lambda))$, all $\mathtt{lpar} \in \mathrm{Supp}(\mathcal{D}_{\mathsf{p}})$ and all $(\mathbb{x}, \mathbb{w}) \in \mathcal{R}_{\mathtt{lpar}}$, the distributions $\mathsf{P}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x}, \mathbb{w})$ and $\mathsf{Sim}(\mathtt{crs}_{\mathtt{lpar}}, \mathtt{td}, \mathbb{x})$ are identical.

Π is *commit-and-prove* if its input $\mathbb{x}$ is a ciphertext, such that the argument convinces the verifier some statement about $\mathsf{Dec}_{\mathsf{sk}}(\mathbb{x})$, i.e., that $\det_{\mathsf{sk}}(\mathbb{x}) \in \mathcal{L}$ for some language $\mathcal{L}$. Commit-and-prove argument systems are usually modular, i.e., one can share the encrypted inputs between several argument systems that prove different properties of the same input. The CLPØ argument system [CLPØ21] (see Section 2.3) is commit-and-prove.

A sound Π is *updatable* [GKM+18] if one can sequentially update the CRS multiple times so that if at least one of the updaters (or the initial CRS creator) is honest, then Π remains sound. We will not give a formal definition. As shown by Groth et al. [GKM+18], a (pairing-based) Π is updatable in the case its CRS is of shape $([f(\boldsymbol{x}) : f \in \mathcal{T}_1]_1, [f(\boldsymbol{x}) : f \in \mathcal{T}_2]_2)$, where $\boldsymbol{x}$ is a vector of trapdoors over $\mathbb{Z}_p$, and $\mathcal{T}_\iota$ are sets of monomials. For example, $\mathtt{crs} = ([1, \tau, \sigma\tau, \ldots, \sigma^N\tau]_1, [1, \sigma, \tau, \sigma\tau]_2)$. On the other hand, Π is not updatable if either $\mathcal{T}_1$ or $\mathcal{T}_2$ contains a non-monomial.

Set (Non-)Membership NIZK. Let $\mathcal{D}$ be some finite domain; in the current paper, $\mathcal{D} = \mathbb{Z}_p$. Let pk be an Elgamal public key and $\mathcal{S}$ be a set of size $\mathcal{S} \in \mathcal{D}^{\leq N}$ for fixed $N = \mathsf{poly}(\lambda)$. Let $\mathtt{lpar} = (\mathsf{pk}, \mathcal{S})$. In the case of NIZKs for set membership and non-membership, we are interested in the following complementary (commit-and-prove) languages:

$$\mathcal{L}^{\mathsf{sm}}_{\mathtt{lpar}} = \left\{ [\mathbf{ct}_\chi]_1 \,\middle|\, \exists \chi, \varrho_\chi \text{ such that } \mathsf{Enc}_{\mathsf{pk}}([\boldsymbol{\chi}]_1; \varrho_\chi) = [\mathbf{ct}_\chi]_1 \wedge \chi \in \mathcal{S} \right\} ,$$
$$\bar{\mathcal{L}}^{\mathsf{sm}}_{\mathtt{lpar}} = \left\{ [\mathbf{ct}_\chi]_1 \,\middle|\, \exists \chi, \varrho_\chi \text{ such that } \mathsf{Enc}_{\mathsf{pk}}([\boldsymbol{\chi}]_1; \varrho_\chi) = [\mathbf{ct}_\chi]_1 \wedge \chi \notin \mathcal{S} \right\} .$$

Instead of defining two NIZKs (for $\mathcal{L}^{\mathsf{sm}}_{\mathtt{lpar}}$ and $\bar{\mathcal{L}}^{\mathsf{sm}}_{\mathtt{lpar}}$), we define a single NIZK where the two arguments share a common CRS. If $\chi \in \mathcal{S}$ (resp., $\chi \notin \mathcal{S}$), then the prover generates a membership (resp., non-membership) argument. The verifier/simulator take an additional argument $mem \in \{\mathsf{Member}, \mathsf{NotMember}\}$. The verifier assumes that its input is a membership argument if $mem = \mathsf{Member}$, and a non-membership argument otherwise. It outputs either Member, $\mathsf{NotMember}$, or Error. We generalize the simulator similarly.

2.2 Accumulators

Benaloh and de Mare defined accumulators in [BdM93]. Universal accumulators [BLL00,BLL02,LLX07] allow non-membership arguments.

We define accumulators in the CRS model only. Hence, within the context of the current paper, universal accumulators are set (non-)membership NIZKs in the case the input χ is public. That is, for $\mathtt{lpar} = \mathcal{S}$, a universal (CRS-model) accumulator is a (non-zk) set (non-)membership non-interactive argument system for the following complementary languages:

$$\mathcal{L}^{\mathsf{acc}}_{\mathtt{lpar}} = \mathcal{S} \ , \quad \bar{\mathcal{L}}^{\mathsf{acc}}_{\mathtt{lpar}} = \mathcal{D} \setminus \mathcal{S} \ .$$

The computation commitment algorithm Com corresponds to the accumulator's commitment algorithm that inputs a set $\mathcal{S}$ and outputs its short commitment. A CRS-model accumulator can have a trapdoor. However, since χ is public (and no zero-knowledge is required) then the trapdoor is not used.

As all argument systems, a universal accumulator must satisfy completeness and soundness properties. Because of the historical reasons, the latter is usually known as *collision-resistance*. Full definitions follow.

A universal accumulator ACC must be *perfectly complete*: for $(\mathtt{crs}, \mathtt{td}) \in \mathsf{Kgen}(1^\lambda)$, $\chi \in \mathcal{D}$, and $\mathcal{S} \in \mathcal{D}^{\leq N}$, $\mathsf{V}(\mathtt{crs}, \mathsf{Com}(\mathtt{crs}, \mathcal{S}), \chi, \mathsf{P}(\mathtt{crs}, \mathcal{S}, \chi))$ outputs Member if $\chi \in \mathcal{S}$ and $\mathsf{NotMember}$ if $\chi \notin \mathcal{S}$.

Definition 1. *Let* ACC *be a universal accumulator.* ACC *is* collision-resistant [BP97] *if for all* $N = \mathsf{poly}(\lambda)$ *and PPT adversaries* $\mathcal{A}$*,*

$$\Pr\left[\begin{array}{c|c} \mathcal{S} \in \mathcal{D}^{\leq N} \wedge \\ \left(\begin{array}{c}(\chi \notin \mathcal{S} \wedge v = \mathsf{Member}) \vee \\ (\chi \in \mathcal{S} \wedge v = \mathsf{NotMember})\end{array}\right) & \begin{array}{c}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \\ (\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{Kgen}(\mathsf{p}, N); \\ (\mathcal{S}, \chi, \psi) \leftarrow \mathcal{A}(\mathtt{crs}); \\ v \leftarrow \mathsf{V}(\mathtt{crs}, \mathsf{Com}(\mathtt{crs}, \mathcal{S}), \chi, \psi)\end{array}\end{array}\right] \approx_c 0 \ .$$

Nguyen [Ngu05] proposed a pairing-based CRS-model accumulator with $\mathcal{D} = \mathbb{Z}_p$. Damgård and Triandopoulos [DT08] and Au et al. [ATSM09] showed independently how to make it universal by adding a non-membership argument.

In Fig. 2, we depict the resulting CRS-model universal accumulator, assuming that $\mathcal{S} \in \mathcal{D}^{\leq N}$. Here, and in what follows, $\mathbf{Z}_{\mathcal{S}}(\Sigma) := \prod_{s \in \mathcal{S}}(\Sigma - s)$ is the vanishing polynomial of $\mathcal{S}$. We slightly simplified its description: Nguyen originally defined $\mathbf{Z}_{\mathcal{S}}(\Sigma) := \prod_{s \in \mathcal{S}}(\Sigma + s)$ (that is, $\mathbf{Z}_{\mathcal{S}}(\Sigma)$ was the vanishing polynomial of $-\mathcal{S} = \{-s : s \in \mathcal{S}\}$), while we define $\mathbf{Z}_{\mathcal{S}}(\Sigma) := \prod_{s \in \mathcal{S}}(\Sigma - s)$; we modified the rest of the formulas in a consistent manner to account for this change. Note that $\mathbf{Z}_{\mathcal{S}}(\chi) = 0$ iff $\chi \in \mathcal{S}$. Intuitively, the prover proves that $\chi \in \mathcal{S}$ by showing that $\mathbf{Z}_{\mathcal{S}}(\chi) = 0$ and $\chi \notin \mathcal{S}$ by showing that $\mathbf{Z}_{\mathcal{S}}(\chi) \neq 0$. A membership argument is shorter since in this case, $\mathbf{Z}_{\mathcal{S}}(\chi) = 0$ and thus the prover does not have to transfer $\mathbf{Z}_{\mathcal{S}}(\chi)$.

Note that $[\mathsf{q}]_1 \leftarrow [(\mathbf{Z}_{\mathcal{S}}(\sigma) - \mathsf{r})/(\sigma - \chi)]_1$ is well defined even if $\sigma = \chi$. In this case, $f(X) = (\mathbf{Z}_{\mathcal{S}}(X) - \mathbf{Z}_{\mathcal{S}}(\chi))/(X - \chi) = \prod_{s \in \mathcal{S}\{\chi\}}(X - s)$ is clearly a polynomial, and thus we can set $[\mathsf{q}]_1 \leftarrow [f(\chi)]_1$.

Com can be seen as a preprocessing algorithm. One can do even more preprocessing in typical accumulators (including Nguyen's). One can precompute accumulator arguments for all $\chi \in \mathcal{S}$ to speed up the online phase of a set membership (but not non-membership) argument. In some applications, one can precompute ψ corresponding to concrete χ. We will always assume this is the case, but, to avoid notational bloat, we will not study preprocessing formally.

2.3 CLPØ NIZK

Since we build on CLPØ [CLPØ21], we will give a lengthier description of their results. Fix $\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda)$ and define $\mathcal{D}_{\mathsf{p}} := \{\mathtt{lpar} = (\mathsf{pk}, F)\}$, where (1) pk is a randomly chosen Elgamal public key for encrypting in $\mathbb{G}_1$, and (2) F is a

$\mathsf{Pgen}(1^\lambda)$**:** the same as the bilinear group generator; returns p.

$\mathsf{Kgen}(\mathsf{p}, N)$**:** $\sigma \leftarrow_\$ \mathbb{Z}_p$; $\mathtt{crs} \leftarrow (\mathsf{p}, [(\sigma^i)_{i=0}^N]_1, [1, \sigma]_2)$;
return $(\mathtt{crs}, \mathtt{td} = \sigma)$;

$\mathsf{Com}(\mathtt{crs}, \mathcal{S})$**:** given $|\mathcal{S}| = N$: output $[\mathsf{C}_\mathcal{S}]_1 \leftarrow [\mathbf{Z}_\mathcal{S}(\sigma)]_1$;

$\mathsf{P}(\mathtt{crs}, \mathcal{S}, \chi)$**:** $\mathsf{r} \leftarrow \mathbf{Z}_\mathcal{S}(\chi)$; $[\mathsf{q}]_1 \leftarrow [(\mathbf{Z}_\mathcal{S}(\sigma) - \mathsf{r})/(\sigma - \chi)]_1$;
If $\chi \in \mathcal{S}$ then $\psi \leftarrow [\mathsf{q}]_1$ else $\psi \leftarrow ([\mathsf{q}]_1, \mathsf{r})$;
return ψ;

$\mathsf{V}(\mathtt{crs}, \mathsf{C}_\mathcal{S}, \chi, \psi)$**:** If ψ parses as $\psi = ([\mathsf{q}]_1, \mathsf{r})$ and $\mathsf{r} = 0$ then return Error;
If ψ parses as $\psi = [\mathsf{q}]_1$ then $\mathsf{r} \leftarrow 0$;
If $[\mathsf{q}]_1 \bullet ([\sigma]_2 - \chi[1]_2) + (\mathsf{r}[1]_1 - [\mathsf{C}_\mathcal{S}]_1) \bullet [1]_2 \neq [0]_T$ then return Error;
If $\mathsf{r} = 0$ then return Member else return $\mathsf{NotMember}$;

Fig. 2. Nguyen's universal accumulator $\mathsf{ACC}_{\mathsf{Nguyen}}$.

polynomial. The simplest version of CLPØ is a set membership NIZK for the set being defined as the set $\mathcal{Z}(F)$ of zeros of the fixed polynomial F.

More precisely, let $\mathcal{S} = \mathcal{Z}(F) := \{x : F(X) = 0\}$ for a polynomial F. Fix $\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda)$. For a fixed Elgamal public key pk, let $\mathtt{lpar} := (\mathsf{pk}, F)$. (Implicitly, $\mathtt{lpar}$ also contains p.) Let $[\mathbf{ct}_\chi]_1 := \mathsf{Enc}_\mathsf{pk}([\boldsymbol{\chi}]_1; \boldsymbol{\varrho}) = (\mathsf{Enc}_\mathsf{pk}([\chi_i]_1; \varrho_i))_i$. Define

$$\mathcal{L}_\mathtt{lpar} = \{[\mathbf{ct}_\chi]_1 : \exists \boldsymbol{\chi} \text{ such that } \mathsf{Dec}_\mathsf{sk}([\mathbf{ct}_\chi]_1) = [\boldsymbol{\chi}]_1 \wedge \boldsymbol{\chi} \in \mathcal{Z}(F)\} \ . \tag{1}$$

Hence, $\mathcal{L}_\mathtt{lpar}$ is a commit-and-prove language. For example, if $F(X) = X^2 - X$, then $\mathcal{L}_{\mathsf{pk},F}$ corresponds to the language of all Elgamal encryptions of Boolean values under the fixed public key pk.

Let $F(\boldsymbol{X}) \in \mathbb{Z}_p[\boldsymbol{X}]$ be a ν-variate polynomial. Let $\ell \geq 1$ be an integer. A matrix $\boldsymbol{C}(\boldsymbol{X}) = (C_{ij}(\boldsymbol{X})) \in \mathbb{Z}_p[\boldsymbol{X}]^{\ell \times \ell}$ is *a quasideterminantal representation (QDR [CLPØ21]) of* F, if the following requirements hold. Here, $\boldsymbol{C}(\boldsymbol{X}) = (\boldsymbol{h}(\boldsymbol{X}) \| \boldsymbol{T}(\boldsymbol{X}))$, where $\boldsymbol{h}(\boldsymbol{X})$ is a column vector.

Affine map: $\boldsymbol{C}(\boldsymbol{X})$ is an affine map. That is, $\boldsymbol{C}(\boldsymbol{X}) = \sum_{k=1}^{\nu} \boldsymbol{P}_k X_k + \boldsymbol{Q}$, where $\boldsymbol{P}_k, \boldsymbol{Q} \in \mathbb{Z}_p^{\ell \times \ell}$ are public matrices.

F-rank: $\det \boldsymbol{C}(\boldsymbol{X}) = F(\boldsymbol{X})$.

First column dependence: For any $\boldsymbol{\chi} \in \mathcal{Z}(F)$, $\boldsymbol{h}(\boldsymbol{\chi}) \in \mathrm{colspace}(\boldsymbol{T}(\boldsymbol{\chi}))$. That is, $\boldsymbol{h}(\boldsymbol{\chi}) = \boldsymbol{T}(\boldsymbol{\chi})\mathbf{w}$ for some $\mathbf{w}$.

The quasideterminantal complexity $\mathsf{qdc}(F)$ of F is the smallest QDR size of F. (Clearly, $\mathsf{qdc}(F) \geq \deg(F)$.) We always assume that the polynomial F in $\mathtt{lpar}$ satisfies $\mathsf{qdc}(F) = \mathsf{poly}(\lambda)$, that is, there exists a $\mathsf{poly}(\lambda)$-size QDR $\boldsymbol{C}(\boldsymbol{X})$ of F. [CLPØ21] showed that such QDRs exist for many F-s.

CLPØ Argument. In Fig. 3, we depict the commit-and-prove updatable universal CLPØ NIZK $\boldsymbol{\Pi}_{\mathsf{clp\emptyset}}$. Intuitively, the verifier checks that $[\begin{smallmatrix}\mathsf{e}\\ \boldsymbol{\delta}\end{smallmatrix}]_2 \bullet [\boldsymbol{C}(\mathbf{ct}_\chi)]_1 = [\boldsymbol{I}_\ell]_2 \bullet [\mathbf{ct}_\gamma]_1 + [\boldsymbol{z}]_2 \bullet \mathsf{pk}$, where $[\boldsymbol{C}(\mathbf{ct}_\chi)]_1 := \sum_{k=1}^{\nu} \boldsymbol{P}_k \cdot [\mathsf{ct}_{\chi k}]_1 + \boldsymbol{Q} \cdot \mathsf{Enc}_\mathsf{pk}(1; 0)$. Couteau et al. [CLPØ21] did not use the terminology of commit-and-prove, universal, and updatable NIZKs. Still, $\boldsymbol{\Pi}_{\mathsf{clp\emptyset}}$ satisfies these properties.

$\mathsf{Pgen}(1^\lambda)$**:** returns the system parameters p, as always.

$\mathsf{Kgen}(\mathsf{p})$**:** $\mathsf{e} \leftarrow_\$ \mathbb{Z}_p$; return $(\mathtt{crs}, \mathtt{td}) \leftarrow ([\mathsf{e}]_2, \mathsf{e})$;

$\mathsf{Com}(\mathtt{crs}, \mathtt{lpar})$**:** return $\mathtt{crs}_{\mathtt{lpar}} \leftarrow (\mathtt{crs}, \mathtt{lpar})$;

$\mathsf{P}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x} = [\mathbf{ct}_\chi]_1, \mathbb{w} = (\boldsymbol{\chi}, \boldsymbol{\varrho}))$**:** Write $\boldsymbol{C}(\boldsymbol{\chi}) = (\boldsymbol{h} \| \boldsymbol{T})(\boldsymbol{\chi})$;
$\boldsymbol{\varrho_\delta} \leftarrow_\$ \mathbb{Z}_p^{\ell-1}$; $\boldsymbol{\gamma} \leftarrow -\boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{\varrho_\delta}$;
Compute $\mathbf{w}$ such that $\boldsymbol{T}(\boldsymbol{\chi})\mathbf{w} = \boldsymbol{h}(\boldsymbol{\chi})$;
$[\boldsymbol{\delta}]_2 \leftarrow -(\mathbf{w}[\mathsf{e}]_2 + \boldsymbol{\varrho_\delta}[1]_2)$;
$\boldsymbol{\varrho}_\gamma \leftarrow_\$ \mathbb{Z}_p^\ell$; $[\mathbf{ct}_\gamma]_1 \leftarrow \mathsf{Enc}_{\mathsf{pk}}([\boldsymbol{\gamma}]_1; \boldsymbol{\varrho}_\gamma) \in \mathbb{G}_1^{\ell \times 2}$;
$[\boldsymbol{z}]_2 \leftarrow (\sum_{k=1}^{\nu} \varrho_k \boldsymbol{P}_k) \left[\begin{smallmatrix}\mathsf{e}\\ \boldsymbol{\delta}\end{smallmatrix}\right]_2 - \boldsymbol{\varrho}_\gamma[1]_2 \in \mathbb{G}_2^\ell$;
Return $\pi \leftarrow ([\mathbf{ct}_\gamma]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2) \in \mathbb{G}_1^{\ell \times 2} \times \mathbb{G}_2^{2\ell-1}$;

$\mathsf{V}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x} = [\mathbf{ct}_\chi]_1, \pi)$**:** check $\sum_{k=1}^{\nu} (\boldsymbol{P}_k \left[\begin{smallmatrix}\mathsf{e}\\ \boldsymbol{\delta}\end{smallmatrix}\right]_2 \bullet [\mathbf{ct}_k]_1) + \boldsymbol{Q} \left[\begin{smallmatrix}\mathsf{e}\\ \boldsymbol{\delta}\end{smallmatrix}\right]_2 \bullet [0 \| 1]_1 = [\boldsymbol{I}_\ell]_2 \bullet [\mathbf{ct}_\gamma]_1 + [\boldsymbol{z}]_2 \bullet \mathsf{pk}$;

$\mathsf{Sim}(\mathtt{crs}_{\mathtt{lpar}}, \mathtt{td}, \mathbb{x} = [\mathbf{ct}_\chi]_1)$**:** $\boldsymbol{\delta} \leftarrow_\$ \mathbb{Z}_p^{\ell-1}$;
$\boldsymbol{z} \leftarrow_\$ \mathbb{Z}_p^\ell$; $[\mathbf{ct}_\gamma]_1 \leftarrow \sum_{k=1}^{\nu} \boldsymbol{P}_k(\left[\begin{smallmatrix}\mathsf{e}\\ \boldsymbol{\delta}\end{smallmatrix}\right])[\mathbf{ct}_k]_1 + \mathsf{Enc}_{\mathsf{pk}}(\boldsymbol{Q}(\left[\begin{smallmatrix}\mathsf{e}\\ \boldsymbol{\delta}\end{smallmatrix}\right])[1]_1; -\boldsymbol{z})$;
Return $\pi \leftarrow ([\mathbf{ct}_\gamma]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2) \in \mathbb{G}_1^{\ell \times 2} \times \mathbb{G}_2^{2\ell-1}$;

Fig. 3. The commit-and-prove CLPØ NIZK $\boldsymbol{\Pi}_{\mathsf{clp\emptyset}}$ for $\mathcal{L}_{\mathsf{pk},F}$.

We will state Fact 1 for the sake of completeness.

Fact 1 ([CLPØ21]) *Let $\{\mathcal{D}_\mathsf{p}\}_\mathsf{p}$ be the family of language distributions, where $\mathcal{D}_\mathsf{p} = \{\mathtt{lpar} = (\mathsf{pk}, F)\}$. Here, $F(\boldsymbol{X})$ is a ν-variate polynomial of degree d, where $\nu, d \in \mathsf{poly}(\lambda)$. Let $\boldsymbol{C}(\boldsymbol{X}) \in \mathbb{Z}_p[\boldsymbol{X}]^{\ell \times \ell}$ be a QDR of F. The NIZK $\boldsymbol{\Pi}_{\mathsf{clp\emptyset}}$ for $\{\mathcal{D}_\mathsf{p}\}_\mathsf{p}$ from Fig. 3 is perfectly complete and perfectly zero-knowledge. It is computationally (adaptive) sound under the $\mathcal{L}_1$-$(\ell-1)$-CED assumption in $\mathbb{G}_2$ relative to* Pgen.

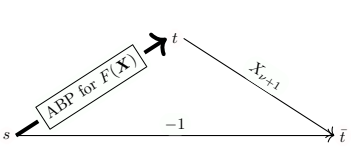
Efficient Instantiation Based on ABP. Couteau et al. [CLPØ21] constructed a QDR $\mathsf{IK}_F(\boldsymbol{X})$ for any polynomial F that can be efficiently computed by an algebraic branching program (ABP).

Fact 2 ([CLPØ21]) *Let $\mathrm{ABP} = (V, E, s, t, \phi)$ be an ABP that computes a ν-variate polynomial $F(\boldsymbol{X})$. Then $\mathsf{IK}_F(\boldsymbol{X})$ is a QDR of F with $\ell = |V| - 1$.*

In particular, $\mathsf{qdc}(F) \le |V| - 1$. This results in NIZKs for $\mathcal{L}_{\mathsf{pk},F}$ whenever F has a small ABP.

3 General Non-Membership NIZK Argument System

For a set $\mathcal{F}$ of polynomials, let $\mathcal{Z}(\mathcal{F})$ be the set of common zeros of all $F_i \in \mathcal{F}$. Next, we construct efficient (commit-and-prove, updatable, universal)



$$\left(\begin{array}{c|c} \mathsf{IK}_F(X_1,\ldots,X_\nu) & \begin{matrix}0\\0\\ \vdots \\ 0\\ -1\end{matrix} \\ \hline -1\ 0\ \ldots\ 0 \qquad 0 & X_{\nu+1} \end{array}\right)$$

Fig. 4. ABP $\overline{\text{ABP}}$ for the $\bar{F}(\boldsymbol{X}, X_{\nu+1}) = F(\boldsymbol{X})X_{\nu+1} - 1$ and the matrix $\mathsf{IK}_{\bar{F}}(\boldsymbol{X}, X_{\nu+1})$.

non-membership NIZKs for $\mathcal{S} = \mathcal{Z}(\mathcal{F})$, given that for each $F_i \in \mathcal{F}$, there exists a small ABP that computes F_i. The modifications are at the level of ABP and thus do not depend on the inner workings of $\mathbf{\Pi}_{\mathsf{clp}\emptyset}$. The current section has independent importance since non-membership NIZKs have their own applications, [ATSM09,BCV15,BDSS16,BBLP21].

New Non-Membership NIZK. Assume $\mathcal{F} = \{F\}$, where $F(\boldsymbol{X}) : \mathbb{F}_p^\nu \mapsto \mathbb{F}_p$ is a polynomial that can be computed by a small ABP $\text{ABP} = (V, E, s, t, \phi)$. We construct a new ABP $\overline{\text{ABP}}$ as follows (see Fig. 4): we add to ABP a new target vertex $\bar{t}$ and two edges, $s \to \bar{t}$ and $t \to \bar{t}$. We naturally extend ϕ to a new labeling function $\bar{\phi}$, such that $\bar{\phi}(s \to \bar{t}) = -1$ and $\bar{\phi}(t \to \bar{t}) = X_{\nu+1}$, where $X_{\nu+1}$ is a new indeterminate. Let $\bar{F}(\boldsymbol{X}, X_{\nu+1}) : \mathbb{F}_p^{\nu+1} \mapsto \mathbb{F}_p$, $\bar{F}(\boldsymbol{X}, X_{\nu+1}) = F(\boldsymbol{X})X_{\nu+1} - 1$, be the polynomial computed by $\overline{\text{ABP}}$. Clearly, if $F(\boldsymbol{\chi}) = 0$ for a concrete input assignment $\boldsymbol{\chi}$, then $\bar{F}(\boldsymbol{\chi}, \chi_{\nu+1}) = -1 \neq 0$ for all values of $\chi_{\nu+1}$. On the other hand, if $F(\boldsymbol{\chi}) \neq 0$, then there exists $\chi_{\nu+1} = F(\boldsymbol{\chi})^{-1}$, such that $\bar{F}(\boldsymbol{\chi}, \chi_{\nu+1}) = 0$.

Thus, to obtain a non-membership NIZK for the algebraic set $\mathcal{S} = \mathcal{Z}(F)$, it suffices to construct a membership NIZK for the algebraic set $\bar{\mathcal{S}} = \mathcal{Z}(\bar{F})$. For this, one can use $\mathbf{\Pi}_{\mathsf{clp}\emptyset}$ from Fig. 4 for the QDR $\mathsf{IK}_{\bar{F}}$. The resulting NIZK is again secure under a CED assumption (see Fact 1). Moreover, if the NIZK for F relies on a falsifiable version of CED, then so does the NIZK for $\bar{F}$.

Examples. To show that $\chi \neq 0$, we can run $\mathbf{\Pi}_{\mathsf{clp}\emptyset}$ with the QDR

$$\bar{\boldsymbol{C}}(\mathsf{X}, \mathsf{S}) := \left(\begin{smallmatrix} \mathsf{X} & -1 \\ -1 & \mathsf{S} \end{smallmatrix}\right) ,$$

where in the honest case, $\mathsf{S} = 1/\mathsf{X}$. One can easily extend it to the proof that two plaintexts χ_1 and χ_2 are unequal, by using the QDR

$$\bar{\boldsymbol{C}}(\mathsf{X}_1, \mathsf{X}_2, \mathsf{S}) := \left(\begin{smallmatrix} \mathsf{X}_1 - \mathsf{X}_2 & -1 \\ -1 & \mathsf{S} \end{smallmatrix}\right) ,$$

where in the honest case, $\mathsf{S} = 1/(\mathsf{X}_1 - \mathsf{X}_2)$.

The argument length of the resulting NIZKs (including encryption of s but not of χ or χ_i) is $6\mathfrak{g}_1 + 3\mathfrak{g}_2$. They are based on a less standard and non-falsifiable assumption (CED instead of SXDH) but are significantly more efficient than Groth-Sahai-based constructions of [BCV15,BDSS16]. In particular, the communication of the NIZK of plaintext inequality of [BCV15] consists of 15 elements of $\mathbb{G}_1$, 4 elements of $\mathbb{G}_2$, and 2 elements of $\mathbb{Z}_p$. (The more efficient construction [BBLP21] works in the random oracle model.)

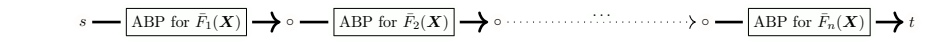


Fig. 5. ABP $\overline{\text{ABP}}$ for $\bar{F}(\boldsymbol{X}) = \prod \bar{F}_i(\boldsymbol{X})$.

Finally, consider the task of proving that an encrypted integer χ is non-Boolean. In this case, one can define the QDR

$$\boldsymbol{C}_{\{0,1\}}(X, \mathsf{S}) := \begin{pmatrix} X & -1 & 0 \\ 0 & X & -1 \\ -1 & 0 & \mathsf{S} \end{pmatrix} .$$

Generalization. Let $\mathcal{F} = \{F_1, \ldots, F_\nu\}$ for $\nu > 1$. To obtain a set non-membership NIZK for $\mathcal{S} = \mathcal{Z}(\mathcal{F})$, we first construct an ABP that computes each $\bar{F}_i$ (see the previous subsubsection). After that, we construct an ABP that computes a polynomial $\bar{F}(\boldsymbol{X})$, such that $\bar{F}(\boldsymbol{\chi}) = 0$ iff $\bar{F}_i(\boldsymbol{\chi}) = 0$ for some i. Define $\bar{F}(\boldsymbol{X}) = \prod \bar{F}_i(\boldsymbol{X})$, and define its ABP as the concatenation of the ABPs for individual polynomials $\bar{F}_i$. See Fig. 5. We then use $\boldsymbol{\Pi}_{\mathsf{clp\emptyset}}$ for the QDR $\mathsf{IK}_{\bar{F}}$ from Fig. 4. The resulting NIZK is secure according to Fact 1.

4 Determinantal Accumulators

It is easy to see that universal accumulator is a *non-zk* set (non-)membership non-interactive argument system (i.e., one that possesses both membership and non-membership arguments). Hence, it is logical to try to construct a set (non-)membership NIZK by first constructing an accumulator and then adding a zero-knowledge layer to obtain privacy.

While the end goal is to define efficient NIZKs, both steps of the descrived blueprint can be expensive per se. In the current paper, we are interested in constructing a CLPØ-style set (non-)membership NIZK where the second step is as simple as possible. To achieve this, we first reinterpret $\boldsymbol{\Pi}_{\mathsf{clp\emptyset}}$. We then use the obtained understanding to define and construct *determinantal accumulators* that allow for a simple zero-knowledge layer. For latter, a determinantal accumulator must have a specific structure consistent with $\boldsymbol{\Pi}_{\mathsf{clp\emptyset}}$'s design.

The relation between determinantal accumulators and CLPØ is similar to the relation between structure-preserving signatures and Groth-Sahai. Hence, we also compare both primitives.

Intuition. Recall that in $\boldsymbol{\Pi}_{\mathsf{clp\emptyset}}$ [CLPØ21], one rewrites the condition $\boldsymbol{\chi} \in \mathcal{S}$ as the condition $F_i(\boldsymbol{\chi}) = 0$ for a set of polynomials $\{F_i\}$.[4] After that, one constructs QDRs $\boldsymbol{C}_i(\boldsymbol{X})$ for each F_i, such that $\det \boldsymbol{C}_i(\boldsymbol{X}) = F_i(\boldsymbol{X})$. This step can be seen as linearization: while F_i can be a high-degree polynomial, each entry of $\boldsymbol{C}_i$ is an affine map. As typical in group-based cryptography, it is easier to solve linearized tasks. After that, [CLPØ21] proposes a technique of constructing QDRs (i.e., linearization algorithm) by using algebraic branching programs.

[4] In our new primitives, the set consists of only one polyomial. However, the framework is valid in the more general case.

Given the QDRs, $\mathbf{\Pi}_{\mathsf{clp\emptyset}}$'s prover P aims to convince the verifier that each $\det \boldsymbol{C}_i(\boldsymbol{\chi})$ is zero. Crucially, the verifier has access only to encrypted $[\boldsymbol{\chi}]_1$ but not to $\boldsymbol{\chi}$ or even $[\boldsymbol{\chi}]_1$. Since each entry of $\boldsymbol{C}_i$ is affine and the cryptosystem is additively homomorphic, the verifier can compute an encryption of $[\boldsymbol{C}_i(\boldsymbol{\chi})]_1$ given an encryption of $[\boldsymbol{\chi}]_1$. Knowing sk, the soundness reduction decrypts ciphertexts, obtains $[\boldsymbol{C}_i(\boldsymbol{\chi})]_1$, and uses it to break CED. To preserve privacy, the verifier cannot $[\boldsymbol{C}_i(\boldsymbol{\chi})]_1$ and thus also not $\det \boldsymbol{C}_i(\boldsymbol{\chi})$.

In a *non-zk* CLPØ-style non-interactive argument system, we proceed as in CLPØ, except that we do not encrypt any of the values. In particular, similarly to the soundness reduction in $\mathbf{\Pi}_{\mathsf{clp\emptyset}}$, the verifier has access to $[\chi]_1$ and thus also to $[\boldsymbol{C}_i(\boldsymbol{\chi})]_1$. To be compatible with CLPØ, the verifier is not however given access to $\det \boldsymbol{C}_i(\boldsymbol{\chi})$ or even $\boldsymbol{\chi}$ as integers. Given this, we must take additional care to ensure that the accumulator will be secure.

4.1 Determinant Verification

The verifier needs to check efficiently that the determinant of a given matrix $\boldsymbol{C}_i(\boldsymbol{\chi})$ is zero. The main problem is that since the verifier sees $[\boldsymbol{C}_i(\boldsymbol{\chi})]_1$ but not $\boldsymbol{C}_i(\boldsymbol{\chi})$, the verifier's task is intractable. Next, we outline a straightforward but non-satisfactory solution to this problem together with three modifications.

First, without any additional hints given to the verifier, we have an accumulator with inefficient verification, where the verifier computes the discrete logarithm of $[\boldsymbol{C}_i(\boldsymbol{\chi})]_1$ to obtain $\boldsymbol{C}_i(\boldsymbol{\chi})$. This might be fine in the NIZK since the NIZK verifier does not have to perform the accumulator verification; instead, the NIZK verifier checks (efficiently) the NIZK argument showing that the accumulator verifier accepts. However, since also the soundness reduction does not get any hints about $\boldsymbol{C}_i(\boldsymbol{\chi})$, it will not be able to verify whether this results in a non-falsifiable NIZK, as explained in [CH20,CLPØ21].

Second, following [ALSZ20], we can allow the prover to output as hints all partial multiplications needed in the Leibniz formula for the determinant. In that case, one can obtain a PPT verifiable accumulator and thus a NIZK based on falsifiable assumptions. However, while PPT, it is concretely very expensive: if the dimension of the matrix is large, the hint is potentially huge [ALSZ20]. [5] Moreover, since in the NIZK, one has to encrypt the matrix elements in both groups, one has to use the less efficient DLIN encryption, see [CLPØ21].

Third, we can use the undergraduate linear-algebraic fact that $\det \boldsymbol{C} = 0$ iff there exists a non-zero vector $\boldsymbol{x}$ such that $\boldsymbol{C}\boldsymbol{x} = \boldsymbol{0}$. We can utilize this fact by outputting $[\boldsymbol{x}]_2$ as a hint to the verifier/soundness reduction. However, $[\boldsymbol{x}]_2$ can reveal secret information and thus must be hidden. We do not want to encrypt $[\boldsymbol{x}]_2$: since $[\boldsymbol{x}]_2$ is given in $\mathbb{G}_2$, this means that one again needs to use DLIN.

Fourth, we rely on CED as follows: recall that CED states that $\det \boldsymbol{C} = 0$ iff one can compute vectors $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ such that $\boldsymbol{C}\left(\begin{smallmatrix}\mathsf{e}\\ \boldsymbol{\delta}\end{smallmatrix}\right) = \boldsymbol{\gamma}$, where $\mathsf{e} \leftarrow\!\!\$\, \mathbb{Z}_p$. (The

[5] In the case of 2×2 matrices, the hint can be $[\boldsymbol{C}_1]_2$, where $\boldsymbol{C}_1$ is the first row of $\boldsymbol{C} \in \mathbb{Z}_p^{2\times 2}$ [ALSZ20]. In the case of a 3×3 matrix $\boldsymbol{C}$, the prover already needs to output six values $[C_{1i}C_{2j}]_2$ for $i \neq j$.

first coordinate of $\boldsymbol{x} = (\begin{smallmatrix}\mathsf{e}\\ \boldsymbol{\delta}\end{smallmatrix})$ is non-zero w.p. $1 - 1/p$ since $\boldsymbol{C}$ is a QDR.) For the security of CED, $\boldsymbol{\gamma}$ must not depend on e. Here, as in [CH20,CLPØ21], $\boldsymbol{\delta}$ is masked by uniformly random addend $\boldsymbol{\varrho}_{\boldsymbol{\delta}}$ and $\boldsymbol{\gamma}$ is needed to balance $\boldsymbol{\varrho}_{\boldsymbol{\delta}}$. Thus, the prover gives $([\boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2)$ as a hint to the verifier/soundness reduction. In the NIZK, $[\boldsymbol{\gamma}]_1$ is encrypted but $[\boldsymbol{\delta}]_2$ (that looks uniformly random after adding $\boldsymbol{\varrho}_{\boldsymbol{\delta}}$) is not. While the resulting accumulator is less efficient than Nguyen's, the new NIZK (see Section 7) is very efficient since it reuses the hints $([\boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2)$.

4.2 Definition

The reasoning from Section 4.1 shows that one can construct an efficient accumulator (and NIZK) even if $\boldsymbol{\chi}$ is only given to the verifier in one source group. This motivates the new definition of determinantal accumulators. For comparison purposes only, we will first define structure-preserving signature schemes [AFG+16].

Definition 2 (Structure-preserving signature scheme [AFG+16]). *A digital signature scheme is* structure preserving *relative to bilinear group generator* Pgen *if (1) the common parameters* p *consist of group description generated by* Pgen*, some constants, and some source group elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*, (2) the verification algorithm* V *consists only of evaluating membership in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *and relations described by paring product equations, (3) verification keys* vk*, messages* χ *and signatures* σ *solely consist of group elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*.*

Our definition of determinantal accumulators is very close in spirit. For clarity, we highlight the differences between "structure preserving" and "determinantal" primitives. Other differences are caused by having an accumulator instead of a signature scheme.

Definition 3 (Determinantal accumulator). *An accumulator is* determinantal *relative to bilinear group generator* Pgen *if*

(a) the common parameters p *consist of group description generated by* Pgen*, some constants, and some source group elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*,*
(b) the verification algorithm V *consists only of evaluating membership in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *and relations described by checking that* $\boldsymbol{C}_i(\chi) = 0$*, where each* $\boldsymbol{C}_i(X)$ *is a QDR,*
(c) the CRS $\mathtt{crs}$*, messages* χ*, commitments* $\mathsf{C}_{\mathcal{S}}$*, and membership arguments* ψ *solely consist of group elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*,*
(d) messages χ *are given to the verifier as elements of* $\mathbb{G}_1$*,*
(e) the set of $\mathbb{G}_2$ *elements in* ψ *is independent of* χ*.*

Items d and e help creating efficient NIZKs, where one only has to encrypt elements of $\mathbb{G}_1$. We assume that all determinantal accumulators use the fourth method from Section 4.1. Since in that case, the only $\mathbb{G}_2$ element in ψ is $\boldsymbol{\delta}$ and the latter is chosen uniformly from $\mathbb{G}_2$ in [CLPØ21], Item e follows automatically.

Clearly, this approach is not restricted to accumulators.

Comparison to Structure-Preserving Primitives (SPPs). Determinantal primitives are quite different from SPPs. First, compared to SPPs, we restrict the

inputs to be from a single source group. While this is a restriction, it potentially boosts efficiency: since all inputs have to be encrypted in one source group, one can use Elgamal instead of less efficient DLIN or Groth-Sahai commitments. Because $\mathbb{G}_2$ elements are often twice longer than $\mathbb{G}_1$ elements, this can make the statement of the NIZK (commitment to χ) three times shorter.

Second, the verifier is not restricted to quadratic equations: the QDRs $\boldsymbol{C}_i$ can be polynomially large. In the new non-membership accumulator, the determinant of the used $\boldsymbol{C}_i$ is a cubic polynomial. This means that some of the known lower-bounds for SPPs (e.g., [AFG+16]) *might* not apply.

Third, and crucially, determinantal accumulators are (efficient) CLPØ-style non-zk non-interactive argument systems. On the other hand, structure-preserving signatures are independent primitives with the property that one can construct (efficient) Groth-Sahai NIZKs for tasks like signature possession. It is not known how to construct structure-preserving accumulators.

5 The New Determinantal Accumulator AC^*

5.1 F-Collision-Resistance

In the new set (non-)membership NIZK, χ is Elgamal-encrypted. In the soundness reduction, the reduction decrypts it to obtain $[\chi]_1$ but does not obtain χ. Because of that, the collision-resistance property must hold against adversaries who return $[\chi]_1$ but not χ. Definition 4 is inspired by the definition of F-unforgeable signature schemes, [BCKL08], where F is an efficiently computable one-way bijection. Since F is a bijection, $\chi \in \mathcal{S}$ iff $F(\chi) \in F(\mathcal{S})$ iff $\exists s \in \mathcal{S}. F(\chi) = F(s)$.

Definition 4. *Let $\mathcal{D}$ be a domain and F be an efficiently computable (one-way) bijection. A universal accumulator* ACC *is* F-collision resistant *if for any* $N = \mathsf{poly}(\lambda)$ *and PPT adversaries* $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{f-cr}}_{\mathsf{Pgen},F,\mathsf{ACC},\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{c} \mathcal{S} \in \mathcal{D}^{\leq N} \wedge \\ \left(\begin{array}{c}(\chi \notin \mathcal{S} \wedge v = \mathsf{Member}) \vee \\ (\chi \in \mathcal{S} \wedge v = \mathsf{NotMember})\end{array}\right)\end{array}\middle|\begin{array}{c}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); (\mathtt{crs}, \sigma) \leftarrow \mathsf{Kgen}(\mathsf{p}, N); \\ (\mathcal{S}, F(\chi), \psi) \leftarrow \mathcal{A}(\mathtt{crs}); \\ v \leftarrow \mathsf{V}(\mathtt{crs}, \mathsf{Com}(\mathtt{crs}, \mathcal{S}), F(\chi), \psi)\end{array}\right] \approx_c 0 \ .$$

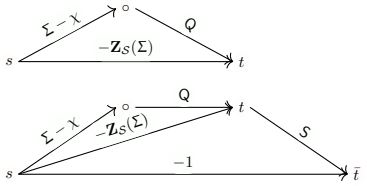
Here, we highlighted the differences with Definition 1.

In what follows, $F = [\cdot]_1$.

5.2 Construction

In Fig. 7, we propose a new F-collision-resistant determinantal (CRS-model, universal) accumulator AC^*. Next, we give the intuition behind its construction.

The first task constructing AC^* is to fix suitable verification equation that defines a polynomial $F(\boldsymbol{X})$, such that the verifier accepts iff $F(\boldsymbol{\chi}) = 0$. Given F, we use an ABP to define a QDR $\boldsymbol{C}(\boldsymbol{X})$ for F.



$$\boldsymbol{C}_\Sigma(\mathsf{X},\mathsf{Q}) = \begin{pmatrix} \Sigma-\mathsf{X} & -1 \\ -\mathbf{Z}_{\mathcal{S}}(\Sigma) & \mathsf{Q} \end{pmatrix}$$

$$\bar{\boldsymbol{C}}_\Sigma(\mathsf{X},\mathsf{Q},\mathsf{S}) = \begin{pmatrix} \Sigma-\mathsf{X} & -1 & 0 \\ -\mathbf{Z}_{\mathcal{S}}(\Sigma) & \mathsf{Q} & -1 \\ -1 & 0 & \mathsf{S} \end{pmatrix}$$

Fig. 6. Above: ABP for $F_\Sigma(\mathsf{X},\mathsf{Q})$ and the corresponding QDR $\boldsymbol{C}_\Sigma(\mathsf{X},\mathsf{Q})$. Below: ABP for $\bar{F}_\Sigma(\mathsf{X},\mathsf{Q},\mathsf{S})$ and the corresponding QDR $\bar{\boldsymbol{C}}_\Sigma(\mathsf{X},\mathsf{Q},\mathsf{S})$.

In the case of the membership argument, we start with the verification equation of $\mathsf{ACC}_{\mathsf{Nguyen}}$ from Fig. 2, which defines the bivariate polynomial

$$F_\Sigma(\mathsf{X},\mathsf{Q}) := (\Sigma - \mathsf{X})\mathsf{Q} - \mathbf{Z}_{\mathcal{S}}(\Sigma) \ .$$

Here, say, Q is the indeterminate corresponding to $\mathsf{q} \in \psi$ (see Fig. 2). Clearly, the membership argument verifier of $\mathsf{ACC}_{\mathsf{Nguyen}}$ accepts iff $[F_\sigma(\chi,\mathsf{q})]_1 = [0]_1$.

In the non-membership argument, we need to prove that $F_\Sigma(\mathsf{X},\mathsf{Q}) \neq 0$. We use the method of Section 3 by defining the polynomial

$$\tilde{F}_\Sigma(\mathsf{X},\mathsf{Q},\mathsf{S}) := ((\Sigma - \mathsf{X})\mathsf{Q} - \mathbf{Z}_{\mathcal{S}}(\Sigma))\,\mathsf{S} - 1 \ .$$

We index F and $\tilde{F}$ with Σ instead of giving Σ as a formal argument. We do it because Σ (a trapdoor indeterminate, with various powers like $[\sigma^i]_1$ being present in the CRS) has a different semantics compared to indeterminates X, Q, and S that correspond to the argument elements. In particular, $[\sigma^i]_1$ do not have to stay hidden in the set (non-)membership NIZK. Crucially, this allows to think of F_Σ and $\tilde{F}_\Sigma$ as low-degree polynomials with coefficients from $\mathcal{R} = \mathbb{Z}_p[\Sigma]$.

Since F_Σ and $\bar{F}_\Sigma$ have degrees ≤ 2 and ≤ 3, they have respectively 2×2 and 3×3 QDRs $\boldsymbol{C}_\Sigma(\mathsf{X},\mathsf{Q})$ and $\bar{\boldsymbol{C}}_\Sigma(\mathsf{X},\mathsf{Q},\mathsf{S})$. We construct these QDRs from algebraic branching programs for F_Σ and $\bar{F}_\Sigma$. See Fig. 6 for the description of the resulting ABP and QDR for F_Σ and $\bar{F}_\Sigma$. The membership (resp., non-membership) argument verifier needs to check that $\det \boldsymbol{C}(\chi,\mathsf{q}) = 0$ (resp., $\det \bar{\boldsymbol{C}}(\chi,\mathsf{q},\mathsf{s}) = 0$).

Membership Argument. Since we construct a determinantal accumulator, we check $\det \boldsymbol{C}(\chi,\mathsf{q}) = 0$ by using the hints $[\boldsymbol{\gamma}]_1$ and $[\delta]_2$. The membership argument verifier checks that $[\boldsymbol{C}(\boldsymbol{\chi})]_1 \bullet \left[\begin{smallmatrix}\mathsf{e}\\ \delta\end{smallmatrix}\right]_2 = [\boldsymbol{\gamma}]_1 \bullet [1]_2$, which can be rewritten as checking

$$\begin{aligned} ([\sigma]_1 - [\chi]_1) \bullet [\mathsf{e}]_2 - [1]_1 \bullet [\delta]_2 &= [\gamma_1]_1 \bullet [1]_2 \ , \\ -[\mathbf{Z}_{\mathcal{S}}(\sigma)]_1 \bullet [\mathsf{e}]_2 + [\mathsf{q}]_1 \bullet [\delta]_2 &= [\gamma_2]_1 \bullet [1]_2 \ . \end{aligned} \tag{2}$$

Here, $[\chi]_1$ is the input, $([\mathsf{q},\boldsymbol{\gamma}]_1, [\delta]_2)$ are parts of the (non-)membership argument, and $[\sigma, \mathbf{Z}_{\mathcal{S}}(\sigma)]_1$ can be computed from $\mathtt{crs}$.

Unfortunately, this is not sufficient. A maliciously chosen $\chi = \chi(\Sigma)$, $\mathsf{q} = \mathsf{q}(\Sigma)$, and $\delta = \delta(\Sigma)$ can depend non-trivially on σ. In a AGM security proof, Eq. (2) guarantees that $\mathbf{Z}_{\mathcal{S}}(\Sigma) = (\Sigma - \chi(\Sigma))\mathsf{q}(\Sigma)$ and thus $(\Sigma - \chi(\Sigma)) \mid \mathbf{Z}_{\mathcal{S}}(\Sigma)$. If χ is an integer, then from this we will get that $\mathbf{Z}_{\mathcal{S}}(\chi) = 0$. However, if χ is not

an integer (it depends on σ), then $\mathbf{Z}_{\mathcal{S}}(\chi) = 0$ does not follow. For example, to break the membership argument, the adversary can fix any $\delta_1, \delta_2 \in \mathbb{Z}_p$ and set $[\chi]_1 \leftarrow [\sigma]_1 - \delta_2[1]_1$, $[\delta]_2 \leftarrow \delta_1[1]_2 + \delta_2[\mathsf{e}]_2$, $[\mathsf{q}]_1 \leftarrow [\mathbf{Z}_{\mathcal{S}}(\sigma)]_1/\delta_2$, $[\gamma_1]_1 \leftarrow -[\delta_1]_1$, $[\gamma_2]_1 \leftarrow \delta_1/\delta_2 \cdot [\mathbf{Z}_{\mathcal{S}}(\sigma)]_1$. This results in Eq. (2) holding and thus breaks the F-collision-resistance of the version of AC^* that only uses Eq. (2) as verification equations. Breaking F-collision-resistance of $\mathsf{ACC}_{\mathsf{Nguyen}}$ is even more trivial.[6]

To counteract this problem, we must guarantee that χ does not depend on σ. We do this by introducing an additional trapdoor τ. We then slightly modify Eq. (2), making the checks explicitly dependent on τ. The resulting modified checks result in b_1 and b_2 in the final construction of AC^* in Fig. 7.

Since now $\mathtt{crs}$ depends on τ, the adversary can make its outputs depend on τ; this opens a new cheating avenue. Hence, our use of τ is non-trivial, especially since we achieve F-collision-resistance without hampering the efficiency of AC^*. We explicitly multiply each term of type $[\alpha]_1 \bullet [\beta]_2$ in b_1 and b_2 by τ, except the terms $[\mathsf{q}]_1 \bullet [\delta]_2$ and $[\boldsymbol{\gamma}]_1 \bullet [1]_2$. In the AGM security proof, we get that values like χ, which are multiplied by τ, are in the span of 1 (that is, integers). However, q must be a polynomial (it depends on σ), that is, in the span of $\{\sigma^i\tau\}$; thus we do not multiply $[\mathsf{q}]_1 \bullet [\delta]_2$ by τ. The same holds for γ_2. Finally, it is not essential whether γ_1 depends on σ or not; not multiplying it by τ simplifies the AGM proof slightly since then we do not need to add $[\tau]_2$ to the CRS. Nevertheless, the AGM proof is very delicate.

Note that the verification equations ($b_1 = b_2 = \mathsf{true}$) are mathematically (but not computationally) equivalent to checking that $\boldsymbol{C}'(\chi, \mathsf{q})\left(\begin{smallmatrix}\mathsf{e}\\ \delta\end{smallmatrix}\right) = \boldsymbol{\gamma}$, where

$$\boldsymbol{C}'(\mathsf{X}, \mathsf{Q}) := \begin{pmatrix} (\Sigma - \mathsf{X})\mathsf{T} & -\mathsf{T} \\ -\mathbf{Z}_{\mathcal{S}}(\Sigma)\mathsf{T} & \mathsf{Q} \end{pmatrix} .$$

Here, $\det \boldsymbol{C}'(\mathsf{X}, \mathsf{Q}) = ((\Sigma - \mathsf{X})\mathsf{Q} - \mathbf{Z}_{\mathcal{S}}(\Sigma)\mathsf{T})\,\mathsf{T}$. That is, we really use the QDR framework of [CLPØ21]. The description of V in Fig. 7 just spells out how to do this verification in PPT.

Non-Membership Argument. The non-membership argument verifier must check that $[\bar{\boldsymbol{C}}(\boldsymbol{\chi})]_1 \bullet \left[\begin{smallmatrix}\mathsf{e}\\ \boldsymbol{\delta}\end{smallmatrix}\right]_2 = [\boldsymbol{\gamma}]_1 \bullet [1]_2$ (where now $\boldsymbol{\delta} \in \mathbb{Z}_p^2$ and $\boldsymbol{\gamma} \in \mathbb{Z}_p^3$; see Fig. 6), which can be rewritten as three checks

$$\begin{aligned} ([\sigma]_1 - [\chi]_1) \bullet [\mathsf{e}]_2 - [1]_1 \bullet [\delta]_2 &= [\gamma_1]_1 \bullet [1]_2 \ , \\ -[\mathbf{Z}_{\mathcal{S}}(\sigma)]_1 \bullet [\mathsf{e}]_2 + [\mathsf{q}]_1 \bullet [\delta_1]_2 - [1]_1 \bullet [\delta_2]_2 &= [\gamma_2]_1 \bullet [1]_2 \ , \\ -[1]_1 \bullet [\mathsf{e}]_2 + [\mathsf{s}]_1 \bullet [\delta_2]_2 &= [\gamma_3]_1 \bullet [1]_2 \ . \end{aligned} \tag{3}$$

As in the case of the membership argument, we need to modify the first two equations by using τ. However, since we require s to be an integer, we do not have to modify the third verification equation.

[6] In the collision-resistance proof of $\mathsf{ACC}_{\mathsf{Nguyen}}$, χ and r are given as integers and thus do not depend on σ. Such a problem did also not exist in [CH20,CLPØ21] since there the CRS only contained a single element $[\mathsf{e}]_2$ and thus did not depend on σ.

$\mathsf{Kgen}(\mathsf{p}, N)$: $\sigma, \tau, \mathsf{e} \leftarrow_\$ \mathbb{Z}_p$; $\mathtt{crs} \leftarrow ([1, (\sigma^i\tau)_{i=0}^N]_1, [1, \mathsf{e}, \sigma\mathsf{e}, \tau\mathsf{e}]_2)$; $\mathtt{td} \leftarrow (\mathsf{e}, \tau)$; return $(\mathtt{crs}, \mathtt{td})$.

$\mathsf{Com}(\mathtt{crs}, \mathcal{S})$: $[\mathsf{C}_\mathcal{S}]_1 \leftarrow [\mathbf{Z}_\mathcal{S}(\sigma)\tau]_1$; return $\mathtt{crs}_{\mathtt{1par}} \leftarrow (\mathtt{crs}, [\mathsf{C}_\mathcal{S}]_1, \mathcal{S})$;

$\mathsf{P}(\mathtt{crs}_{\mathtt{1par}}, \chi)$: $\mathsf{r} \leftarrow \mathbf{Z}_\mathcal{S}(\chi)$; $f(X) \leftarrow (\mathbf{Z}_\mathcal{S}(X) - \mathsf{r})/(X - \chi)$; $[\mathsf{q}]_1 \leftarrow [f(\sigma)\tau]_1$;
if $\chi \in \mathcal{S}$ then
1. $\varrho_\delta \leftarrow_\$ \mathbb{Z}_p$; $[\boldsymbol{\gamma}]_1 \leftarrow -\left[\begin{smallmatrix} -\tau \\ \mathsf{q} \end{smallmatrix}\right]_1 \varrho_\delta$; $[\delta]_2 \leftarrow [\sigma\mathsf{e}]_2 - \chi[\mathsf{e}]_2 - \varrho_\delta[1]_2$;
2. $\psi \leftarrow ([\mathsf{q}, \boldsymbol{\gamma}]_1, [\delta]_2)$; // $3\mathfrak{g}_1 + \mathfrak{g}_2$

else
1. $\mathsf{s} \leftarrow \frac{1}{\mathsf{r}}$; $\boldsymbol{\varrho_\delta} \leftarrow_\$ \mathbb{Z}_p^2$; $[\boldsymbol{\gamma}]_1 \leftarrow -\left[\begin{smallmatrix} -\tau & 0 \\ \mathsf{q} & -\tau \\ 0 & \mathsf{s} \end{smallmatrix}\right]_1 \boldsymbol{\varrho_\delta}$; $[\boldsymbol{\delta}]_2 \leftarrow \left(\begin{smallmatrix} [\sigma\mathsf{e}]_2 - \chi[\mathsf{e}]_2 \\ \mathsf{r}\cdot[\mathsf{e}]_2 \end{smallmatrix}\right) - \boldsymbol{\varrho_\delta}[1]_2$;
2. $\psi \leftarrow ([\mathsf{q}, \mathsf{s}, \boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2)$; // $5\mathfrak{g}_1 + 2\mathfrak{g}_2$

return ψ;

$\mathsf{V}(\mathtt{crs}_{\mathtt{1par}}, [\chi]_1, \psi)$: $mem \leftarrow \mathsf{NotMember}$;
If ψ parses as $\psi = ([\mathsf{q}, \boldsymbol{\gamma}]_1, [\delta]_2)$ then $mem \leftarrow \mathsf{Member}$;
If $mem = \mathsf{Member}$ then
1. $b_1 \leftarrow [\sigma\tau]_1 \bullet [\mathsf{e}]_2 - [\chi]_1 \bullet [\tau\mathsf{e}]_2 - [\tau]_1 \bullet [\delta]_2 \stackrel{?}{=} [\gamma_1]_1 \bullet [1]_2$;
2. $b_2 \leftarrow -[\mathsf{C}_\mathcal{S}]_1 \bullet [\mathsf{e}]_2 + [\mathsf{q}]_1 \bullet [\delta]_2 \stackrel{?}{=} [\gamma_2]_1 \bullet [1]_2$;
3. if b_1 and b_2 then return Member else return Error;

else
1. $\bar{b}_1 \leftarrow ([\sigma]_1 - [\chi]_1) \bullet [\tau\mathsf{e}]_2 - [\tau]_1 \bullet [\delta_1]_2 \stackrel{?}{=} [\gamma_1]_1 \bullet [1]_2$;
2. $\bar{b}_2 \leftarrow -[\mathsf{C}_\mathcal{S}]_1 \bullet [\mathsf{e}]_2 + [\mathsf{q}]_1 \bullet [\delta_1]_2 - [\tau]_1 \bullet [\delta_2]_2 \stackrel{?}{=} [\gamma_2]_1 \bullet [1]_2$;
3. $\bar{b}_3 \leftarrow -[1]_1 \bullet [\mathsf{e}]_2 + [\mathsf{s}]_1 \bullet [\delta_2]_2 \stackrel{?}{=} [\gamma_3]_1 \bullet [1]_2$;
4. if $\bar{b}_1$ and $\bar{b}_2$ and $\bar{b}_3$ then return $\mathsf{NotMember}$ else return Error;

Fig. 7. The new $[\cdot]_1$-collision-resistant determinantal universal accumulator AC^*.

The verification equations (that is, $\bar{b}_1 = \bar{b}_2 = \bar{b}_3 = \mathsf{true}$, see Fig. 7) are equivalent to checking that $\bar{\boldsymbol{C}}'(\chi, \mathsf{q}, \mathsf{s})\left(\begin{smallmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{smallmatrix}\right) = \boldsymbol{\gamma}$, where

$$\bar{\boldsymbol{C}}'(\mathsf{X}, \mathsf{Q}, \mathsf{S}) := \begin{pmatrix} (\Sigma - \mathsf{X})\mathsf{T} & -\mathsf{T} & 0 \\ -\mathbf{Z}_\mathcal{S}(\Sigma)\mathsf{T} & \mathsf{Q} & -\mathsf{T} \\ -1 & 0 & \mathsf{S} \end{pmatrix},$$

with $\det \bar{\boldsymbol{C}}'(\mathsf{X}, \mathsf{Q}) = ((\Sigma - \chi)\mathsf{Q} - \mathbf{Z}_\mathcal{S}(\Sigma)\mathsf{T})\,\mathsf{s}\mathsf{T} - \mathsf{T}^2$.

Description. We depict AC^* in Fig. 7. As explained before, the membership verifier checks (on pairings) that $\boldsymbol{C}'(\chi, \mathsf{q}) \cdot \left(\begin{smallmatrix} \mathsf{e} \\ \delta \end{smallmatrix}\right) = \boldsymbol{\gamma}$, and the non-membership verifier checks that $\bar{\boldsymbol{C}}'(\chi, \mathsf{q}, \mathsf{s}) \cdot \left(\begin{smallmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{smallmatrix}\right) = \boldsymbol{\gamma}$. Fig. 7 does it in PPT.

Lemma 1. AC^* *is perfectly complete.*

Proof. One can straightforwardly check that the choice of $\boldsymbol{\varrho_\delta}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\delta}$ is consistent with Fig. 3 when one uses the correct matrices $\boldsymbol{C}'$ and $\bar{\boldsymbol{C}}'$. Completeness follows straightforwardly. In particular, writing $\boldsymbol{C}' = (\boldsymbol{h}' \| \boldsymbol{T}')$, we get that $\boldsymbol{h}' = \boldsymbol{T}'\mathsf{w}'$, where $\mathsf{w}' = -(\sigma - \chi)$. This explains why say $[\delta]_2 = -\mathsf{w}'[\mathsf{e}]_2 - \varrho_\delta[1]_2 = [(\sigma - \chi)\mathsf{e}]_2 - \varrho_\delta[1]_2 = [\sigma\mathsf{e}]_2 - \chi[\mathsf{e}]_2 - \varrho_\delta[1]_2$. Then, say $b_1 = \mathsf{true}$

since $(\sigma-\chi)\tau\mathsf{e}-\tau\delta=\gamma_1 \iff (\sigma-\chi)\tau\mathsf{e}-\tau((\sigma-\chi)\mathsf{e}-\varrho_\delta)=\tau\varrho_\delta$, which is trivially true. In the case of non-membership proof, writing $\bar{\boldsymbol{C}}'=(\bar{\boldsymbol{h}}'\|\bar{\boldsymbol{T}}')$, we get similarly that $\bar{\boldsymbol{h}}'=\bar{\boldsymbol{T}}'\bar{\mathbf{w}}'$, where $\bar{\mathbf{w}}'=\begin{pmatrix}-(\sigma-\chi)\\ -\mathsf{r}\end{pmatrix}$. □

On Semantics of Non-Membership. Recall that AC^* must be F-collision-resistant. Since the CRS contains trapdoor-dependent elements, one must make it precise how to define non-membership. As a motivating example, if $\mathcal{S}=\{0,1\}$, then $[\chi]_1 \leftarrow [\sigma]_1$ satisfies $\chi\in\mathcal{S}$ iff $\sigma\in\{0,1\}$. The AGM security proof handles σ as an indeterminate, and thus it cannot decide whether σ (or, more generally, some known affine map of σ) belongs to $\mathcal{S}$. To avoid such artefacts, we constructed AC^* so that the verifier returns Error when the prover makes $[\chi]_1$ to depend on $[\sigma]_1$ (see the proof of Theorems 1 and 2). While we do not do it here, it allows one to define the extractability of the accumulator naturally; from the proof of Theorems 1 and 2, it is easy to see that AC^* is extractable.

6 AC^*'s F-Collision-Resistance

The actual F-collision-resistance proof is complicated. We first define two tautological assumptions N-DETACM and N-$\mathsf{DETACNM}$ that essentially state that AC^* is F-collision-resistant against adversaries that try to create fake membership (resp., non-membership) arguments. After that, we prove in AGM that DETACM and $\mathsf{DETACNM}$ reduce to PDL.

The most efficient structure-preserving signatures are proven to be secure in the AGM (or in the generic group model), though the assumption of their security by itself is a falsifiable assumption. We can similarly prove the security of AC^* in AGM. However, the collision-resistance of an accumulator is a much simpler (in particular, it is non-interactive) assumption than the unforgeability of a signature scheme and thus the tautological assumption looks less intimidating.

6.1 DETACM And DETACNM

Next, we define assumptions N-DETACM and N-$\mathsf{DETACNM}$.

Definition 5. *Let $\mathcal{A}$ be a PPT adversary. Let $N=\mathsf{poly}(\lambda)$. N-DETACM holds relative to Pgen, if for every PPT $\mathcal{A}$,*

$$\Pr\left[\begin{array}{c|c}\mathcal{S}\in\mathcal{D}^{\le N}\wedge\\ \chi\notin\mathcal{S}\wedge\\ \boldsymbol{C}'(\chi,\mathsf{q})\left(\begin{smallmatrix}\mathsf{e}\\ \delta\end{smallmatrix}\right)=\boldsymbol{\gamma}\end{array}\middle|\begin{array}{c}\mathsf{p}\leftarrow\mathsf{Pgen}(1^\lambda);\sigma,\tau,\mathsf{e}\leftarrow_\$\mathbb{Z}_p;\\ \mathtt{crs}\leftarrow(\mathsf{p},[1,(\sigma^i\tau)_{i=0}^N]_1,[1,\mathsf{e},\sigma\mathsf{e},\tau\mathsf{e}]_2);\\ (\mathcal{S},[\chi,\mathsf{q},\boldsymbol{\gamma}]_1,[\delta]_2)\leftarrow\mathcal{A}(\mathtt{crs});\\ \boldsymbol{C}'(\chi,\mathsf{q})\leftarrow\left(\begin{smallmatrix}(\sigma-\chi)\tau & -\tau\\ -\mathbf{Z}_{\mathcal{S}}(\sigma)\tau & \mathsf{q}\end{smallmatrix}\right)\end{array}\right]\approx_c 0\ .$$

N-$\mathsf{DETACNM}$ holds relative to Pgen, if for every PPT $\mathcal{A}$,

$$\Pr\left[\begin{array}{c|c}\mathcal{S}\in\mathcal{D}^{\le N}\wedge\\ \chi\in\mathcal{S}\wedge\\ \bar{\boldsymbol{C}}'(\chi,\mathsf{q},\mathsf{s})\left(\begin{smallmatrix}\mathsf{e}\\ \boldsymbol{\delta}\end{smallmatrix}\right)=\boldsymbol{\gamma}\end{array}\middle|\begin{array}{c}\mathsf{p}\leftarrow\mathsf{Pgen}(1^\lambda);\sigma,\tau,\mathsf{e}\leftarrow_\$\mathbb{Z}_p;\\ \mathtt{crs}\leftarrow(\mathsf{p},[1,(\sigma^i\tau)_{i=0}^N]_1,[1,\mathsf{e},\sigma\mathsf{e},\tau\mathsf{e}]_2);\\ (\mathcal{S},[\chi,\mathsf{q},\mathsf{s},\boldsymbol{\gamma}]_1,[\boldsymbol{\delta}]_2)\leftarrow\mathcal{A}(\mathtt{crs});\\ \bar{\boldsymbol{C}}'(\chi,\mathsf{q},\mathsf{s})\leftarrow\left(\begin{smallmatrix}(\sigma-\chi)\tau & -\tau & 0\\ -\mathbf{Z}_{\mathcal{S}}(\sigma)\tau & \mathsf{q} & -\tau\\ -1 & 0 & \mathsf{s}\end{smallmatrix}\right)\end{array}\right]\approx_c 0\ .$$

$\mathcal{B}(\mathtt{crs} = (\mathsf{p}, [1, (\sigma^i\tau)_{i=0}^N]_1, [1, \mathsf{e}, \sigma\mathsf{e}, \tau\mathsf{e}]_2))$
$(\mathcal{S}, [\chi]_1, \psi) \leftarrow \mathcal{A}(\mathtt{crs})$;
return $(\mathcal{S}, [\chi]_1, \psi)$; **endif**

Fig. 8. The adversary $\mathcal{B}$ in the proof of Lemma 2

Compared to CED, DETACM and DETACNM do not rely on the (possibly, inefficiently verifiable) condition that $\boldsymbol{C}(\boldsymbol{\chi})$ has a full rank. Thus, importantly, DETACM and DETACNM are efficiently verifiable and thus falsifiable. For example, as explained above, the verification $\bar{\boldsymbol{C}}'(\chi, \mathsf{q}, \mathsf{s})(\begin{smallmatrix}\mathsf{e}\\ \boldsymbol{\delta}\end{smallmatrix}) = \boldsymbol{\gamma}$ is equivalent to checking that $\bar{b}_1$, $\bar{b}_2$, and $\bar{b}_3$ hold. Thus, it can be checked efficiently.

6.2 F-Collision-Resistance of AC^*

Lemma 2 is trivial since DETACM and DETACNM are tautological assumptions for the F-collision-resistance of AC^*. The complicated step (see Section 6.3) is establishing that DETACM and DETACNM are secure in the AGM.

Lemma 2. *Let $F = [\cdot]_1$ and $N = \mathsf{poly}(\lambda)$. AC^* is F-collision-resistant under N-DETACM and N-DETACNM.*

Proof. Let $\mathcal{A}$ be an F-collision-resistance (see Definition 4) adversary for AC^*, such that $\mathsf{Adv}^{\mathrm{f-cr}}_{\mathsf{Pgen},F,\mathsf{AC}^*,\mathcal{A}}(\lambda) = \varepsilon_{\mathcal{A}}$ for non-negligible $\varepsilon_{\mathcal{A}}$. In Fig. 8, we depict a trivial DETACM/DETACNM adversary $\mathcal{B}$. Clearly, with probability at least $\varepsilon_{\mathcal{A}}$, $\mathcal{B}$ succeeds in breaking N-DETACM (resp., N-DETACNM), given $\mathcal{A}$ outputs an accepting fake membership (resp., non-membership) argument. □

6.3 AGM Security of DETACM And DETACNM

Theorem 1. *If $(N+1, 2)$-PDL holds, then N-DETACM is secure in the AGM.*

Proof. Let $\mathcal{A}_{\mathrm{alg}}$ be an algebraic DETACM adversary. Assume that $\mathcal{A}_{\mathrm{alg}}(\mathtt{crs})$ outputs $\psi = (\mathcal{S}, [\chi, \mathsf{q}, \boldsymbol{\gamma}]_1, [\delta]_2)$, such that V accepts with a non-negligible probability. Since $\mathcal{A}_{\mathrm{alg}}$ is algebraic, with every group element $G \in \mathbb{G}_\iota$, it also outputs a vector $\boldsymbol{a}$ explaining how G is constructed from the elements of $\mathtt{crs}$ that belong to $\mathbb{G}_\iota$. Next, we will make this more precise.

Let $\boldsymbol{X} = (\Sigma, \mathsf{T}, \mathsf{E})$ and $\boldsymbol{x} = (\sigma, \tau, \mathsf{e})$. Here, say T is the indeterminate corresponding to the trapdoor τ. We express each output of $\mathcal{A}_{\mathrm{alg}}$ as a polynomial evaluation, with say $[\chi]_1 = [\chi(\boldsymbol{x})]_1$. The involved polynomials are

$$\chi(\boldsymbol{X}) = \chi_1(\Sigma)\mathsf{T} + \chi_2 \ , \qquad \mathsf{q}(\boldsymbol{X}) = \mathsf{q}_1(\Sigma)\mathsf{T} + \mathsf{q}_2 \ ,$$
$$\gamma_1(\boldsymbol{X}) = \gamma_{11}(\Sigma)\mathsf{T} + \gamma_{12} \ , \qquad \gamma_2(\boldsymbol{X}) = \gamma_{21}(\Sigma)\mathsf{T} + \gamma_{22} \ ,$$
$$\delta(\boldsymbol{X}) = \delta_1 + \delta_2\mathsf{E} + \delta_3\Sigma\mathsf{E} + \delta_4\mathsf{T}\mathsf{E} \ ,$$

where each polynomial (like q_1) on the RHS is of degree $\leq N$. That is, the algebraic adversary $\mathcal{A}_{\text{alg}}$ also outputs coefficients of all above polynomials. The DETACM verifier's checks guarantee that $V_1(\sigma, \tau, \mathsf{e}) = V_2(\sigma, \tau, \mathsf{e}) = 0$, where

$$V_1(\boldsymbol{X}) = ((\Sigma - \chi(\boldsymbol{X}))\,\mathsf{E} - \delta(\boldsymbol{X})) \cdot \mathsf{T} - \gamma_1(\boldsymbol{X}) \ ,$$
$$V_2(\boldsymbol{X}) = (\mathsf{r}(\boldsymbol{X}) - \mathbf{Z}_{\mathcal{S}}(\Sigma))\,\mathsf{TE} + \mathsf{q}(\boldsymbol{X})\delta(\boldsymbol{X}) - \gamma_2(\boldsymbol{X}) \ .$$

Consider separately the cases (1) $V_1 = V_2 = 0$ as polynomials, and (2) either $V_1 \neq 0$ or $V_2 \neq 0$.

Case 1. First, assume $V_1 = V_2 = 0$ as a polynomial. Think of the polynomials as members of $\mathcal{R}[\mathsf{T}, \mathsf{E}]$, where $\mathcal{R} = \mathbb{Z}_p[\Sigma]$. We now enlist the coefficients of $\mathsf{T}^i\mathsf{E}^j$ in both V_1 and V_2, highlighting the coefficients that are actually needed in this proof (we give other coefficients only for the sake of completeness):

(i,j) V_1
$(2,1)$ $-\delta_4 - \chi_1(\Sigma)$
$(1,1)$ $-\delta_2 + (1-\delta_3)\,\Sigma - \chi_2$
$(1,0)$ $-\gamma_{11}(\Sigma) - \delta_1$
$(0,0)$ $-\gamma_{12}$

(i,j) V_2
$(2,1)$ $\delta_4\mathsf{q}_1(\Sigma)$
$(1,1)$ $\delta_4\mathsf{q}_2 + (\delta_2 + \delta_3\Sigma)\,\mathsf{q}_1(\Sigma) - \mathbf{Z}_{\mathcal{S}}(\Sigma)$
$(1,0)$ $\delta_1\mathsf{q}_1(\Sigma) - \gamma_{21}(\Sigma)$
$(0,1)$ $(\delta_2 + \delta_3\Sigma)\mathsf{q}_2$
$(0,0)$ $-\gamma_{22} + \delta_1\mathsf{q}_2$

For example, the coefficient of $\mathsf{T}^2\mathsf{E}^1 = \mathsf{T}^2\mathsf{E}$ in V_1 is $-\delta_4 - \chi_1(\Sigma)$. Since $V_i = 0$ as a polynomial, the coefficient of any monomial $\mathsf{T}^j\mathsf{E}^k$ in any V_i is also 0.

From the coefficient of $\mathsf{T}^2\mathsf{E}$ of V_1, we get $\chi_1(\Sigma) = -\delta_4$. From the coefficient of TE of V_1, after separating the coefficients of different Σ^i, we get $\delta_3 = 1$ and $\delta_2 = -\chi_2$. From the coefficient of $\mathsf{T}^2\mathsf{E}$ of V_2, we get $\delta_4\mathsf{q}_1(\Sigma) = 0$. Thus, either $\mathsf{q}_1(\Sigma) = 0$ or $\delta_4 = 0$. Taking into account what we already know, from the coefficient of TE of V_2, we get $\mathbf{Z}_{\mathcal{S}}(\Sigma) = \delta_4\mathsf{q}_2 + (\Sigma - \chi_2)\,\mathsf{q}_1(\Sigma)$. Recall that we have either $\mathsf{q}_1(\Sigma) = 0$ or $\delta_4 = 0$. If $\mathsf{q}_1(\Sigma) = 0$, then $\mathbf{Z}_{\mathcal{S}}(\Sigma) = \delta_4\mathsf{q}_2 \in \mathbb{Z}_p$, a contradiction. Hence, $\delta_4 = 0$. Thus, $\mathbf{Z}_{\mathcal{S}}(\Sigma) = (\Sigma - \chi_2)\,\mathsf{q}_1(\Sigma)$ and $(\Sigma - \chi_2) \mid \mathbf{Z}_{\mathcal{S}}(\Sigma)$, which gives us $\mathbf{Z}_{\mathcal{S}}(\chi_2) = 0$. Moreover, $\chi(\boldsymbol{X}) = \chi_1(\Sigma)\mathsf{T} + \chi_2 = \chi_2$, and thus we have proven AGM security in Case 1.

Case 2. The case $V_i \neq 0$ for some i can be handled in a standard way. Assume for example that $V_2 \neq 0$. We construct a PDL reduction $\mathcal{B}(\{[\sigma^i]_1\}_{i=0}^{N+1}, \{[\sigma^i]_1\}_{i=0}^{2})$. $\mathcal{B}$ samples $\alpha_1, \alpha_2, \beta_1, \beta_2 \leftarrow_{\$} \mathbb{Z}_p$ and sets implicitly $\tau \leftarrow \alpha_1\sigma + \beta_1$ and $\mathsf{e} \leftarrow \alpha_2\sigma + \beta_2$. Then, $\mathcal{B}$ creates $\mathtt{crs}$ for the DETACM adversary $\mathcal{A}_{\text{alg}}$ and calls $\mathcal{A}_{\text{alg}}$ with $\mathtt{crs}$. After obtaining π, together with the coefficients of the polynomials like $\chi(\Sigma)$, from $\mathcal{A}_{\text{alg}}$, $\mathcal{B}$ reconstructs the coefficients of the degree-$\leq (N+2)$ polynomial V_2 (which is now univariate since τ and e are affine maps of σ). We know $V_2 \neq 0$ but $V_2(\sigma) = 0$. $\mathcal{B}$ factorizes V_2 and finds up to $N+2$ roots x_i of V_2. $\mathcal{B}$ tests which one of them is equal to σ, and returns σ. □

Theorem 2. *If* $(N+1, 2)$-PDL *holds, then* N-$\mathsf{DETACNM}$ *is secure in the AGM.*

We postpone the proof of this theorem to Appendix A.1.

7 New Set (Non-)Membership NIZK

Next, we use AC^* to construct a succinct set (non-)membership NIZK $\boldsymbol{\Pi}^*$. First, $\boldsymbol{\Pi}^*$'s CRS is equal to AC^*'s CRS. Second, the NIZK prover proves that AC^*'s honest verifier accepts the encrypted χ and the encrypted accumulator argument $\psi = \mathsf{AC}^*.\mathsf{P}(\mathtt{crs}, \mathcal{S}, \chi)$. That is, the prover encrypts χ and ψ, and then proves that the verification equation is satisfied.

Description. Following the described blueprint, we construct the new set (non-)membership NIZK $\boldsymbol{\Pi}^*$ (see Fig. 9). $\boldsymbol{\Pi}^*$ handles both $\mathcal{L}^{\mathsf{sm}}_{\mathtt{lpar}}$ (set membership arguments, $mem = \mathsf{Member}$) and $\overline{\mathcal{L}}_{\mathtt{lpar}}$ (set non-membership arguments, $mem = \mathsf{NotMember}$). The prover of $\boldsymbol{\Pi}^*$ implements the prover of AC^* but it also additionally encrypts all $\mathbb{G}_1$. To make the verification on ciphertexts possible, the prover outputs addtional randomizer hints $[\boldsymbol{z}]_2$. The verifier performs AC^* verification on ciphertexts (this relies on the homomorphic properties of Elgamal), taking $[\boldsymbol{z}]_2$ into account. $\boldsymbol{\Pi}^*$ also defines the simulator algorithm.

Alternatively, $\boldsymbol{\Pi}^*$ is a version of $\boldsymbol{\Pi}_{\mathsf{clp}\emptyset}$ for the concrete choice of the QDRs (and different CRS). To see the connection between Fig. 9 and Fig. 3, note that

$$\boldsymbol{C}'(\mathsf{X},\mathsf{Q}) = \underbrace{\begin{pmatrix} \Sigma\mathsf{T} & -1 \\ -\mathbf{Z}_{\mathcal{S}(\Sigma)}\mathsf{T} & 0 \end{pmatrix}}_{\boldsymbol{Q}} + \underbrace{\begin{pmatrix} -\mathsf{T} & 0 \\ 0 & 0 \end{pmatrix}}_{\boldsymbol{P}_1}\mathsf{X} + \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}}_{\boldsymbol{P}_2}\mathsf{Q} \ .$$

For example, starting with Fig. 3, $[\boldsymbol{z}]_2 = (\sum_{k=1}^{\nu} \varrho_k \boldsymbol{P}_k \left[\begin{smallmatrix}\mathsf{e}\\ \delta\end{smallmatrix}\right]_2) - \boldsymbol{\varrho}_\gamma[1]_2 = \varrho_\chi \left(\begin{smallmatrix}-\tau & 0\\ 0 & 0\end{smallmatrix}\right)\left[\begin{smallmatrix}\mathsf{e}\\ \delta\end{smallmatrix}\right]_2 + \varrho_{\mathsf{q}}\left(\begin{smallmatrix}0 & 0\\ 0 & 1\end{smallmatrix}\right)\left[\begin{smallmatrix}\mathsf{e}\\ \delta\end{smallmatrix}\right]_2 - \boldsymbol{\varrho}_\gamma[1]_2 = \varrho_\chi \left[\begin{smallmatrix}-\tau\mathsf{e}\\ 0\end{smallmatrix}\right]_2 + \varrho_{\mathsf{q}}\left[\begin{smallmatrix}0\\ \delta\end{smallmatrix}\right]_2 - \boldsymbol{\varrho}_\gamma[1]_2 = \left(\begin{smallmatrix}-\varrho_\chi[\tau\mathsf{e}]_2\\ \varrho_{\mathsf{q}}[\delta]_2\end{smallmatrix}\right) - \boldsymbol{\varrho}_\gamma[1]_2$. One can represent $\bar{\boldsymbol{C}}'(\mathsf{X},\mathsf{Q},\mathsf{R})$ similarly.

Clearly, $\boldsymbol{\Pi}^*$ is commit-and-prove, updatable, and universal.

7.1 Security

Theorem 3. *The set membership argument $\boldsymbol{\Pi}^*$ in Fig. 9 is perfectly complete. Assuming Elgamal is IND-CPA secure, it is computationally zero-knowledge.*

We postpone the proof of this theorem to Appendix A.2. The following straightforward soundness reduction relies on the security of AC^*.

Theorem 4. *Let $\ell = 2$ and $\mathsf{k} = 1$. Let $\mathcal{D}_{\mathsf{k}}$ be the distribution of $\left[\begin{smallmatrix}1\\ \mathsf{e}\end{smallmatrix}\right]_2$ for $\mathsf{e} \leftarrow\!\!\$\, \mathbb{Z}_p$. Let $N = \mathsf{poly}(\lambda)$ be an upper bound on $|\mathcal{S}|$. The set membership NIZK $\boldsymbol{\Pi}^*$ in Fig. 9 is sound, assuming AC^* is $[\cdot]_1$-collision-resistant.*

Proof. Let $\mathcal{A}_{\boldsymbol{\Pi}^*}$ be a successful soundness adversary (as defined in Section 2.1) for $\boldsymbol{\Pi}^*$. That is, with a non-negligible probability $\varepsilon_{\mathcal{A}_{\boldsymbol{\Pi}^*}}$, for $(\mathsf{p}, \mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{K}_{\mathtt{crs}}(1^\lambda)$ and for any valid $\mathtt{lpar}$, $\mathcal{A}_{\boldsymbol{\Pi}^*}(\mathtt{crs}, \mathtt{lpar})$ outputs $(\mathbb{x}, \pi)$, such that $\mathsf{V}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x}, \pi) = 1$ but either (1) π is a membership argument but $\mathbb{x} \notin \mathcal{L}^{\mathsf{sm}}_{\mathtt{lpar}}$ or (1) π is a non-membership argument but $\mathbb{x} \in \mathcal{L}^{\mathsf{sm}}_{\mathtt{lpar}}$.

Decrypting all verification equations, the verifier checks guarantee that the AC^* verifier accepts $[\chi]_1 \leftarrow \mathsf{Dec}_{\mathsf{sk}}(\mathbf{ct}_\chi)$. Essentially, the constructed adversary

$\mathsf{Pgen}(1^\lambda)$: $\mathsf{p} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, [1]_1, [1]_2) \leftarrow \mathsf{Pgen}(1^\lambda)$.

$\mathsf{Kgen}(\mathsf{p})$: $(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{AC}^*.\mathsf{Kgen}(\mathsf{p})$;

$\mathsf{Com}(\mathtt{crs}, \mathtt{lpar} = (\mathsf{pk}, \mathcal{S}))$: $\mathsf{AC}^*.\mathtt{lpar} \leftarrow \mathcal{S}$; $\mathsf{AC}^*.\mathtt{crs}_{\mathtt{lpar}} \leftarrow \mathsf{AC}^*.\mathsf{Com}(\mathtt{crs}, \mathsf{AC}^*.\mathtt{lpar})$; return $\mathtt{crs}_{\mathtt{lpar}} \leftarrow (\mathsf{AC}^*.\mathtt{crs}_{\mathtt{lpar}}, \mathsf{pk})$;

$\mathsf{P}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x} = [\mathbf{ct}_\chi]_1, \mathbb{w} = (\chi, \varrho_\chi))$:

$\mathsf{AC}^*.\psi \leftarrow \mathsf{AC}^*.\mathsf{P}(\mathsf{AC}^*.\mathtt{crs}_{\mathtt{lpar}}, \chi)$; // $\psi = ([\mathsf{q}, \boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2)$ or $\psi = ([\mathsf{q}, \mathsf{s}, \boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2)$

$\varrho_\mathsf{q} \leftarrow\!\!\$\, \mathbb{Z}_p$; $[\mathsf{ct}_\mathsf{q}]_1 \leftarrow \mathsf{Enc}_\mathsf{pk}([\mathsf{q}]_1; \varrho_\mathsf{q})$;

If $\chi \in \mathcal{S}$ then

1. $\boldsymbol{\varrho}_\gamma \leftarrow\!\!\$\, \mathbb{Z}_p^2$; $[\mathbf{ct}_\gamma]_1 \leftarrow \mathsf{Enc}_\mathsf{pk}([\boldsymbol{\gamma}]_1; \boldsymbol{\varrho}_\gamma) \in \mathbb{G}_1^{2\times 2}$; $[\boldsymbol{z}]_2 \leftarrow \begin{pmatrix} -\varrho_\chi[\tau\mathsf{e}]_2 \\ \varrho_\mathsf{q}[\delta]_2 \end{pmatrix} - \boldsymbol{\varrho}_\gamma[1]_2 \in \mathbb{G}_2^2$;
2. $\pi \leftarrow ([\mathsf{ct}_\mathsf{q}, \mathbf{ct}_\gamma]_1, [\delta, \boldsymbol{z}]_2)$

else

1. $\varrho_\mathsf{s} \leftarrow\!\!\$\, \mathbb{Z}_p$; $[\mathsf{ct}_\mathsf{s}]_1 \leftarrow \mathsf{Enc}_\mathsf{pk}([\mathsf{s}]_1; \varrho_\mathsf{s}) \in \mathbb{G}_1^{1\times 2}$;
2. $\boldsymbol{\varrho}_\gamma \leftarrow\!\!\$\, \mathbb{Z}_p^3$; $[\mathbf{ct}_\gamma]_1 \leftarrow \mathsf{Enc}_\mathsf{pk}([\boldsymbol{\gamma}]_1; \boldsymbol{\varrho}_\gamma) \in \mathbb{G}_1^{3\times 2}$; $[\boldsymbol{z}]_2 \leftarrow \begin{pmatrix} -\varrho_\chi[\tau\mathsf{e}]_2 \\ \varrho_\mathsf{q}[\delta_1]_2 \\ \varrho_\mathsf{s}[\delta_2]_2 \end{pmatrix} - \boldsymbol{\varrho}_\gamma[1]_2 \in \mathbb{G}_2^3$;
3. $\pi \leftarrow ([\mathsf{ct}_\mathsf{q}, \mathsf{ct}_\mathsf{s}, \mathbf{ct}_\gamma]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2)$;

return π; // membership: $6\mathfrak{g}_1 + 3\mathfrak{g}_2$; non-membership: $10\mathfrak{g}_1 + 5\mathfrak{g}_2$

$\mathsf{Sim}(\mathtt{crs}_{\mathtt{lpar}}, \mathtt{td} = (\mathsf{e}, \tau), \mathbb{x} = [\mathbf{ct}_\chi]_1, mem \in \{\mathsf{Member}, \mathsf{NotMember}\})$:

If $mem = \mathsf{Member}$ then

1. $\delta \leftarrow\!\!\$\, \mathbb{Z}_p$; $\boldsymbol{z} \leftarrow\!\!\$\, \mathbb{Z}_p^2$; $\varrho_\mathsf{q} \leftarrow\!\!\$\, \mathbb{Z}_p$; $[\mathsf{ct}_\mathsf{q}]_1 \leftarrow \mathsf{Enc}_\mathsf{pk}(0; \varrho_\mathsf{q})$;
2. $[\mathbf{ct}_\gamma]_1 \leftarrow \begin{pmatrix} \mathsf{Enc}_\mathsf{pk}([\sigma\tau]_1; 0) - [\mathbf{ct}_\chi]_1 \cdot \tau & -\mathsf{Enc}_\mathsf{pk}([\tau]_1; 0) \\ -\mathsf{Enc}_\mathsf{pk}([\mathsf{C}_\mathcal{S}]_1; 0) & [\mathsf{ct}_\mathsf{q}]_1 \end{pmatrix} (\begin{smallmatrix} \mathsf{e} \\ \delta \end{smallmatrix}) - \mathsf{Enc}_\mathsf{pk}(\mathbf{0}; \boldsymbol{z})$;
3. $\pi \leftarrow ([\mathsf{ct}_\mathsf{q}, \mathbf{ct}_\gamma]_1, [\delta, \boldsymbol{z}]_2)$

else

1. $\boldsymbol{\delta} \leftarrow\!\!\$\, \mathbb{Z}_p^2$; $\boldsymbol{z} \leftarrow\!\!\$\, \mathbb{Z}_p^3$;
2. $\varrho_\mathsf{q}, \varrho_\mathsf{s} \leftarrow\!\!\$\, \mathbb{Z}_p$; $[\mathsf{ct}_\mathsf{q}]_1 \leftarrow \mathsf{Enc}_\mathsf{pk}(0; \varrho_\mathsf{q})$; $[\mathsf{ct}_\mathsf{s}]_1 \leftarrow \mathsf{Enc}_\mathsf{pk}(0; \varrho_\mathsf{s})$;
3. $[\mathbf{ct}_\gamma]_1 \leftarrow -\begin{pmatrix} \mathsf{Enc}_\mathsf{pk}([\sigma\tau]_1; 0) - [\mathbf{ct}_\chi]_1 \cdot \tau & -\mathsf{Enc}_\mathsf{pk}([\tau]_1; 0) & \mathsf{Enc}_\mathsf{pk}(0; 0) \\ -\mathsf{Enc}_\mathsf{pk}([\mathsf{C}_\mathcal{S}]_1; 0) & [\mathsf{ct}_\mathsf{q}]_1 & -\mathsf{Enc}_\mathsf{pk}([\tau]_1; 0) \\ -\mathsf{Enc}_\mathsf{pk}(1; 0) & \mathsf{Enc}_\mathsf{pk}(0; 0) & [\mathsf{ct}_\mathsf{s}]_1 \end{pmatrix} (\begin{smallmatrix} \mathsf{e} \\ \delta \end{smallmatrix}) - \mathsf{Enc}_\mathsf{pk}(\mathbf{0}; \boldsymbol{z})$;
4. $\pi \leftarrow ([\mathsf{ct}_\mathsf{q}, \mathsf{ct}_\mathsf{s}, \mathbf{ct}_\gamma]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2)$;

return π;

$\mathsf{V}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x} = [\mathbf{ct}_\chi]_1, \pi)$: $mem \leftarrow \mathsf{NotMember}$;

if π parses as $\pi = ([\mathsf{ct}_\mathsf{q}, \mathbf{ct}_\gamma]_1, [\delta, \boldsymbol{z}]_2)$ then $mem \leftarrow \mathsf{Member}$;

If $mem = \mathsf{Member}$ then check

1. $b_1 \leftarrow \mathsf{Enc}_\mathsf{pk}([\sigma\tau]_1; 0) \bullet [\mathsf{e}]_2 - [\mathbf{ct}_\chi]_1 \bullet [\tau\mathsf{e}]_2 - \mathsf{Enc}_\mathsf{pk}([\tau]_1; 0) \bullet [\delta]_2 \stackrel{?}{=} [\mathsf{ct}_{\gamma 1}]_1 \bullet [1]_2 + [z_1]_2 \bullet \mathsf{pk}$;
2. $b_2 \leftarrow -\mathsf{Enc}_\mathsf{pk}([\mathsf{C}_\mathcal{S}]_1; 0) \bullet [\mathsf{e}]_2 + [\mathsf{ct}_\mathsf{q}]_1 \bullet [\delta]_2 \stackrel{?}{=} [\mathsf{ct}_{\gamma 2}]_1 \bullet [1]_2 + [z_2]_2 \bullet \mathsf{pk}$;
3. if b_1 and b_2 then return Member else return Error;

else check

1. $\bar{b}_1 \leftarrow \mathsf{Enc}_\mathsf{pk}([\sigma\tau]_1; 0) \bullet [\mathsf{e}]_2 - [\mathbf{ct}_\chi]_1 \bullet [\tau\mathsf{e}]_2 - \mathsf{Enc}_\mathsf{pk}([\tau]_1; 0) \bullet [\delta_1]_2 \stackrel{?}{=} [\mathsf{ct}_{\gamma 1}]_1 \bullet [1]_2 + [z_1]_2 \bullet \mathsf{pk}$;
2. $\bar{b}_2 \leftarrow -\mathsf{Enc}_\mathsf{pk}([\mathsf{C}_\mathcal{S}]_1; 0) \bullet [\mathsf{e}]_2 + [\mathsf{ct}_\mathsf{q}]_1 \bullet [\delta_1]_2 - \mathsf{Enc}_\mathsf{pk}([\tau]_1; 0) \bullet [\delta_2]_2 \stackrel{?}{=} [\mathsf{ct}_{\gamma 2}]_1 \bullet [1]_2 + [z_2]_2 \bullet \mathsf{pk}$;
3. $\bar{b}_3 \leftarrow -\mathsf{Enc}(1; 0) \bullet [\mathsf{e}]_2 + [\mathsf{ct}_\mathsf{s}]_1 \bullet [\delta_2]_2 \stackrel{?}{=} [\mathsf{ct}_{\gamma 3}]_1 \bullet [1]_2 + [z_3]_2 \bullet \mathsf{pk}$;
4. if $\bar{b}_1$ and $\bar{b}_2$ and $\bar{b}_3$ then return $\mathsf{NotMember}$ else return Error;

Fig. 9. The new set (non-)membership NIZK $\boldsymbol{\Pi}^*$.

$\mathcal{B}_{cr}(\mathtt{crs} = (\mathsf{p}, [1, (\sigma^i\tau)_{i=0}^N]_1, [1, \mathsf{e}, \sigma\mathsf{e}, \tau\mathsf{e}]_2))$ // $[\cdot]_1$-CR adversary, see Definition 4

Choose any set $\mathcal{S}$ of size $\leq N$;
$\mathsf{sk} \leftarrow_\$ \mathbb{Z}_p$; $\mathsf{pk} \leftarrow [1\|\mathsf{sk}]_1$; $\mathtt{lpar} \leftarrow (\mathsf{pk}, \mathcal{S})$;
$\mathtt{crs}_{\mathtt{lpar}} \leftarrow \mathsf{Com}(\mathtt{crs}, \mathtt{lpar})$;
$(\mathbb{x}, \pi) \leftarrow \mathcal{A}_{\mathbf{\Pi}^*}(\mathtt{crs}_{\mathtt{lpar}})$;
$[\chi]_1 \leftarrow \mathsf{Dec}_{\mathsf{sk}}([\mathbf{ct}_\chi]_1)$; $[\mathsf{q}]_1 \leftarrow \mathsf{Dec}_{\mathsf{sk}}([\mathsf{ct}_\mathsf{q}]_1)$; $[\boldsymbol{\gamma}]_1 \leftarrow \mathsf{Dec}_{\mathsf{sk}}([\mathbf{ct}_\gamma]_1)$;
if π parses as $([\mathsf{ct}_\mathsf{q}, \mathbf{ct}_\gamma]_1, [\delta, \boldsymbol{z}]_2)$ **then** $\psi \leftarrow ([\mathsf{q}, \boldsymbol{\gamma}]_1, [\delta]_2)$; **return** $(\mathcal{S}, [\chi]_1, \psi)$;
else $[\mathsf{s}]_1 \leftarrow \mathsf{Dec}_{\mathsf{sk}}([\mathsf{ct}_\mathsf{s}]_1)$; $\psi \leftarrow ([\mathsf{q}, \mathsf{s}, \boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2)$; **return** $(\mathcal{S}, [\chi]_1, \psi)$; **fi**

Fig. 10. Reduction $\mathcal{B}_{cr}$ in the proof of $\mathbf{\Pi}^*$

$\mathcal{B}_{cr}$ (see Fig. 10), on its input, creates a new Elgamal key-pair. Based on that, $\mathcal{B}_{cr}$ then creates a correct $\mathtt{crs}_{\mathtt{lpar}}$ for $\mathcal{A}_{\mathbf{\Pi}^*}$. After obtaining $(\mathbb{x}, \pi)$ from $\mathcal{A}_{\mathbf{\Pi}^*}$, $\mathcal{B}_{cr}$ decrypts $\mathcal{A}_{\mathbf{\Pi}^*}$'s answer, obtaining and returning the input and the argument as expected from a $[\cdot]_1$-collision-resistance adversary.

Clearly, $\mathcal{B}_{cr}$ succeeds iff $\mathcal{A}_{\mathbf{\Pi}^*}$ succeeds. □

7.2 Efficiency

$\mathbf{\Pi}^*$'s CRS length is $N + 1$ elements of $\mathbb{G}_1$ and 4 elements of $\mathbb{G}_2$. The set membership argument length is $6\mathfrak{g}_1 + 3\mathfrak{g}_2$, which comes close to the $\mathbf{\Pi}_{\mathsf{clp}\emptyset}$ argument length $4\mathfrak{g}_1 + 3\mathfrak{g}_2$ for the simple OR language (this corresponds to $\ell = 2$). The difference comes from the fact that here we also need to encrypt AC^*'s argument ψ. On the other hand, the set non-membership argument length is ten elements of $\mathbb{G}_1$ and five elements of $\mathbb{G}_2$.

The prover's computation can be divided into precomputation and online computation. In precomputation, P computes $f(X)$ ($\Theta(|\mathcal{S}|)$ field operations) and $[\mathsf{q}]_1$ ($|\mathcal{S}|$ scalar multiplications in $\mathbb{G}_1$). In online precomputation, (1) the membership prover computes 8 scalar multiplications in $\mathbb{G}_1$ and 6 in $\mathbb{G}_2$ ($2\mathfrak{m}_1 + 2\mathfrak{m}_2$ to compute $\mathsf{AC}^*.\psi$ and $6\mathfrak{m}_1 + 4\mathfrak{m}_2$ in the rest of $\mathbf{\Pi}^*$), and (2) the non-membership prover computes fourteen scalar multiplications in $\mathbb{G}_1$ and ten in $\mathbb{G}_2$ ($4\mathfrak{m}_1 + 4\mathfrak{m}_2$ to compute $\mathsf{AC}^*.\psi$ and $10\mathfrak{m}_1 + 6\mathfrak{m}_2$ in the rest of $\mathbf{\Pi}^*$). (The online computation includes the computation of $[\mathsf{ct}_\mathsf{q}]_1$ and other ciphertexts.)

The set membership verifier's computation is dominated by fifteen pairings (eight to check b_1, seven to check b_2). However, two pairings (the pairings involved in $\mathsf{Enc}_{\mathsf{pk}}([\sigma\tau]_1; 0) \bullet [\mathsf{e}]_2$ and $\mathsf{Enc}_{\mathsf{pk}}([\mathsf{C}_\mathcal{S}]_1; 0) \bullet [\mathsf{e}]_2$) can be precomputed. Thus, online the verifier has to only compute thirteen pairings. Similarly, the set non-membership verifier's computation is dominated by 23 pairings (eight to check $\bar{b}_1$, eight to check $\bar{b}_2$, and seven to check $\bar{b}_3$), but three can be precomputed so the online computation is 20 pairings.

We refer to Table 1 for an extensive efficiency comparison.

8 On Handling Group Elements with CLPØ

The CLPØ NIZK [CLPØ21] works in the case where the prover knows all the elements of all DRs as integers. This seems to exclude applications where one needs to prove statements about group elements. In $\mathbf{\Pi}^*$, we overcome this issue by making the following observation. Consider the case of a single DR $\boldsymbol{C}(\boldsymbol{X}) = (h(\boldsymbol{X})\|T(\boldsymbol{X}))$, where $h(\boldsymbol{X})$ is a column vector. Then, for CLPØ to work, it suffices that the prover (1) knows $[\boldsymbol{C}(\boldsymbol{\chi})]_1$, and (2) is able to compute $[\boldsymbol{\delta}]_2$; for this, it suffices to compute $[\mathbf{we}]_2$, where $\mathbf{w}$ is such that $h(\boldsymbol{X}) = T(\boldsymbol{X})\mathbf{w}$ (this follows from CLPØ's construction).

In the case of $\mathbf{\Pi}^*$, (1) means that the prover must be able to compute $[\mathsf{q}, \mathbf{Z}_{\mathcal{S}}(\sigma), \mathsf{s}]_1$ (and thus χ, but not σ, must be available as an integer, and one must include to the CRS information needed to recompute $[\mathbf{Z}_{\mathcal{S}}(\sigma)]_1$), and (2) means that $[\sigma\mathsf{e}, \mathsf{e}]_2$ must be given as part of the CRS.

We leave the grand generalization of this observation for future work.

References

AFG+16. Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. *Journal of Cryptology*, 29(2):363–421, April 2016.

ALSZ20. Behzad Abdolmaleki, Helger Lipmaa, Janno Siim, and Michal Zajac. On QA-NIZK in the BPK model. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 590–620. Springer, Heidelberg, May 2020.

AN11. Tolga Acar and Lan Nguyen. Revocation for delegatable anonymous credentials. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 423–440. Springer, Heidelberg, March 2011.

ATSM09. Man Ho Au, Patrick P. Tsang, Willy Susilo, and Yi Mu. Dynamic universal accumulators for DDH groups and their application to attribute-based anonymous credential systems. In Marc Fischlin, editor, *CT-RSA 2009*, volume 5473 of *LNCS*, pages 295–308. Springer, Heidelberg, April 2009.

BBLP21. Olivier Blazy, Xavier Bultel, Pascal Lafourcade, and Octavio Perez-Kempner. Generic Plaintext Equality and Inequality Proofs. In Nikita Borisov and Claudia Diaz, editors, *FC 2021 (1)*, volume 12674 of *LNCS*, pages 415–435, Virtual, March 1–15, 2021. Springer, Cham.

BCF+21. Daniel Benarroch, Matteo Campanelli, Dario Fiore, Kobi Gurkan, and Dimitris Kolonelos. Zero-Knowledge Proofs for Set Membership: Efficient, Succinct, Modular. In Nikita Borisov and Claudia Diaz, editors, *FC 2021 (1)*, volume 12674 of *LNCS*, pages 393–414, Virtual, March 1–15, 2021. Springer, Cham.

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

BCV15. Olivier Blazy, Céline Chevalier, and Damien Vergnaud. Non-interactive zero-knowledge proofs of non-membership. In Kaisa Nyberg, editor, *CT-RSA 2015*, volume 9048 of *LNCS*, pages 145–164. Springer, Heidelberg, April 2015.

BdM93. Josh Benaloh and Michael de Mare. One-Way Accumulators: A Decentralized Alternative to Digital Signatures. In Tor Helleseth, editor, *EUROCRYPT 1993*, volume 765 of *LNCS*, pages 274–285, Lofthus, Norway, May 23–27, 1993. Springer, Heidelberg, 1994.

BDSS16. Olivier Blazy, David Derler, Daniel Slamanig, and Raphael Spreitzer. Non-interactive plaintext (in-)equality proofs and group signatures with verifiable controllable linkability. In Kazue Sako, editor, *CT-RSA 2016*, volume 9610 of *LNCS*, pages 127–143. Springer, Heidelberg, February / March 2016.

BLL00. Ahto Buldas, Peeter Laud, and Helger Lipmaa. Accountable certificate management using undeniable attestations. In Dimitris Gritzalis, Sushil Jajodia, and Pierangela Samarati, editors, *ACM CCS 2000*, pages 9–17. ACM Press, November 2000.

BLL02. Ahto Buldas, Peeter Laud, and Helger Lipmaa. Eliminating Counterevidence with Applications to Accountable Certificate Management. *Journal of Computer Security*, 10(3):273–296, 2002.

BP97. Niko Barić and Birgit Pfitzmann. Collision-Free Accumulators and Fail-Stop Signature Schemes without Trees. In Walter Fumy, editor, *EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 480–494, Konstanz, Germany, 11–15 May 1997. Springer, Heidelberg.

CCs08. Jan Camenisch, Rafik Chaabouni, and abhi shelat. Efficient protocols for set membership and range proofs. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 234–252. Springer, Heidelberg, December 2008.

CH20. Geoffroy Couteau and Dominik Hartmann. Shorter non-interactive zero-knowledge arguments and ZAPs for algebraic languages. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 768–798. Springer, Heidelberg, August 2020.

CLPØ21. Geoffroy Couteau, Helger Lipmaa, Roberto Parisella, and Arne Tobias Ødegaard. Efficient NIZKs for algebraic sets. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part III*, volume 13092 of *LNCS*, pages 128–158. Springer, Heidelberg, December 2021.

DGP+19. Vanesa Daza, Alonso González, Zaira Pindado, Carla Ràfols, and Javier Silva. Shorter quadratic QA-NIZK proofs. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part I*, volume 11442 of *LNCS*, pages 314–343. Springer, Heidelberg, April 2019.

DT08. Ivan Damgård and Nikos Triandopoulos. Supporting non-membership proofs with bilinear-map accumulators. Cryptology ePrint Archive, Report 2008/538, 2008. `https://eprint.iacr.org/2008/538`.

FKL18. Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Heidelberg, August 2018.

GKM+18. Jens Groth, Markulf Kohlweiss, Mary Maller, Sarah Meiklejohn, and Ian Miers. Updatable and universal common reference strings with applications to zk-SNARKs. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 698–728. Springer, Heidelberg, August 2018.

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

IK00. Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *41st FOCS*, pages 294–304. IEEE Computer Society Press, November 2000.

IK02. Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan Eidenbenz, and Ricardo Conejo, editors, *ICALP 2002*, volume 2380 of *LNCS*, pages 244–256. Springer, Heidelberg, July 2002.

Lip12. Helger Lipmaa. Secure accumulators from euclidean rings without trusted setup. In Feng Bao, Pierangela Samarati, and Jianying Zhou, editors, *ACNS 12*, volume 7341 of *LNCS*, pages 224–240. Springer, Heidelberg, June 2012.

LLX07. Jiangtao Li, Ninghui Li, and Rui Xue. Universal accumulators with efficient nonmembership proofs. In Jonathan Katz and Moti Yung, editors, *ACNS 07*, volume 4521 of *LNCS*, pages 253–269. Springer, Heidelberg, June 2007.

LSZ22. Helger Lipmaa, Janno Siim, and Michał Zając. Counting Vampires: From Univariate Sumcheck to Updatable ZK-SNARK. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022*, volume ? of *LNCS*, pages ?–?, Taipei, Taiwan, December 5–9, 2022. Springer, Cham. Accepted.

Ngu05. Lan Nguyen. Accumulators from bilinear pairings and applications. In Alfred Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 275–292. Springer, Heidelberg, February 2005.

A Some Missing Proofs

A.1 Proof of Theorem 2

Proof. Let $\mathcal{A}_{\mathrm{alg}}$ be an algebraic $\mathsf{DETACNM}$ adversary. Assume that $\mathcal{A}_{\mathrm{alg}}(\mathtt{crs})$ outputs $\psi = (\mathcal{S}, [\chi, \mathsf{q}, \mathsf{s}, \boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2)$, such that V accepts with a non-negligible probability. Since $\mathcal{A}_{\mathrm{alg}}$ is algebraic, with every group element $G \in \mathbb{G}_\iota$, $\mathcal{A}_{\mathrm{alg}}$ also outputs a vector $\boldsymbol{a}$ explaining how G is constructed from the elements of $\mathtt{crs}$ that belong to $\mathbb{G}_i$. Next, we will make this more precise.

Let $\boldsymbol{X} = (\Sigma, \mathsf{T}, \mathsf{E})$ and $\boldsymbol{x} = (\sigma, \tau, \mathsf{e})$. E.g., T is the indeterminate corresponding to the trapdoor τ. We express each output of the $\mathsf{DETACNM}$ adversary $\mathcal{A}_{\mathrm{alg}}$ as a polynomial evaluation, with say $[\chi]_1 = [\chi(\boldsymbol{x})]_1$. The relevant polynomials are

$$\begin{aligned}
\chi(\boldsymbol{X}) &= \chi_1(\Sigma)\mathsf{T} + \chi_2 \ , & \mathsf{q}(\boldsymbol{X}) &= \mathsf{q}_1(\Sigma)\mathsf{T} + \mathsf{q}_2 \ , \\
\mathsf{s}(\boldsymbol{X}) &= \mathsf{s}_1(\Sigma)\mathsf{T} + \mathsf{s}_2 \ , & \gamma_1(\boldsymbol{X}) &= \gamma_{11}(\Sigma)\mathsf{T} + \gamma_{12} \ , \\
\gamma_2(\boldsymbol{X}) &= \gamma_{21}(\Sigma)\mathsf{T} + \gamma_{22} \ , & \gamma_3(\boldsymbol{X}) &= \gamma_{31}(\Sigma)\mathsf{T} + \gamma_{32} \ , \\
\delta_1(\boldsymbol{X}) &= \delta_{11} + \delta_{12}\mathsf{E} + \delta_{13}\Sigma\mathsf{E} + \delta_{14}\mathsf{T}\mathsf{E} \ , & \delta_2(\boldsymbol{X}) &= \delta_{21} + \delta_{22}\mathsf{E} + \delta_{23}\Sigma\mathsf{E} + \delta_{24}\mathsf{T}\mathsf{E} \ ,
\end{aligned}$$

where each polynomial (like q_1) on the RHS is of degree $\leq N$. That is, the algebraic adversary $\mathcal{A}_{\mathrm{alg}}$ also outputs coefficients of all above polynomials.

The DETACNM verifier's checks guarantee that $V_1(\sigma,\tau,\mathsf{e}) = V_2(\sigma,\tau,\mathsf{e}) = 0$. Moreover, if $R(\boldsymbol{X}) \neq 0$ then $V_3(\sigma,\tau,\mathsf{e}) = V_4(\sigma,\tau,\mathsf{e}) = 0$. Here,

$$\begin{aligned} V_1(\boldsymbol{X}) &= ((\Sigma - \chi(\boldsymbol{X}))\,\mathsf{E} - \delta_1(\boldsymbol{X}))\cdot\mathsf{T} - \gamma_1(\boldsymbol{X}) \ , \\ V_2(\boldsymbol{X}) &= -\mathbf{Z}_{\mathcal{S}}(\Sigma)\mathsf{TE} + \mathsf{q}(\boldsymbol{X})\delta_1(\boldsymbol{X}) - \delta_2(\boldsymbol{X})\mathsf{T} - \gamma_2(\boldsymbol{X}) \ , \\ V_3(\boldsymbol{X}) &= -\mathsf{E} + \mathsf{s}(\boldsymbol{X})\delta_2(\boldsymbol{X}) - \gamma_3(\boldsymbol{X}) \ . \end{aligned}$$

Consider separately the cases (1) $V_1 = V_2 = V_3 = 0$ as polynomials, and (2) either $V_1 \neq 0$ or $V_2 \neq 0$ or $V_3 \neq 0$.

Case 1. Assume $V_1 = V_2 = V_3 = 0$ as a polynomial. Think of the polynomials as members of $\mathcal{R}[\mathsf{T},\mathsf{E}]$, where $\mathcal{R} = \mathbb{Z}_p[\Sigma]$. We now enlist the non-zero coefficients of all monomials $\mathsf{T}^i\mathsf{E}^j$ of all polynomials, highlighting the coefficients that are actually needed in this proof (we give other coefficients for completeness' sake):

(i,j) V_1
$(2,1)$ $-\delta_{14} - \chi_1(\Sigma)$
$(1,1)$ $-\delta_{12} + (1-\delta_{13})\Sigma - \chi_2$
$(1,0)$ $-\delta_{11} - \gamma_{11}(\Sigma)$
$(0,0)$ $-\gamma_{12}$

(i,j) V_3
$(2,1)$ $\delta_{24}\mathsf{s}_1(\Sigma)$
$(1,1)$ $\delta_{22}\mathsf{s}_1(\Sigma) + \delta_{23}\mathsf{s}_1(\Sigma)\Sigma + \mathsf{s}_2\delta_{24}$
$(1,0)$ $\delta_{21}\mathsf{s}_1(\Sigma) - \gamma_{31}(\Sigma)$
$(0,1)$ $\mathsf{s}_2\delta_{23}\Sigma + \mathsf{s}_2\delta_{22} - 1$
$(0,0)$ $\mathsf{s}_2\delta_{21} - \gamma_{32}$

(i,j) V_2
$(2,1)$ $\delta_{14}\mathsf{q}_1(\Sigma) - \delta_{24}$
$(1,1)$ $\delta_{14}\mathsf{q}_2 + \delta_{12}\mathsf{q}_1(\Sigma) + (\delta_{13}\mathsf{q}_1(\Sigma) - \delta_{23})\Sigma - \mathbf{Z}_{\mathcal{S}}(\Sigma) - \delta_{22}$
$(1,0)$ $\delta_{11}\mathsf{q}_1(\Sigma) - \gamma_{21}(\Sigma) - \delta_{21}$,
$(0,1)$ $\delta_{12}\mathsf{q}_2 + \delta_{13}\mathsf{q}_2\Sigma$
$(0,0)$ $\delta_{11}\mathsf{q}_2 - \gamma_{22}$

For example, the coefficient of $\mathsf{T}^2\mathsf{E}$ in V_1 is $-\delta_{14} - \chi_1(\Sigma)$. Since $V_i = 0$ as a polynomial, the coefficient of any monomial $\mathsf{T}^j\mathsf{E}^k$ in any V_i is also 0.

From the coefficient of $\mathsf{T}^2\mathsf{E}$ of V_1, we get $\chi_1(\Sigma) = -\delta_{14}$. From the coefficient of TE of V_1, after separating the coefficients of Σ^i, we get $\delta_{13} = 1$ and $\delta_{12} = -\chi_2$. Consider the coefficients of V_3:

- E: separating the coefficients of Σ, $\mathsf{s}_2\delta_{23} = 0$ and $\mathsf{s}_2\delta_{22} = 1$. Hence $\mathsf{s}_2 \neq 0$, and thus $\delta_{23} = 0$. Moreover, $\delta_{22} = 1/\mathsf{s}_2$.
- TE: $\mathsf{s}_1(\Sigma)/\mathsf{s}_2 + \mathsf{s}_2\delta_{24} = 0$ and thus $\mathsf{s}_1(\Sigma) = \mathsf{s}_2^2\delta_{24}$.
- $\mathsf{T}^2\mathsf{E}$: $\mathsf{s}_2^2\delta_{24}^2 = 0$. Since $\mathsf{s}_2 \neq 0$, $\delta_{24} = 0$.

  Going back to the coefficient of TE, we get $\mathsf{s}_1(\Sigma) = 0$.

Consider the coefficients of V_2:

- TE: $\mathbf{Z}_{\mathcal{S}}(\Sigma) - \delta_{14}\mathsf{q}_2 + 1/\mathsf{s}_2 = (\Sigma - \chi_2)\mathsf{q}_1(\Sigma)$.
  Since $\mathbf{Z}_{\mathcal{S}}(\Sigma)$ is non-constant, $\mathsf{q}_1(\Sigma) \neq 0$.
- $\mathsf{T}^2\mathsf{E}$: $\delta_{14}\mathsf{q}_1(\Sigma) = 0$. Since $\mathsf{q}_1(\Sigma) \neq 0$, we get $\delta_{14} = 0$.

Hence, the coefficient of TE of V_2 gives $\mathbf{Z}_{\mathcal{S}}(\Sigma) + 1/\mathsf{s}_2 = (\Sigma - \chi_2)\mathsf{q}_1(\Sigma)$. Thus, $(\Sigma - \chi_2) \mid \mathbf{Z}_{\mathcal{S}}(\Sigma) + 1/\mathsf{s}_2$. Since $\chi(\boldsymbol{X}) = \chi_2$, $\mathbf{Z}_{\mathcal{S}}(\chi_2) = -1/\mathsf{s}_2 \neq 0$.

Case 2. The case $V_i \neq 0$ for some i can be handled in a standard way. Assume for example that $V_2 \neq 0$. We construct a PDL reduction

$\mathcal{B}(\{[\sigma^i]_1\}_{i=0}^{N+1}, \{[\sigma^i]_1\}_{i=0}^{2})$. $\mathcal{B}$ samples $\alpha_1, \alpha_2, \beta_1, \beta_2 \leftarrow_\$ \mathbb{Z}_p$ and sets implicitly $\tau \leftarrow \alpha_1\sigma + \beta_1$ and $\mathsf{e} \leftarrow \alpha_2\sigma + \beta_2$. Then, $\mathcal{B}$ creates $\mathtt{crs}$ for an $\mathsf{DETACNM}$ adversary $\mathcal{A}_{\mathrm{alg}}$, and calls $\mathcal{A}_{\mathrm{alg}}$ with $\mathtt{crs}$. After obtaining π, together with the coefficients of the polynomials like $\chi(\Sigma)$, from $\mathsf{Ext}_{\mathcal{A}_{\mathrm{alg}}}$, $\mathcal{B}$ reconstructs the coefficients of the degree-$\leq (N+2)$ polynomial V_2 (which is now univariate since τ and e are affine maps of σ). We know $V_2 \neq 0$ but $V_2(\sigma) = 0$. $\mathcal{B}$ factorizes V_2 and finds up to $N+2$ roots x_i of V_2. $\mathcal{B}$ tests which one of them is equal to σ, and returns σ. □

A.2 Proof of Theorem 3

Proof. **Perfect completeness.** We consider separately membership and non-membership arguments.

Membership Argument. Clearly, $b_1 = \mathsf{true}$ iff $\mathsf{Enc}_{\mathsf{pk}}([\sigma]_1; 0)\tau\mathsf{e} - [\mathbf{ct}_\chi]_1\tau\mathsf{e} - \mathsf{Enc}_{\mathsf{pk}}([\tau]_1; 0)\delta \stackrel{?}{=} [\mathsf{ct}_{\gamma 1}]_1 + z_1 \cdot \mathsf{pk} \iff \mathsf{Enc}\left((\sigma-\chi)\tau\mathsf{e} - \tau\delta; -\varrho_\chi\tau\mathsf{e}\right) \stackrel{?}{=} \mathsf{Enc}\left(\gamma_1; \varrho_{\gamma 1}\right) + \mathsf{Enc}_{\mathsf{pk}}(0; z_1)$. Clearly, $(\sigma-\chi)\tau\mathsf{e} - \tau\delta = (\sigma-\chi)\tau\mathsf{e} - \tau((\sigma-\chi)\mathsf{e} - \varrho_\delta) = \varrho_\delta\tau = \gamma_1\tau$, and thus the ciphertext part is correct. On the other hand, randomizers are correct by definition.

Similarly, $b_2 = \mathsf{true}$ iff $-\mathsf{Enc}_{\mathsf{pk}}(\mathbf{Z}_{\mathcal{S}}(\sigma); 0)\tau\mathsf{e} + [\mathsf{ct}_\mathsf{q}]_1\delta \stackrel{?}{=} [\mathsf{ct}_{\gamma 2}]_1 + z_2 \cdot \mathsf{pk} \iff \mathsf{Enc}\left(-\mathbf{Z}_{\mathcal{S}}(\sigma)\tau\mathsf{e} + \mathsf{q}\delta; \varrho_\mathsf{q}\delta\right) \stackrel{?}{=} \mathsf{Enc}\left(\gamma_2; \varrho_{\gamma 2}\right) + \mathsf{Enc}_{\mathsf{pk}}(0; z_2)$. Consider first the ciphertexts. Clearly, $-\mathbf{Z}_{\mathcal{S}}(\sigma)\tau\mathsf{e} + \mathsf{q}\delta = -\mathbf{Z}_{\mathcal{S}}(\sigma)\tau\mathsf{e} + \mathsf{q}\cdot((\sigma-\chi)\mathsf{e} - \varrho_\delta) = -\varrho_\delta\mathsf{q} = \gamma_2$. On the other hand, randomizers are correct by definition.

Non-Membership Argument. Clearly, $\bar{b}_1 = \mathsf{true}$ iff $\mathsf{Enc}_{\mathsf{pk}}([\sigma]_1; 0)\tau\mathsf{e} - [\mathbf{ct}_\chi]_1\tau\mathsf{e} - \mathsf{Enc}_{\mathsf{pk}}([\tau]_1; 0)\delta_1 \stackrel{?}{=} [\mathsf{ct}_{\gamma 1}]_1 + z_1 \cdot \mathsf{pk} \iff \mathsf{Enc}_{\mathsf{pk}}\left((\sigma-\chi)\tau\mathsf{e} - \tau\delta_1; \varrho_\chi\tau\mathsf{e}\right) \stackrel{?}{=} \mathsf{Enc}_{\mathsf{pk}}(\gamma_1; \varrho_{\gamma 1}) + \mathsf{Enc}_{\mathsf{pk}}(0; z_1)$. Consider first the ciphertexts. Clearly, $(\sigma-\chi)\tau\mathsf{e} - \tau\delta_1 = y_1\tau = \gamma_1\tau$. On the other hand, randomizers are correct by definition.

Similarly, $\bar{b}_2 = \mathsf{true}$ iff $-\mathsf{Enc}_{\mathsf{pk}}(\mathbf{Z}_{\mathcal{S}}(\sigma); 0)\tau\mathsf{e} + [\mathsf{ct}_\mathsf{q}]_1\delta_1 - \mathsf{Enc}_{\mathsf{pk}}(1; 0)\tau\delta_2 \stackrel{?}{=} [\mathsf{ct}_{\gamma 2}]_1 + z_2 \cdot \mathsf{pk} \iff \mathsf{Enc}_{\mathsf{pk}}\left(-\mathbf{Z}_{\mathcal{S}}(\sigma)\tau\mathsf{e} + \mathsf{q}\delta_1 - \tau\delta_2; \varrho_\mathsf{q}\delta_1\right) \stackrel{?}{=} \mathsf{Enc}_{\mathsf{pk}}(\gamma_2; \varrho_{\gamma 2}) + \mathsf{Enc}_{\mathsf{pk}}(0; z_2)$. Consider first the ciphertexts. Clearly, $-\mathbf{Z}_{\mathcal{S}}(\sigma)\tau\mathsf{e} + \mathsf{q}\delta_1 - \tau\delta_2 = -\mathbf{Z}_{\mathcal{S}}(\sigma)\tau\mathsf{e} + \mathsf{q}\cdot((\sigma-\chi)\mathsf{e} - y_1) - \tau(1/\mathsf{s}\cdot\mathsf{e} - y_2) = -y_1\mathsf{q} + y_2\tau = \gamma_2$. On the other hand, randomizers are correct by definition.

Finally, $\bar{b}_3 = \mathsf{true}$ iff $-\mathsf{Enc}(1; 0)\mathsf{e} + \mathsf{ct}_\mathsf{s}\delta_2 \stackrel{?}{=} [\mathsf{ct}_{\gamma 3}]_1 + z_3 \cdot \mathsf{pk} \iff \mathsf{Enc}_{\mathsf{pk}}\left(-\mathsf{e} + \mathsf{s}\delta_2; \varrho_\mathsf{s}\delta_2\right) \stackrel{?}{=} \mathsf{Enc}_{\mathsf{pk}}(\gamma_3; \varrho_{\gamma 3}) + \mathsf{Enc}_{\mathsf{pk}}(0; z_3)$. Consider first the ciphertexts. Clearly, $-\mathsf{e} + \mathsf{s}\delta_2 = -\mathsf{e} + \mathsf{s}(1/\mathsf{s}\cdot\mathsf{e} - y_2) = -y_2\mathsf{s} = \gamma_3$. On the other hand, randomizers are correct by definition.

Computational zero-knowledge: First, consider the membership argument. Fix any λ, and let $(\mathtt{crs}, \mathtt{td}) \in \mathrm{Supp}(\mathsf{K}_{\mathtt{crs}}(1^\lambda))$. Let $\mathtt{lpar} = (\mathsf{pk}, \mathcal{S})$ and $(\mathbb{x}, \mathbb{w}) \in \mathcal{R}_{\mathtt{lpar}}$. To show zero-knowledge we first define an hybrid simulator Sim_H. The hybrid Sim_H receives as additional input an Elgamal ciphertext $[\mathsf{ct}_\mathsf{q}]_1$, that is an encryption of $[\mathsf{q}]_1$ such that $\mathsf{q}(\sigma - \chi) = \mathbf{Z}_{\mathcal{S}}(\sigma)$, where $[\chi]_1 = \mathsf{Dec}_{\mathsf{sk}}([\mathsf{ct}_\chi]_1)$. Then Sim_H computes its output as the simulator in Fig. 9, except that it computes $[\mathsf{ct}_\mathsf{q}]_1$ as an encryption of $[\mathsf{q}]_1$ and not of 0. The output of Sim_H is perfectly close to the output of the honest prover. The proof of the last statement is the same as the perfect zero-knowledge proof in [CLPØ21]. For completeness, we state a proof for this concrete case. In the honest prover's algorithm, since $\boldsymbol{\varrho}_\gamma$ is uniformly random, then also $\boldsymbol{z}$ is uniformly random. As in

Fact 1, $\boldsymbol{\delta}$ output by an honest prover is uniformly random. On the other hand, Sim_H also samples uniformly random $\boldsymbol{\delta}$ and $\boldsymbol{z}$. Finally, in both the prover's and simulator's case, one can verify manually that $[\mathbf{ct}_\gamma]_1$ is the unique value that makes the verifier accept the argument π. Then we show that the output of the real simulator (see Fig. 9) is computationally close to the output of Sim_H. This follows directly from Elgamal IND-CPA security (which holds under the XDH assumption).

In the non-membership argument, zero-knowledge holds analogously. □