

Set (Non-)Membership NIZKs from Determinantal Accumulators^{*}

Thursday 10th August, 2023

Helger Lipmaa and Roberto Parisella

Simula UiB, Bergen, Norway

Abstract. We construct a falsifiable set (non-)membership NIZK Π^* that is considerably more efficient than known falsifiable set (non-)membership NIZKs. It also has a universal CRS. Π^* 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 Groth-Sahai NIZK. 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 safelist (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* block-listed. 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, BP97]. A universal accumulator [BLL00, BLL02, LLX07, DT08] 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

^{*} Second eprint, the full version corresponding to the publication [LP23]. The main update is an explicit description of the verifier batching technique for CLPØ in general and accumulators in particular. We also added a discussion on falsifiable versus non-falsifiable assumptions.

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.

On Models and Assumptions. While the random oracle model, other idealized models (like the algebraic group model, AGM [FKL18]), and non-falsifiable knowledge assumptions have been used successfully in many practical schemes, they have also drawn severe criticism [Nao03,GK16]. See Appendix A.1 for a longer discussion. Because of this, there has been an enormous focus in the literature on constructing efficient non-falsifiable NIZKs. The most famous such NIZK is GS (Groth-Sahai, [GS08]) which has been used since then in many different applications, including (non-)membership NIZKs.

On the other hand, there is less of a qualitative distinction between different *falsifiable* assumptions — as long as such assumptions have reasonably tight security reductions in the AGM. All previous “falsifiable” accumulators are based on relatively strong falsifiable q -type assumptions. *Hence, we aim to construct maximally efficient (non-)membership NIZKs under (possibly novel) falsifiable computational assumptions.* There are many non-falsifiable or random-oracle-based set (non-)membership NIZKs [CCs08,BCF⁺21,VB22]; we do not compete with them, and thus we omit almost any discussion.

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-universal¹ (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 explicitly disallow the construction of set non-membership arguments.

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 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 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 Nguyen accumulator [Ngu05] allows for that.) Thus, their CRS is universal. Since there is no need to add all accumulator arguments to the

¹ We follow the literature by using “universal” both in *universal accumulators* (have a non-membership argument) and *universal CRS* (does not depend on the language).

CRS, one can hope to construct efficient accumulator-based set *non-membership* NIZKs.

Next, we will summarize the published falsifiable set-membership NIZKs. In all cases $\mathcal{S} \subset \mathbb{Z}_p$ and hence $\chi \in \mathbb{Z}_p$. Since the cited papers did not write down all efficiency numbers, our efficiency comparison (see Table 1) is not 100% 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 whose signature the prover knows, 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], [DGP⁺19] modifies it slightly. Using signature schemes means that the CRS of BCKL and DGPRS-GS is non-universal. Daza et al. [DGP⁺19] also propose a succinct set membership QA-NIZK (DGPRS-QA). However, their verifier’s computation is $O_\lambda(|\mathcal{S}|)$; thus, it is unsuitable for our applications. In particular, all solutions in Table 1 have verifier’s computation $O_\lambda(1)$. Hence, we omit DGPRS-QA in the comparison.

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 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 rely on new (though falsifiable) q -type security assumptions. The intuition 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 (say², $[\chi]_1$) of χ . BCKL, AN, and DGPRS-GS all modified underlying primitives to stay secure against adversaries that output $[\chi]_1$. In each construction, this resulted in a new but falsifiable assumption. Moreover, such a modification often introduces a noticeable loss of efficiency. We describe all assumptions in Appendix A.2 for completeness.

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 executes 34 multi-scalar-multiplications ([BCKL08] does not give separately the number of scalar-multiplications in \mathbb{G}_1 and \mathbb{G}_2) and the verifier 68 pairings. Neither AN [AN11] nor Daza et al. [DGP⁺19] give any efficiency numbers. The corresponding entries (marked with an asterisk) in Table 1 are based on our estimations.

² We use the standard additive bracket notation for pairing-based setting. For example, $[\chi]_1 = \chi[1]_1$, where $[1]_1$ is a generator of \mathbb{G}_1 .

Table 1. Comparison of known fully succinct falsifiable set (non-)membership arguments for univariate sets of size $|\mathcal{S}| \leq q$. Here, \mathbf{g}_i denotes the bit-length of an element of \mathbb{G}_i , \mathbf{m}_i denotes the cost of a scalar multiplication in \mathbb{G}_i , \mathbf{m} denotes the cost of a scalar multiplication in either \mathbb{G}_1 or \mathbb{G}_2 , and \mathbf{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 computation, i.e., assuming precomputation. We only mention non-standard assumptions; this excludes say SXDH.

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

Recent NIZKs of Couteau et al. Most 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 $[e]_2$ to the CRS and then modifies the computation of z and the verifier’s algorithm to work on $[e]_2$ 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 but natural 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.

[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

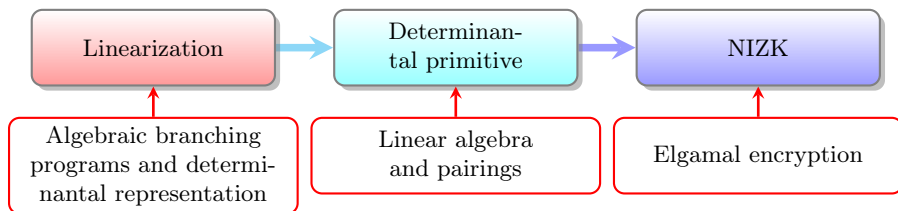


Fig. 1. Our general blueprint for constructing efficient falsifiable NIZKs.

\mathbb{G}_1 . Because of this, the encrypted witness is often three times shorter in $\text{CLP}\emptyset$ than in Groth-Sahai; see [CH20,CLP \emptyset 21] for examples.

Our first main question is whether one can construct $\text{CLP}\emptyset$ -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 $\text{CLP}\emptyset$ NIZKs.

Our Contributions. Since a universal accumulator is a non-zk (non-)membership non-interactive argument system, 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 $\text{CLP}\emptyset$ NIZK. We define a *determinantal accumulator* as a universal accumulator with a structure that supports efficient ZK compilation to $\text{CLP}\emptyset$. 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 $\text{CLP}\emptyset$'s technique of using algebraic branching programs. Based on AC^* , we then construct Π^* , a commit-and-prove, updatable set (non-)membership NIZK with a universal CRS.

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 $\text{CLP}\emptyset$ -inspired ZK compiler to achieve zero knowledge. See Fig. 1 for a high-level diagram of the new approach.

We develop a general efficient technique that allows one to construct non-membership NIZKs for a large class of languages where $\text{CLP}\emptyset$ only supported membership NIZKs. We use this technique for AC^* and Π^* , but it potentially has many more applications. We show that $\text{CLP}\emptyset$, in general, is amenable to a batching technique that allows decreasing the verifier's computation significantly.

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. Our work demonstrates that the CH/CLP \emptyset framework gives concretely better results than the seminal Groth-Sahai framework in this case. Importantly, this is the only known falsifiable framework that improves on the Groth-Sahai. Because of that, we argue that it is important to study different aspects of the CH/CLP \emptyset framework.

Finally, in Section 7, we discuss using CLP \emptyset to handle group elements.

Determinantal Accumulators. We assume the standard pairing-based setting (see Section 2). We follow [CLP \emptyset 21], but we reinterpret their constructions. First, the verifier has access to input (namely, χ), auxiliary input (e.g., commitment to \mathcal{S}), and output (accumulator’s argument) only in \mathbb{G}_1 , i.e., not as integers. The availability of all private values in \mathbb{G}_1 enables us to use an efficient ZK compiler, where only \mathbb{G}_1 elements will be encrypted. (In many pairing-based settings, \mathbb{G}_2 elements 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, in prior falsifiable pairing-based accumulators, the verification equations were pairing-product equations. The determinantal accumulator’s verifier checks that the determinants (potentially high-degree polynomials) of some matrices, whose entries are affine maps, are zero. This can be seen as a linearization of a polynomial $F(\mathbf{X})$ by using affine maps. More precisely, the determinantal accumulator’s verifier accepts iff $\det \mathbf{C}_i(\chi) = 0$ for DRs $\mathbf{C}_i(\mathbf{X})$ of some well-chosen polynomials $F_i(\mathbf{X})$. Here, a DR (determinantal representation) $\mathbf{C}(\mathbf{X})$ of $F(\mathbf{X})$ is a matrix, where each entry of $\mathbf{C}(\mathbf{X})$ is an affine map of \mathbf{X} , and the determinant of $\mathbf{C}(\mathbf{X})$ is $F(\mathbf{X})$.

Since we only need to test that the determinant is zero, we follow the underlying ideas of [CH20,CLP \emptyset 21] to make the accumulator efficiently and publicly verifiable. Namely, we use the undergraduate linear algebraic fact that $\det \mathbf{C}(\mathbf{X}) = 0$ iff there exists a non-zero vector \mathbf{d} , such that $\mathbf{C}(\mathbf{X}) \cdot \mathbf{d} = \mathbf{0}$. To simplify the construction of accumulators and NIZKs, we follow [CLP \emptyset 21] and require that the first coordinate of \mathbf{d} is non-zero. Moreover, to achieve both soundness and zero-knowledge in the case of NIZKs, we define $\mathbf{d} = \begin{pmatrix} \mathbf{e} \\ \delta \end{pmatrix}$ for a new trapdoor $\mathbf{e} \leftarrow_s \mathbb{Z}_p$. (For such \mathbf{e} to exist, the matrices $\mathbf{C}(\mathbf{X})$ need to satisfy an additional requirement, see [CLP \emptyset 21].) We mask δ additively with well-chosen randomness to achieve zero knowledge. To balance the randomness, we introduce an additional (\mathbf{e} -independent) vector γ and prove that $\mathbf{C}(\mathbf{X}) \cdot \begin{pmatrix} \mathbf{e} \\ \delta \end{pmatrix} = \gamma$.

Thus, in a determinantal accumulator, the prover outputs $[\chi]_1$ (this includes $[\chi]_1$, the candidate element for $\chi \in \mathcal{S}$) and hints $[\delta]_2$ and $[\gamma]_1$. The verifier checks that $[\mathbf{C}(\chi)]_1 \bullet [\delta]_2 = [\gamma]_1 \bullet [1]_2$. (Here, χ is the vector of concrete values of the indeterminates \mathbf{X} .) Assuming $\mathbf{C}(\mathbf{X})$ has constant size, the verification is constant time. Defining determinantal accumulators is an important contribution of this paper. In particular, one can take another primitive (for example, a

signature scheme) and define its determinantal variant. This may result in other efficient CLPØ-style NIZKs. We leave any generalizations to future work.

New Determinantal Accumulator AC*. AC* uses a DR $C(\mathbf{X})$ that is motivated by Nguyen’s accumulator [Ngu05]. Define

$$C_{\Sigma}(\mathbf{X}, \mathbf{Q}) := \begin{bmatrix} \Sigma - \mathbf{X} & -1 \\ -\mathbf{Z}_{\Sigma}(\Sigma) & \mathbf{Q} \end{bmatrix}_1$$

and

$$C_{\sigma}(\chi, \mathbf{q}) = \begin{bmatrix} \sigma - \chi & -1 \\ -\mathbf{Z}_{\Sigma}(\sigma) & \mathbf{q} \end{bmatrix}_1 .$$

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

In Nguyen’s accumulator, given the membership argument $[\mathbf{q}]_2 \in \mathbb{G}_2$, the verifier checks that $[\sigma - \chi]_1 \bullet [\mathbf{q}]_2 \bullet = [\mathbf{Z}_{\Sigma}(\sigma)]_1 \bullet [1]_2$. In all known Groth-Sahai based solutions, to verify that $\det C_{\sigma}(\chi, \mathbf{q}) = 0$, either the encryption of χ or \mathbf{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 $([\mathbf{q}, \gamma]_1, [\delta]_2)$, where $\gamma \in \mathbb{Z}_p^2$ and $\delta \in \mathbb{Z}_p$. (We will define γ and δ in Section 5.)

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

Previous works [BCKL08, AN11, DGP⁺19] solve this problem from scratch, each using a new tailor-made assumption. We approach it systematically. We define a new security property, F -collision-resistance. 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 make AC* F -collision-resistant by introducing another trapdoor τ . The goal of τ is to guarantee that if the verifier accepts, then χ and \mathbf{r} do not depend on σ . We also change AC*’s verification equations. Crucially, we do it without increasing communication complexity. 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 prove that DETACM and DETACNM

are implied by the standard q -type power discrete logarithm (PDL, [Lip12a]) assumption in AGM. We emphasize that DETACM and DETACNM are falsifiable. We use AGM only to argue that DETACM and DETACNM are reasonable and very plausible (falsifiable) assumptions. Importantly, both assumptions are Maurer-type [Zha22] and thus do not fall under the criticism of recent papers [Zha22,ZZK22]. Note that the most efficient structure-preserving signatures are proven secure, under tautological assumptions, validated in the generic group model or AGM. The main difference is that the collision resistance of accumulators is a weaker assumption than the unforgeability of signature schemes; in particular, the former is a non-interactive assumption while the latter is not.

General Non-Membership NIZK. As a result of independent importance, in Section 3, we develop a generic technique for constructing NIZKs for encryption of elements that are *not* a root of a given polynomial. 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 Π^* . We are now ready to describe an efficient commit-and-prove NIZK Π^* for showing that an Elgamal-encrypted χ belongs (or does not belong) to the set \mathcal{S} . Π^* 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 $[z]_2$ that balances off the randomizers used in such encryptions.

The NIZK verifier performs the accumulator verification on the ciphertexts, taking $[z]_2$ into account.

Π^* is computationally sound, assuming that AC^* is F -collision-resistant, i.e., under a falsifiable assumption. The reduction uses the Elgamal secret key to decrypt encrypted data and returns it with the hint $[\delta]_2$. We prove that Π^* 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 common case when \mathbb{G}_2 elements are twice longer than \mathbb{G}_1 , the communication of Π^* is $\approx 42\%$ of the communication of [AN11]. Most impressively, the verifier has to execute 7...10 times fewer pairings than in [AN11]. As a related contribution, we show that CLP \emptyset is (extremely) batching-friendly. Such an efficiency improvement over the best-known

Groth-Sahai-based solution is quite remarkable. In the case of prover’s computation, we have taken the standard approach and assumed that the accumulator argument ($[q]_1$ 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,Lip12b] 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.

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

2 Preliminaries

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 $F : \mathbb{F}_p^\nu \rightarrow \mathbb{F}_p$. Here, ϕ assigns to each edge in E a fixed affine function in input variables, and $F(\mathbf{X})$ 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 $\text{abp} = (V, E, s, t, \phi)$ computing $F : \mathbb{F}_p^\nu \rightarrow \mathbb{F}_p$, we can efficiently (and deterministically) compute a function $\text{IK}_F(\boldsymbol{\chi})$ 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) $\det \text{IK}_F(\boldsymbol{\chi}) = F(\boldsymbol{\chi})$, (2) each entry of $\text{IK}_F(\boldsymbol{\chi})$ is an affine map in a single variable χ_i , (3) $\text{IK}_F(\boldsymbol{\chi})$ 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 $\text{adj}(\mathbf{X}) - \mathbf{I}$. Here, $\text{adj}(\mathbf{X})$ is the adjacency matrix for abp with $\text{adj}(\mathbf{X})_{ij} = x$ iff $\phi(i \rightarrow j) = x$ and $\text{adj}(\mathbf{X})_{ij} = 0$ if there is no edge $i \rightarrow j$. For example, assuming $F(X) = X^2 - X$, one can define an ABP with

$$\text{adj}(X) = \begin{pmatrix} 0 & X & 0 \\ 0 & 0 & X-1 \\ 0 & 0 & 0 \end{pmatrix}$$

and

$$\mathbb{K}_F(X) = \begin{pmatrix} X & -1 \\ 0 & X-1 \end{pmatrix} .$$

Cryptography. A bilinear group generator $\text{Pgen}(1^\lambda)$ returns $\mathfrak{p} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, [1]_1, [1]_2)$, where $\mathbb{G}_1, \mathbb{G}_2$, and \mathbb{G}_T are three additive cyclic (thus, abelian) groups of prime order p , $\mathcal{P}_\iota = [1]_\iota$ is a generator of \mathbb{G}_ι for $\iota \in \{1, 2, T\}$ with $\mathcal{P}_T = [1]_T := \hat{e}([1]_1, [1]_2)$, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ 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 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 $\mathbf{A}\mathbf{B} = \mathbf{C}$ then $[\mathbf{A}]_1 \bullet [\mathbf{B}]_2 = [\mathbf{C}]_T$. We also define $[\mathbf{A}]_2 \bullet [\mathbf{B}]_1 := ([\mathbf{B}]_1^\top \bullet [\mathbf{A}]_2^\top)^\top = [\mathbf{A}\mathbf{B}]_T$.

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

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

$$\Pr \left[b' = b \left| \begin{array}{l} \mathfrak{p} \leftarrow \text{Pgen}(1^\lambda); \sigma, \tau, \zeta \leftarrow_s \mathbb{Z}_p; b \leftarrow_s \{0, 1\}; \\ b' \leftarrow \mathcal{A}([1], \sigma, \tau, \sigma\tau + b\zeta)_\iota \end{array} \right. \right] \approx_\lambda 0 .$$

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

$$\Pr \left[\begin{array}{l} \delta \in \mathbb{Z}_p^{(\ell-1) \times k} \wedge \gamma \in \mathbb{Z}_p^{\ell \times k} \wedge \\ \mathbf{C} \in \mathbb{Z}_p^{\ell \times \ell} \wedge (\gamma \| \mathbf{C}) \begin{pmatrix} \mathbf{D} \\ \delta \end{pmatrix} = \mathbf{0} \wedge \\ \text{rk}(\mathbf{C}) = \ell \end{array} \left| \begin{array}{l} \mathfrak{p} \leftarrow \text{Pgen}(1^\lambda), [\mathbf{D}]_\iota \leftarrow_s \mathcal{D}_k, \\ ([\gamma, \mathbf{C}]_{3-\iota}, [\delta]_\iota) \leftarrow \mathcal{A}(\mathfrak{p}, [\mathbf{D}]_\iota) \end{array} \right. \right] \approx_\lambda 0 .$$

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

Following [CH20, CLPØ21], we will be only concerned with the case $k = 1$ and $\mathcal{D}_k = \mathcal{L}_1$. Then, $(\gamma \| \mathbf{C}) \begin{pmatrix} \mathbf{D} \\ \delta \end{pmatrix} = \mathbf{0}$ iff, after changing the sign of γ , $\mathbf{C} \begin{pmatrix} \delta \\ \gamma \end{pmatrix} = \gamma$.

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}_{alg} be a PPT algorithm, run on initial inputs including description \mathfrak{p} with oracles or other parties and receive further inputs including obviously sampled group elements (which it cannot sample directly). Let $\mathbf{L} \in \mathbb{G}^n$ 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 $\mathbf{a} = (a_i)_{i=1}^n \in \mathbb{Z}_p^n$, such that $G = \sum_{i=1}^n a_i L_i = \langle \mathbf{a}, \mathbf{L} \rangle$.

AGM reductions are usually given to the following assumption. Let $d_1, d_2 \in \text{poly}(\lambda)$. The (d_1, d_2) -PDL (*Power Discrete Logarithm, [Sta08, Lip12a]*) assumption holds relative to Pgen, if for any PPT \mathcal{A} ,

$$\Pr [\sigma' = \sigma \mid \mathbf{p} \leftarrow \text{Pgen}(1^\lambda); \sigma \leftarrow_{\$} \mathbb{Z}_p; \sigma' \leftarrow \mathcal{A}(\mathbf{p}; [(\sigma^i)_{i=0}^{d_1}]_1, [(\sigma^i)_{i=0}^{d_2}]_2)] \approx_{\lambda} 0 .$$

Elgamal encryption. In Elgamal, the public key is $\text{pk} = [1 \parallel \text{sk}]_1$, and $\text{Enc}_{\text{pk}}(\chi; \varrho) \leftarrow (\varrho[1]_1 \parallel \chi[1]_1 + \varrho[\text{sk}]_1)$, where $\varrho \leftarrow_{\$} \mathbb{Z}_p$. We denote the encryption of $[\chi]_1$ by

$$\text{Enc}_{\text{pk}}([\chi]_1; \varrho) = (\varrho[1]_1 \parallel [\chi]_1 + \varrho[\text{sk}]_1) .$$

To decrypt, one computes

$$[\chi]_1 = \text{Dec}_{\text{sk}}([l]_1) \leftarrow -\text{sk}[c_1]_1 + [c_2]_1 ;$$

the result $[\chi]_1$ of the decryption is a group element and not an integer. Note that $\text{pk} = \text{Enc}_{\text{pk}}(0; 1)$ and $[0 \parallel \chi]_1 = \text{Enc}_{\text{pk}}(\chi; 0)$. As always, we denote $\text{Enc}_{\text{pk}}([\mathbf{a}]_1; \varrho) := (\text{Enc}_{\text{pk}}([a_i]_1; \varrho_i))_i$. Elgamal is IND-CPA secure under the XDH assumption.

2.1 Universal NIZK Arguments

Let $\{\mathcal{D}_{\mathbf{p}}\}_{\mathbf{p}}$ be a family of distributions, s.t. each $\mathbf{1}_{\mathbf{p}} \in \mathcal{D}_{\mathbf{p}}$ defines a language $\mathcal{L}_{\mathbf{1}_{\mathbf{p}}}$. A universal NIZK Π for $\{\mathcal{D}_{\mathbf{p}}\}_{\mathbf{p}}$ consists of six probabilistic algorithms:

Pgen(1^λ): generates public parameters \mathbf{p} that fix a distribution $\mathcal{D}_{\mathbf{p}}$.

Kgen(\mathbf{p}, q): generates a CRS crs and a trapdoor td . Here, q is a public size parameter (an upper bound of $|\mathcal{S}|$ in our case); we assume q is implicitly in the CRS. We omit q if the CRS does not depend on it. We assume that any group parameters are implicitly included in the CRS. We denote the sequence “ $\mathbf{p} \leftarrow \text{Pgen}(1^\lambda); (\text{crs}, \text{td}) \leftarrow \text{Kgen}(\mathbf{p}, q)$ ” by $(\mathbf{p}, \text{crs}, \text{td}) \leftarrow \text{Kgen}(1^\lambda, q)$.

Com($\text{crs}, \mathbf{1}_{\mathbf{p}}$): Given a CRS crs and a language description $\mathbf{1}_{\mathbf{p}} \in \mathcal{D}_{\mathbf{p}}$, outputs a specialized CRS $\text{crs}_{\mathbf{1}_{\mathbf{p}}}$. We assume that $\text{crs}_{\mathbf{1}_{\mathbf{p}}}$ implicitly contains $\mathbf{1}_{\mathbf{p}}$. **Com** is a deterministic algorithm that can be run by both the prover and the verifier. (Com is also known as CRS specialization algorithm, indexer, or derive.)

P($\text{crs}_{\mathbf{1}_{\mathbf{p}}}, \mathbf{x}, \mathbf{w}$): Given a specialized CRS $\text{crs}_{\mathbf{1}_{\mathbf{p}}}$ and a statement \mathbf{x} with witness \mathbf{w} , outputs an argument π for $\mathbf{x} \in \mathcal{L}_{\mathbf{1}_{\mathbf{p}}}$.

V($\text{crs}_{\mathbf{1}_{\mathbf{p}}}, \mathbf{x}, \pi$): Given a specialized CRS $\text{crs}_{\mathbf{1}_{\mathbf{p}}}$, a statement, and an argument, either accepts or rejects the argument.

Sim($\text{crs}_{\mathbf{1}_{\mathbf{p}}}, \text{td}, \mathbf{x}$): Given a specialized CRS $\text{crs}_{\mathbf{1}_{\mathbf{p}}}$, a trapdoor td , and a statement \mathbf{x} , outputs a simulated argument for $\mathbf{x} \in \mathcal{L}_{\mathbf{1}_{\mathbf{p}}}$.

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 $\text{crs}_{\mathbf{1}_{\mathbf{p}}}$.

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

$$\Pr \left[\mathbf{V}(\text{crs}_{\mathbf{1}_{\mathbf{p}}}, \mathbf{x}, \pi) = 1 \mid \begin{array}{l} (\mathbf{p}, \text{crs}, \text{td}) \leftarrow_{\$} \text{K}_{\text{crs}}(1^\lambda); \mathbf{1}_{\mathbf{p}} \in \text{Supp}(\mathcal{D}_{\mathbf{p}}); \\ \text{crs}_{\mathbf{1}_{\mathbf{p}}} \leftarrow \text{Com}(\text{crs}, \mathbf{1}_{\mathbf{p}}); \\ (\mathbf{x}, \mathbf{w}) \in \mathcal{R}_{\mathbf{1}_{\mathbf{p}}}; \pi \leftarrow_{\$} \text{P}(\text{crs}_{\mathbf{1}_{\mathbf{p}}}, \mathbf{x}, \mathbf{w}) \end{array} \right] = 1 .$$

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

$$\Pr \left[\begin{array}{l} \mathbf{V}(\mathbf{crs}_{1p}, \mathbf{x}, \pi) = 1 \wedge \\ \mathbf{x} \notin \mathcal{L}_{1p} \end{array} \middle| \begin{array}{l} (\mathbf{p}, \mathbf{crs}, \mathbf{td}) \leftarrow_s \mathbf{K}_{\mathbf{crs}}(1^\lambda); \mathbf{1p} \in \text{Supp}(\mathcal{D}_p); \\ \mathbf{crs}_{1p} \leftarrow \text{Com}(\mathbf{crs}, \mathbf{1p}); (\mathbf{x}, \pi) \leftarrow \mathcal{A}(\mathbf{crs}, \mathbf{1p}) \end{array} \right] \approx 0 .$$

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

Π is *commit-and-prove* if its input \mathbf{x} is a ciphertext, such that the argument convinces the verifier that $\text{det}_{\text{sk}}(\mathbf{x}) \in \mathcal{L}$ 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 2.2) 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 $([f(\mathbf{x}) : f \in \mathcal{T}_1]_1, [f(\mathbf{x}) : f \in \mathcal{T}_2]_2)$, where \mathbf{x} is a vector of trapdoors over \mathbb{Z}_p , and \mathcal{T}_i are sets of monomials. For example, $\mathbf{crs} = ([1, \tau, \sigma\tau, \dots, \sigma^q\tau]_1, [1, \sigma, \tau, \sigma\tau]_2)$. 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; next, $\mathcal{D} = \mathbb{Z}_p$. Let pk be an Elgamal public key and \mathcal{S} be a set of size $\mathcal{S} \in \mathcal{D}^{\leq q}$ for fixed $q = \text{poly}(\lambda)$. Let $\mathbf{1p} = (\text{pk}, \mathcal{S})$. In the case of NIZKs for set membership and non-membership, we are interested in the following complementary (commit-and-prove) languages:

$$\begin{aligned} \mathcal{L}_{1p}^{\text{sm}} &= \{ [\text{ct}_\chi]_1 \mid \exists \chi, \varrho_\chi \text{ such that } \text{Enc}_{\text{pk}}([\chi]_1; \varrho_\chi) = [\text{ct}_\chi]_1 \wedge \chi \in \mathcal{S} \} , \\ \bar{\mathcal{L}}_{1p}^{\text{sm}} &= \{ [\text{ct}_\chi]_1 \mid \exists \chi, \varrho_\chi \text{ such that } \text{Enc}_{\text{pk}}([\chi]_1; \varrho_\chi) = [\text{ct}_\chi]_1 \wedge \chi \notin \mathcal{S} \} . \end{aligned}$$

Instead of defining two NIZKs for $\mathcal{L}_{1p}^{\text{sm}}$ and $\bar{\mathcal{L}}_{1p}^{\text{sm}}$, 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 $\text{mem} \in \{\text{Member}, \text{NotMember}\}$. The verifier assumes that its input is a membership (resp., non-membership) argument if $\text{mem} = \text{Member}$ (resp., NotMember). It outputs either Member , NotMember , or Error . We generalize the simulator similarly.

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. Thus, universal accumulators are set (non-)membership NIZKs in the case the input χ is public. That is, for $\mathbf{1p} = \mathcal{S}$, a universal (CRS-model) accumulator is a (non-zk) set

(non-)membership non-interactive argument system for the complementary languages $\mathcal{L}_{1p}^{\text{acc}} = \mathcal{S}$ and $\bar{\mathcal{L}}_{1p}^{\text{acc}} = \mathcal{D} \setminus \mathcal{S}$. 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*.

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

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

$$\Pr \left[\begin{array}{l} \mathcal{S} \in \mathcal{D}^{\leq q} \wedge \\ \left((\chi \notin \mathcal{S} \wedge v = \text{Member}) \vee \right. \\ \left. (\chi \in \mathcal{S} \wedge v = \text{NotMember}) \right) \end{array} \middle| \begin{array}{l} \mathbf{p} \leftarrow \text{Pgen}(1^\lambda); (\text{crs}, \text{td}) \leftarrow \text{Kgen}(\mathbf{p}, q); \\ (\mathcal{S}, \chi, \psi) \leftarrow \mathcal{A}(\text{crs}); \\ v \leftarrow \mathbf{V}(\text{crs}, \text{Com}(\text{crs}, \mathcal{S}), \chi, \psi) \end{array} \right] \approx_\lambda 0 .$$

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 $\mathcal{S} \in \mathcal{D}^{\leq q}$. 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 $[\mathbf{q}]_1 \leftarrow [(\mathbf{Z}_{\mathcal{S}}(\sigma) - r)/(\sigma - \chi)]_1$ is well defined even if $\sigma = \chi$. In this case,

$$f(X) = \frac{\mathbf{Z}_{\mathcal{S}}(X) - \mathbf{Z}_{\mathcal{S}}(\chi)}{X - \chi} = \prod_{s \in \mathcal{S} \setminus \{\chi\}} (X - s)$$

is clearly a polynomial, and thus we can set $[\mathbf{q}]_1 \leftarrow [f(\chi)]_1$.

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.

Pgen (1^λ): the same as the bilinear group generator; returns \mathbf{p} .
Kgen (\mathbf{p}, q): $\sigma \leftarrow \mathbb{Z}_p$; $\mathbf{crs} \leftarrow (\mathbf{p}, [(\sigma^i)_{i=0}^q]_1, [1, \sigma]_2)$; return $(\mathbf{crs}, \mathbf{td} = \sigma)$;
Com ($\mathbf{crs}, \mathcal{S}$): given $ \mathcal{S} = q$: output $[\mathbf{C}_\mathcal{S}]_1 \leftarrow [\mathbf{Z}_\mathcal{S}(\sigma)]_1$;
P ($\mathbf{crs}, \mathcal{S}, \chi$): $r \leftarrow \mathbf{Z}_\mathcal{S}(\chi)$; $[\mathbf{q}]_1 \leftarrow [(\mathbf{Z}_\mathcal{S}(\sigma) - r)/(\sigma - \chi)]_1$; If $\chi \in \mathcal{S}$ then $\psi \leftarrow [\mathbf{q}]_1$ else $\psi \leftarrow ([\mathbf{q}]_1, r)$; return ψ ;
V ($\mathbf{crs}, \mathbf{C}_\mathcal{S}, \chi, \psi$): If ψ parses as $\psi = ([\mathbf{q}]_1, r)$ and $r = 0$ then return Error ; If ψ parses as $\psi = [\mathbf{q}]_1$ then $r \leftarrow 0$; If $[\mathbf{q}]_1 \bullet ([\sigma]_2 - \chi[1]_2) + (r[1]_1 - [\mathbf{C}_\mathcal{S}]_1) \bullet [1]_2 \neq [0]_T$ then return Error ; If $r = 0$ then return Member else return NotMember ;

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

2.2 CLPØ NIZK

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

$$[\mathbf{ct}_\chi]_1 := \text{Enc}_{\mathbf{pk}}([\chi]_1; \mathbf{q}) = (\text{Enc}_{\mathbf{pk}}([\chi_i]_1; \mathbf{q}_i))_i .$$

Define

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

Hence, $\mathcal{L}_{\mathbf{1p}}$ is a commit-and-prove language. For example, if $F(X) = X^2 - X$, then $\mathcal{L}_{\mathbf{pk}, F}$ corresponds to the language of all Elgamal encryptions of Boolean values under the fixed public key \mathbf{pk} .

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

Affine map: $\mathbf{C}(\mathbf{X}) = \sum_{k=1}^\nu \mathbf{P}_k X_k + \mathbf{Q}$, where $\mathbf{P}_k, \mathbf{Q} \in \mathbb{Z}_p^{\ell \times \ell}$.

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

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

The quasideterminantal complexity $\text{qdc}(F)$ of F is the smallest QDR size of F . (Clearly, $\text{qdc}(F) \geq \deg(F)$.) We always assume that the polynomial F in $\mathbf{1p}$ satisfies $\text{qdc}(F) = \text{poly}(\lambda)$, that is, there exists a $\text{poly}(\lambda)$ -size QDR $\mathbf{C}(\mathbf{X})$ of F . [CLPØ21] showed that such QDRs exist for many F -s.

$\text{Pgen}(1^\lambda)$: returns the system parameters \mathbf{p} , as always.
$\text{Kgen}(\mathbf{p})$: $\mathbf{e} \leftarrow_{\$} \mathbb{Z}_p$; return $(\mathbf{crs}, \mathbf{td}) \leftarrow ([\mathbf{e}]_2, \mathbf{e})$;
$\text{Com}(\mathbf{crs}, \mathbf{lp})$: return $\mathbf{crs}_{1\mathbf{p}} \leftarrow (\mathbf{crs}, \mathbf{lp})$;
$\text{P}(\mathbf{crs}_{1\mathbf{p}}, \mathbf{x} = [\mathbf{ct}_\chi]_1, \mathbf{w} = (\chi, \varrho))$: Write $\mathbf{C}(\chi) = (\mathbf{h} \parallel \mathbf{T})(\chi)$; $\varrho \delta \leftarrow_{\$} \mathbb{Z}_p^{\ell-1}$; $\gamma \leftarrow -\mathbf{T}(\chi) \varrho \delta$; Compute \mathbf{w} such that $\mathbf{T}(\chi) \mathbf{w} = \mathbf{h}(\chi)$; $[\delta]_2 \leftarrow -(\mathbf{w}[\mathbf{e}]_2 + \varrho \delta [1]_2)$; $\varrho_\gamma \leftarrow_{\$} \mathbb{Z}_p^\ell$; $[\mathbf{ct}_\gamma]_1 \leftarrow \text{Enc}_{\text{pk}}([\gamma]_1; \varrho_\gamma) \in \mathbb{G}_1^{\ell \times 2}$; $[\mathbf{z}]_2 \leftarrow (\sum_{k=1}^\nu \varrho_k \mathbf{P}_k) [\delta]_2 - \varrho_\gamma [1]_2 \in \mathbb{G}_2^\ell$; Return $\pi \leftarrow ([\mathbf{ct}_\gamma]_1, [\delta, \mathbf{z}]_2) \in \mathbb{G}_1^{\ell \times 2} \times \mathbb{G}_2^{2\ell-1}$;
$\text{V}(\mathbf{crs}_{1\mathbf{p}}, \mathbf{x} = [\mathbf{ct}_\chi]_1, \pi)$: check $\sum_{k=1}^\nu (\mathbf{P}_k [\delta]_2 \bullet [\mathbf{ct}_{\chi_k}]_1) + \mathbf{Q} [\delta]_2 \bullet [0 \parallel 1]_1 = [\mathbf{I}_\ell]_2 \bullet [\mathbf{ct}_\gamma]_1 + [\mathbf{z}]_2 \bullet \mathbf{pk}$;
$\text{Sim}(\mathbf{crs}_{1\mathbf{p}}, \mathbf{td}, \mathbf{x} = [\mathbf{ct}_\chi]_1)$: $\delta \leftarrow_{\$} \mathbb{Z}_p^{\ell-1}$; $\mathbf{z} \leftarrow_{\$} \mathbb{Z}_p^\ell$; $[\mathbf{ct}_\gamma]_1 \leftarrow \sum_{k=1}^\nu \mathbf{P}_k (\delta) [\mathbf{ct}_{\chi_k}]_1 + \text{Enc}_{\text{pk}}(\mathbf{Q}(\delta) [1]_1; -\mathbf{z})$; Return $\pi \leftarrow ([\mathbf{ct}_\gamma]_1, [\delta, \mathbf{z}]_2) \in \mathbb{G}_1^{\ell \times 2} \times \mathbb{G}_2^{2\ell-1}$;

Fig. 3. The commit-and-prove CLP \emptyset NIZK $\Pi_{\text{clp}\emptyset}$ for $\mathcal{L}_{\text{pk}, F}$.

CLP \emptyset Argument. In Fig. 3, we depict the commit-and-prove updatable universal CLP \emptyset NIZK $\Pi_{\text{clp}\emptyset}$. Intuitively, the verifier checks that

$$[\delta]_2 \bullet [\mathbf{C}(\mathbf{ct}_\chi)]_1 = [\mathbf{I}_\ell]_2 \bullet [\mathbf{ct}_\gamma]_1 + [\mathbf{z}]_2 \bullet \mathbf{pk} ,$$

where

$$[\mathbf{C}(\mathbf{ct}_\chi)]_1 := \sum_{k=1}^\nu \mathbf{P}_k \cdot [\mathbf{ct}_{\chi_k}]_1 + \mathbf{Q} \cdot \text{Enc}_{\text{pk}}(1; 0)$$

encrypts $\mathbf{C}(\chi)$. Couteau et al. [CLP \emptyset 21] did not use the terminology of commit-and-prove, universal, and updatable NIZKs. Still, $\Pi_{\text{clp}\emptyset}$ satisfies these properties.

We will state Fact 1 from [CLP \emptyset 21] for the sake of completeness.

Fact 1. *Let $\{\mathcal{D}_\mathbf{p}\}_\mathbf{p}$ be the family of language distributions, where $\mathcal{D}_\mathbf{p} = \{1\mathbf{p} = (\mathbf{pk}, F)\}$. Here, $F(\mathbf{X})$ is a ν -variate polynomial of degree d , where $\nu, d \in \text{poly}(\lambda)$. Let $\mathbf{C}(\mathbf{X}) \in \mathbb{Z}_p[\mathbf{X}]^{\ell \times \ell}$ be a QDR of F . The NIZK $\Pi_{\text{clp}\emptyset}$ for $\{\mathcal{D}_\mathbf{p}\}_\mathbf{p}$ 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 .*

Couteau et al. [CLP \emptyset 21] constructed a QDR $\text{IK}_F(\mathbf{X})$ for any polynomial F that can be efficiently computed by an algebraic branching program (ABP). As proven in [CLP \emptyset 21], if $\text{abp} = (V, E, s, t, \phi)$ is an ABP that computes a ν -variate polynomial $F(\mathbf{X})$, then $\text{IK}_F(\mathbf{X})$ is a QDR of F with $\ell = |V| - 1$. In particular, $\text{qdc}(F) \leq |V| - 1$. This results in NIZKs for $\mathcal{L}_{\text{pk}, F}$ whenever F has a small ABP.

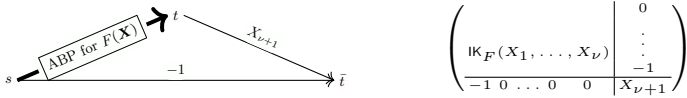


Fig. 4. ABP $\overline{\text{abp}}$ for the $\bar{F}(\mathbf{X}, X_{\nu+1}) = F(\mathbf{X})X_{\nu+1} - 1$ and the matrix $\text{IK}_{\bar{F}}(\mathbf{X}, X_{\nu+1})$.

Optimized verifier. As an independent contribution, we observe that the verifier can be batched by sampling $\eta \leftarrow_s \mathbb{F}$ and then batching together all pairings with the same \mathbb{G}_2 argument. It is evident from Fig. 3 that in this case, the verifier needs to perform pairings (at most) with \mathbf{e} , each coefficient of δ , $[1]_2$, and each coefficient of \mathbf{z} . Moreover, pairings with different z_i can also be batched together since they will have related \mathbb{G}_1 elements. This results in at most $\ell + 2$ pairings. On the other hand, a non-batched verifier may have to execute $\Theta(\ell^2)$ pairings.

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) 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 details of $\Pi_{\text{clp}\emptyset}$. Since non-membership NIZKs have their own applications [ATSM09,BCV15,BDSS16,BBLP21], the current section has independent importance.

New Non-Membership NIZK. Assume $\mathcal{F} = \{F\}$, where $F(\mathbf{X}) : \mathbb{F}_p^\nu \mapsto \mathbb{F}_p$ 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, $s \rightarrow \bar{t}$ and $t \rightarrow \bar{t}$. We naturally extend ϕ to a new labeling function $\bar{\phi}$, such that $\bar{\phi}(s \rightarrow \bar{t}) = -1$ and $\bar{\phi}(t \rightarrow \bar{t}) = X_{\nu+1}$, where $X_{\nu+1}$ is a new indeterminate. Let

$$\bar{F}(\mathbf{X}, X_{\nu+1}) : \mathbb{F}_p^{\nu+1} \mapsto \mathbb{F}_p, \quad \bar{F}(\mathbf{X}, X_{\nu+1}) = F(\mathbf{X})X_{\nu+1} - 1,$$

be the polynomial computed by $\overline{\text{abp}}$. Clearly, if $F(\chi) = 0$ for a concrete input assignment χ , then $\bar{F}(\chi, \chi_{\nu+1}) = -1 \neq 0$ for all values of $\chi_{\nu+1}$. On the other hand, if $F(\chi) \neq 0$, then there exists $\chi_{\nu+1} = F(\chi)^{-1}$, such that $\bar{F}(\chi, \chi_{\nu+1}) = 0$.

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 $\mathcal{S} = \mathcal{Z}(\bar{F})$. For this, one can use $\Pi_{\text{clp}\emptyset}$ from Fig. 4 for the QDR $\text{IK}_{\bar{F}}$. 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 $\Pi_{\text{clp}\emptyset}$ with the QDR $\bar{\mathcal{C}}(\mathbf{X}, \mathbf{S}) := \begin{pmatrix} \mathbf{X} & \\ -1 & \mathbf{S} \end{pmatrix}$ where in the honest case, $\mathbf{S} = 1/\mathbf{X}$. One can easily extend it to the proof

that two plaintexts χ_1 and χ_2 are unequal, by using the QDR $\bar{C}(X_1, X_2, S) := \begin{pmatrix} X_1 - X_2 & -1 \\ -1 & S \end{pmatrix}$, where in the honest case, $S = 1/(X_1 - X_2)$.

The argument length of the resulting NIZKs (including encryption of s but not of χ or χ_i) is $6g_1 + 3g_2$. 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.)

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

$$C_{\{0,1\}}(X, S) := \begin{pmatrix} X & -1 & 0 \\ 0 & X-1 & -1 \\ -1 & 0 & S \end{pmatrix} .$$

Generalization. Let $\mathcal{F} = \{F_1, \dots, F_\nu\}$ 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 subsection). Then, we construct an ABP that computes a polynomial $\bar{F}(\mathbf{X})$, s.t. $\bar{F}(\chi) = 0$ iff $\bar{F}_i(\chi) = 0$ for some i . Define $\bar{F}(\mathbf{X}) = \prod \bar{F}_i(\mathbf{X})$, and define its ABP as the concatenation of the ABPs for individual polynomials \bar{F}_i :



We then use $\Pi_{\text{clp}\emptyset}$ for the QDR $\text{IK}_{\bar{F}}$ from Fig. 4. The resulting NIZK is secure under the CED assumption (see Fact 1).

4 Determinantal Accumulators

Clearly, universal accumulator is a *non-zk* set (non-)membership non-interactive argument system that possesses both membership and non-membership arguments. It makes sense to construct a set (non-)membership NIZK by constructing an accumulator and then adding a zero-knowledge layer to obtain privacy.

Both steps of the described blueprint can be expensive per se. We are interested in constructing a CLP \emptyset -style set (non-)membership NIZK where the second step is as simple as possible. To achieve this, we first reinterpret $\Pi_{\text{clp}\emptyset}$. We then use the obtained understanding to define and construct *determinantal accumulators* that allow for a lightweight zero-knowledge layer. For the latter, a determinantal accumulator must have a specific structure consistent with $\Pi_{\text{clp}\emptyset}$'s design.

Intuition. Recall that in $\Pi_{\text{clp}\emptyset}$ [CLP \emptyset 21], one rewrites the condition $\chi \in \mathcal{S}$ as the condition $F_i(\chi) = 0$ for a set of polynomials $\{F_i\}$.³ After that, one constructs QDRs $C_i(\mathbf{X})$ for each F_i , such that $\det C_i(\mathbf{X}) = F_i(\mathbf{X})$. This step can be seen as linearization: while F_i can be a high-degree polynomial, each entry of C_i is an

³ In our new primitives, the set consists of only one polynomial. However, the framework is valid in the more general case.

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.

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

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 $\Pi_{\text{clp}\emptyset}$, V has access to $[\boldsymbol{\chi}]_1$ and thus also to $[\mathbf{C}_i(\boldsymbol{\chi})]_1$. To be compatible with CLPØ, the verifier is not however given access to $\det \mathbf{C}_i(\boldsymbol{\chi})$ or even $\boldsymbol{\chi}$ as integers. Given this, we must take additional care to ensure the accumulator's security.

4.1 Determinant Verification

The verifier needs to check efficiently that the determinant of a given matrix $\mathbf{C}_i(\boldsymbol{\chi})$ is zero. The main problem is that since the verifier sees $[\mathbf{C}_i(\boldsymbol{\chi})]_1$ but not $\mathbf{C}_i(\boldsymbol{\chi})$, 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 $[\mathbf{C}_i(\boldsymbol{\chi})]_1$ to obtain $\mathbf{C}_i(\boldsymbol{\chi})$. 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 $\mathbf{C}_i(\boldsymbol{\chi})$, 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].⁴ Moreover, since in the NIZK, one has to encrypt the matrix elements in both groups, and use the less efficient DLIN encryption, see [CLPØ21].

Third, we can use the undergraduate linear-algebraic fact that $\det \mathbf{C} = 0$ iff there exists a non-zero vector \boldsymbol{x} such that $\mathbf{C}\boldsymbol{x} = \mathbf{0}$. We can utilize this fact by outputting $[\boldsymbol{x}]_2$ as a hint to the verifier/soundness reduction. However, $[\boldsymbol{x}]_2$ can

⁴ In the case of 2×2 matrices, the hint can be $[\mathbf{C}_1]_2$, where \mathbf{C}_1 is the first row of $\mathbf{C} \in \mathbb{Z}_p^{2 \times 2}$ [ALSZ20]. In the case of $\mathbf{C} \in \mathbb{Z}_p^{3 \times 3}$, the prover already needs to output six values $[C_{1i}C_{2j}]_2$ for $i \neq j$.

reveal secret information and thus must be hidden. We do not want to encrypt $[\mathbf{x}]_2$: since $[\mathbf{x}]_2$ is given in \mathbb{G}_2 , this means that one again needs to use DLIN.

Fourth, we rely on CED. Recall that CED states that $\det \mathbf{C} = 0$ iff one can compute vectors γ and δ such that $\mathbf{C} \begin{pmatrix} \mathbf{e} \\ \delta \end{pmatrix} = \gamma$, where $\mathbf{e} \leftarrow_{\$} \mathbb{Z}_p$. (The first coordinate of $\mathbf{x} = \begin{pmatrix} \mathbf{e} \\ \delta \end{pmatrix}$ is non-zero w.p. $1 - 1/p$ since \mathbf{C} is a QDR.) For the security of CED, γ must not depend on \mathbf{e} . Here, as in [CH20,CLPØ21], δ is masked by uniformly random addend ρ_δ and γ is needed to balance ρ_δ . Thus, the prover gives $([\gamma]_1, [\delta]_2)$ as a hint to the verifier/soundness reduction. In the NIZK, $[\gamma]_1$ is encrypted but $[\delta]_2$ (that looks uniformly random after adding ρ_δ) is not. While the resulting accumulator is less efficient than Nguyen’s, the new NIZK (see Section 6) is very efficient since it reuses the hints $([\gamma]_1, [\delta]_2)$.

Definition. The reasoning from Section 4.1 shows that one can construct an efficient accumulator (and NIZK) even if χ is only given to the verifier in \mathbb{G}_1 . This motivates the new definition of determinantal accumulators. The relation between determinantal accumulators and CLPØ is similar to that between structure-preserving signatures and Groth-Sahai. For comparison purposes only, we will define structure-preserving signature schemes [AFG⁺16].

Definition 2 (Structure-preserving signatures [AFG⁺16]). *A digital signature scheme is structure preserving relative to bilinear group generator Pgen if*

- (1) *the common parameters \mathbf{p} and the CRS 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 \mathcal{V} consists only of evaluating membership in \mathbb{G}_1 and \mathbb{G}_2 and relations described by paring product equations,*
- (3) *verification keys \mathbf{vk} , messages χ and signatures σ solely consist of group elements in \mathbb{G}_1 and \mathbb{G}_2 .*

([AFG⁺16] did not mention the CRS, but follow-up works do.) 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 \mathbf{p} and the CRS 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 \mathcal{V} consists only of evaluating membership in \mathbb{G}_1 and \mathbb{G}_2 and relations described by checking that $\mathbf{C}_i(\chi) = 0$, where each $\mathbf{C}_i(X)$ is a QDR,*
- (c) *the CRS \mathbf{crs} , messages χ , commitments \mathbf{C}_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 δ 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 C_i can be polynomially large. In the new non-membership accumulator, the determinant of the used 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 \mathbf{AC}^*

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 \mathbf{ACC} is F -collision resistant if for any $q = \text{poly}(\lambda)$ and PPT adversaries \mathcal{A} , $\text{Adv}_{\text{Pgen}, F, \mathbf{ACC}, \mathcal{A}}^{\text{f-cr}}(\lambda) :=$

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

Here, we highlighted the differences with Definition 1.

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

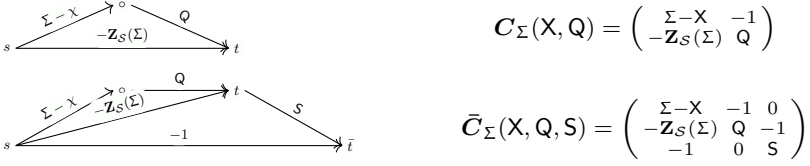


Fig. 5. Above: ABP for $F_{\Sigma}(X, Q)$ and the QDR $C_{\Sigma}(X, Q)$. Below: ABP for $\bar{F}_{\Sigma}(X, Q, S)$ and the QDR $\bar{C}_{\Sigma}(X, Q, S)$.

Construction. In Fig. 6, 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 $F(\mathbf{X})$, such that the verifier accepts iff $F(\chi) = 0$. Given F , we use an ABP to define a QDR $C(\mathbf{X})$ for F .

In the membership argument, we start with the verification equation of ACC_{Nguyen} which defines the bivariate polynomial

$$F_{\Sigma}(X, Q) := (\Sigma - X)Q - Z_S(\Sigma) .$$

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

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

$$\tilde{F}_{\Sigma}(X, Q, S) := ((\Sigma - X)Q - Z_S(\Sigma))S - 1 .$$

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}_{Σ} as low-degree polynomials with coefficients from $\mathcal{R} = \mathbb{Z}_p[\Sigma]$.

Since F_{Σ} and \tilde{F}_{Σ} have degrees ≤ 2 and ≤ 3 , they have respectively 2×2 and 3×3 QDRs $C_{\Sigma}(X, Q)$ and $\bar{C}_{\Sigma}(X, Q, S)$. We construct these QDRs from algebraic branching programs for F_{Σ} and \tilde{F}_{Σ} . See Fig. 5 for the description of the resulting ABP and QDR for F_{Σ} and \tilde{F}_{Σ} . The membership (resp., non-membership) argument verifier needs to check that $\det C(\chi, q) = 0$ (resp., $\det \bar{C}(\chi, q, s) = 0$).

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

$$\begin{aligned} ([\sigma]_1 - [\chi]_1) \bullet [e]_2 - [1]_1 \bullet [\delta]_2 &= [\gamma]_1 \bullet [1]_2 , \\ -[Z_S(\sigma)]_1 \bullet [e]_2 + [q]_1 \bullet [\delta]_2 &= [\gamma]_2 \bullet [1]_2 . \end{aligned} \tag{2}$$

Here, $[\chi]_1$ is the input, $([q, \gamma]_1, [\delta]_2)$ are parts of the (non-)membership argument, and $[\sigma, Z_S(\sigma)]_1$ can be computed from crs .

Unfortunately, this is not sufficient. Maliciously chosen $\chi = \chi(\Sigma)$, $\mathbf{q} = \mathbf{q}(\Sigma)$, and $\delta = \delta(\Sigma)$ can depend non-trivially on σ . Intuitively, Eq. (2) guarantees that $\mathbf{Z}_S(\Sigma) = (\Sigma - \chi(\Sigma))\mathbf{q}(\Sigma)$ and thus $(\Sigma - \chi(\Sigma)) \mid \mathbf{Z}_S(\Sigma)$. If χ is an integer, we get $\mathbf{Z}_S(\chi) = 0$. However, if χ depends on σ , then $\mathbf{Z}_S(\chi) = 0$ does not follow. E.g., to break the membership argument, the adversary can fix any $\delta_1, \delta_2 \in \mathbb{Z}_p$ and set $[\chi]_1 \leftarrow [\sigma]_1 - \delta_2[1]_1$, $[\delta]_2 \leftarrow \delta_1[1]_2 + \delta_2[\mathbf{e}]_2$, $[\mathbf{q}]_1 \leftarrow [\mathbf{Z}_S(\sigma)]_1/\delta_2$, $[\gamma_1]_1 \leftarrow -[\delta_1]_1$, $[\gamma_2]_1 \leftarrow \delta_1/\delta_2 \cdot [\mathbf{Z}_S(\sigma)]_1$. 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 $\text{ACC}_{\text{Nguyen}}$ is even more trivial.⁵

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. 6.

Since now 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 $[\mathbf{q}]_1 \bullet [\delta]_2$ and $[\gamma]_1 \bullet [1]_2$. In the AGM security proof of the underlying assumption, we get that values like χ , which are multiplied by τ , are in the span of 1 (that is, integers). However, \mathbf{q} must be a polynomial (it depends on σ), that is, in the span of $\{\sigma^i\tau\}$; thus we do not multiply $[\mathbf{q}]_1 \bullet [\delta]_2$ 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 ($b_1 = b_2 = \text{true}$) are mathematically (but not computationally) equivalent to checking that $\mathbf{C}'(\chi, \mathbf{q})(\frac{\mathbf{e}}{\delta}) = \gamma$, where

$$\mathbf{C}'(\mathbf{X}, \mathbf{Q}) := \begin{pmatrix} (\Sigma - \mathbf{X})^\top & -\mathbf{T} \\ -\mathbf{Z}_S(\Sigma)^\top & \mathbf{Q} \end{pmatrix} .$$

Here,

$$\det \mathbf{C}'(\mathbf{X}, \mathbf{Q}) = ((\Sigma - \mathbf{X})\mathbf{Q} - \mathbf{Z}_S(\Sigma)\mathbf{T})^\top .$$

That is, we really use the QDR framework of [CLPØ21]. The description of \mathbb{V} in Fig. 6 just spells out how to do this verification in PPT.

Non-Membership Argument. The non-membership argument verifier checks that $[\tilde{\mathbf{C}}(\chi)]_1 \bullet [\frac{\mathbf{e}}{\delta}]_2 = [\gamma]_1 \bullet [1]_2$ (where now $\delta \in \mathbb{Z}_p^2$ and $\gamma \in \mathbb{Z}_p^3$; see Fig. 5), which can be rewritten as checking

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

⁵ In the collision-resistance proof of $\text{ACC}_{\text{Nguyen}}$, χ and \mathbf{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 $[\mathbf{e}]_2$ and thus did not depend on σ .

$\text{Kgen}(\mathbf{p}, q): \sigma, \tau, \mathbf{e} \leftarrow_{\$} \mathbb{Z}_p; \text{crs} \leftarrow ([1, (\sigma^i \tau)_{i=0}^q]_1, [1, \mathbf{e}, \sigma \mathbf{e}, \tau \mathbf{e}]_2); \text{td} \leftarrow (\mathbf{e}, \tau);$ return (crs, td) .
$\text{Com}(\text{crs}, \mathcal{S}): [\mathbf{C}_S]_1 \leftarrow [\mathbf{Z}_S(\sigma)\tau]_1; \text{return crs}_{1p} \leftarrow (\text{crs}, [\mathbf{C}_S]_1, \mathcal{S});$
$\text{P}(\text{crs}_{1p}, \chi): \mathbf{r} \leftarrow \mathbf{Z}_S(\chi); f(X) \leftarrow (\mathbf{Z}_S(X) - \mathbf{r}) / (X - \chi); [\mathbf{q}]_1 \leftarrow [f(\sigma)\tau]_1;$ if $\chi \in \mathcal{S}$ then <ol style="list-style-type: none"> 1. $\varrho_\delta \leftarrow_{\\$} \mathbb{Z}_p; [\gamma]_1 \leftarrow - \begin{bmatrix} -\tau \\ \mathbf{q} \end{bmatrix}_1 \varrho_\delta; [\delta]_2 \leftarrow [\sigma \mathbf{e}]_2 - \chi[\mathbf{e}]_2 - \varrho_\delta[1]_2;$ 2. $\psi \leftarrow ([\mathbf{q}, \gamma]_1, [\delta]_2); \quad // \quad 3\mathbf{g}_1 + \mathbf{g}_2$ else <ol style="list-style-type: none"> 1. $\mathbf{s} \leftarrow \frac{1}{r}; \varrho_\delta \leftarrow_{\\$} \mathbb{Z}_p^2; [\gamma]_1 \leftarrow - \begin{bmatrix} -\tau & 0 \\ \mathbf{q} & -\tau \\ 0 & \mathbf{s} \end{bmatrix}_1 \varrho_\delta; [\delta]_2 \leftarrow \begin{pmatrix} [\sigma \mathbf{e}]_2 - \chi[\mathbf{e}]_2 \\ r \cdot [\mathbf{e}]_2 \end{pmatrix} - \varrho_\delta[1]_2;$ 2. $\psi \leftarrow ([\mathbf{q}, \mathbf{s}, \gamma]_1, [\delta]_2); \quad // \quad 5\mathbf{g}_1 + 2\mathbf{g}_2$ return ψ ;
$\text{V}(\text{crs}_{1p}, [\chi]_1, \psi): \text{mem} \leftarrow \text{NotMember};$ If ψ parses as $\psi = ([\mathbf{q}, \gamma]_1, [\delta]_2)$ then $\text{mem} \leftarrow \text{Member};$ If $\text{mem} = \text{Member}$ then <ol style="list-style-type: none"> 1. $b_1 \leftarrow [\sigma\tau]_1 \bullet [\mathbf{e}]_2 - [\chi]_1 \bullet [\tau \mathbf{e}]_2 - [\tau]_1 \bullet [\delta]_2 \stackrel{?}{=} [\gamma]_1 \bullet [1]_2;$ 2. $b_2 \leftarrow -[\mathbf{C}_S]_1 \bullet [\mathbf{e}]_2 + [\mathbf{q}]_1 \bullet [\delta]_2 \stackrel{?}{=} [\gamma]_2 \bullet [1]_2;$ 3. if b_1 and b_2 then return Member else return Error; else <ol style="list-style-type: none"> 1. $\bar{b}_1 \leftarrow ([\sigma]_1 - [\chi]_1) \bullet [\tau \mathbf{e}]_2 - [\tau]_1 \bullet [\delta]_2 \stackrel{?}{=} [\gamma]_1 \bullet [1]_2;$ 2. $\bar{b}_2 \leftarrow -[\mathbf{C}_S]_1 \bullet [\mathbf{e}]_2 + [\mathbf{q}]_1 \bullet [\delta]_2 - [\tau]_1 \bullet [\delta]_2 \stackrel{?}{=} [\gamma]_2 \bullet [1]_2;$ 3. $\bar{b}_3 \leftarrow -[1]_1 \bullet [\mathbf{e}]_2 + [\mathbf{s}]_1 \bullet [\delta]_2 \stackrel{?}{=} [\gamma]_3 \bullet [1]_2;$ 4. if \bar{b}_1 and \bar{b}_2 and \bar{b}_3 then return NotMember else return Error;

Fig. 6. The new $[\cdot]_1$ -collision-resistant determinantal universal accumulator AC^* .

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

The verification equations (that is, $\bar{b}_1 = \bar{b}_2 = \bar{b}_3 = \text{true}$, see Fig. 6) are equivalent to checking that $\bar{\mathbf{C}}'(\chi, \mathbf{q}, \mathbf{s}) \begin{pmatrix} \mathbf{e} \\ \delta \end{pmatrix} = \gamma$, where

$$\bar{\mathbf{C}}'(\chi, \mathbf{Q}, \mathbf{S}) := \begin{pmatrix} (\Sigma - \chi)\mathbf{T} & -\mathbf{T} & 0 \\ -\mathbf{Z}_S(\Sigma)\mathbf{T} & \mathbf{Q} & -\mathbf{T} \\ -1 & 0 & \mathbf{S} \end{pmatrix},$$

with $\det \bar{\mathbf{C}}'(\chi, \mathbf{Q}) = ((\Sigma - \chi)\mathbf{Q} - \mathbf{Z}_S(\Sigma)\mathbf{T})\mathbf{s}\mathbf{T} - \mathbf{T}^2$.

Description. We depict AC^* in Fig. 6. As explained before, the membership verifier checks (on pairings) that $\mathbf{C}'(\chi, \mathbf{q}) \cdot \begin{pmatrix} \mathbf{e} \\ \delta \end{pmatrix} = \gamma$, and the non-membership verifier checks that $\bar{\mathbf{C}}'(\chi, \mathbf{q}, \mathbf{s}) \cdot \begin{pmatrix} \mathbf{e} \\ \delta \end{pmatrix} = \gamma$. Fig. 6 does it in PPT. A batched membership verifier has to execute four pairings, while a batched non-membership verifier has to execute five pairings. For example, the membership verifier samples a random $\eta \leftarrow_{\$} \mathbb{F}$, and then checks whether

$$\begin{aligned} ([\sigma\tau]_1 - \eta[\mathbf{C}_S]_1) \bullet [\mathbf{e}]_2 - [\chi]_1 \bullet [\tau \mathbf{e}]_2 + (\eta[\mathbf{q}]_1 - [\tau]_1) \bullet [\delta]_2 & \stackrel{?}{=} \\ ([\gamma]_1 + \eta[\gamma]_2)_1 \bullet [1]_2 & . \end{aligned}$$

Lemma 1. *AC* is perfectly complete.*

Proof. One can straightforwardly check that the choice of ϱ_δ , γ , and δ is consistent with Fig. 3 when one uses the correct matrices C' and \bar{C}' . Completeness follows straightforwardly. In particular, writing $C' = (\mathbf{h}' \| \mathbf{T}')$, we get that $\mathbf{h}' = \mathbf{T}'\mathbf{w}'$, where $\mathbf{w}' = -(\sigma - \chi)$. This explains why say

$$[\delta]_2 = -\mathbf{w}'[\mathbf{e}]_2 - \varrho_\delta[1]_2 = [(\sigma - \chi)\mathbf{e}]_2 - \varrho_\delta[1]_2 = [\sigma\mathbf{e}]_2 - \chi[\mathbf{e}]_2 - \varrho_\delta[1]_2 .$$

Then, say $b_1 = \text{true}$ since

$$(\sigma - \chi)\tau\mathbf{e} - \tau\delta = \gamma_1 \iff (\sigma - \chi)\tau\mathbf{e} - \tau((\sigma - \chi)\mathbf{e} - \varrho_\delta) = \tau\varrho_\delta ,$$

which is trivially true. In the case of non-membership proof, writing $\bar{C}' = (\bar{\mathbf{h}}' \| \bar{\mathbf{T}}')$, we get similarly that $\bar{\mathbf{h}}' = \bar{\mathbf{T}}'\bar{\mathbf{w}}'$, where

$$\bar{\mathbf{w}}' = \begin{pmatrix} -(\sigma - \chi) \\ -r \end{pmatrix} .$$

□

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.

F-Collision-Resistance. We define two tautological assumptions q -DETACM and q -DETACNM that essentially state that AC* is F -collision-resistant. Then, we prove in AGM that DETACM and 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.

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

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


```

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

```

Fig. 7. The adversary \mathcal{B} in the proof of Lemma 2

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

$$\Pr \left[\begin{array}{l} \mathcal{S} \in \mathcal{D}^{\leq q} \wedge \\ \chi \in \mathcal{S} \wedge \\ \bar{\mathbf{C}}'(\chi, \mathbf{q}, \mathbf{s})(\frac{\mathbf{e}}{\delta}) = \gamma \end{array} \middle| \begin{array}{l} \mathbf{p} \leftarrow \text{Pgen}(1^\lambda); \sigma, \tau, \mathbf{e} \leftarrow_{\$} \mathbb{Z}_p; \\ \mathbf{crs} \leftarrow (\mathbf{p}, [1, (\sigma^i \tau)_{i=0}^q]_1, [1, \mathbf{e}, \sigma \mathbf{e}, \tau \mathbf{e}]_2); \\ (\mathcal{S}, [\chi, \mathbf{q}, \mathbf{s}, \gamma]_1, [\delta]_2) \leftarrow \mathcal{A}(\mathbf{crs}); \\ \bar{\mathbf{C}}'(\chi, \mathbf{q}, \mathbf{s}) \leftarrow \begin{pmatrix} (\sigma - \chi)\tau & -\tau & 0 \\ -\mathbf{z}_{\mathcal{S}}(\sigma)\tau & \mathbf{q} & -\tau \\ -1 & 0 & \mathbf{s} \end{pmatrix} \end{array} \right] \approx_{\lambda} 0 .$$

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

Lemma 2 is trivial since DETACM and DETACNM are tautological assumptions for the F -collision-resistance of AC^* . The complicated step (see Theorem 1) is establishing that DETACM and DETACNM are secure in the AGM.

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

Proof. Let \mathcal{A} be an F -collision-resistance (see Definition 4) adversary for AC^* , such that $\text{Adv}_{\text{Pgen}, F, \text{AC}^*, \mathcal{A}}^{\text{f-cr}}(\lambda) = \varepsilon_{\mathcal{A}}$ for non-negligible $\varepsilon_{\mathcal{A}}$. In Fig. 7, we depict a trivial DETACM/DETACNM adversary \mathcal{B} . Clearly, with probability at least $\varepsilon_{\mathcal{A}}$, \mathcal{B} succeeds in breaking q -DETACM (resp., q -DETACNM), given \mathcal{A} outputs an accepting fake membership (resp., non-membership) argument. \square

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

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

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

$$\begin{aligned} \chi(\mathbf{X}) &= \chi_1(\Sigma)\mathbf{T} + \chi_2 , & \mathbf{q}(\mathbf{X}) &= \mathbf{q}_1(\Sigma)\mathbf{T} + \mathbf{q}_2 , \\ \gamma_1(\mathbf{X}) &= \gamma_{11}(\Sigma)\mathbf{T} + \gamma_{12} , & \gamma_2(\mathbf{X}) &= \gamma_{21}(\Sigma)\mathbf{T} + \gamma_{22} , \end{aligned}$$

$$\delta(\mathbf{X}) = \delta_1 + \delta_2 \mathbf{E} + \delta_3 \Sigma \mathbf{E} + \delta_4 \mathbf{T} \mathbf{E} \text{ ,}$$

where each polynomial (like \mathbf{q}_1) on the RHS is of degree $\leq q$. That is, the algebraic adversary \mathcal{A}_{alg} also outputs coefficients of all above polynomials. The DETACM verifier's checks guarantee that $V_1(\sigma, \tau, \mathbf{e}) = V_2(\sigma, \tau, \mathbf{e}) = 0$, where

$$\begin{aligned} V_1(\mathbf{X}) &= ((\Sigma - \chi(\mathbf{X})) \mathbf{E} - \delta(\mathbf{X})) \cdot \mathbf{T} - \gamma_1(\mathbf{X}) \text{ ,} \\ V_2(\mathbf{X}) &= (\mathbf{r}(\mathbf{X}) - \mathbf{Z}_S(\Sigma)) \mathbf{T} \mathbf{E} + \mathbf{q}(\mathbf{X}) \delta(\mathbf{X}) - \gamma_2(\mathbf{X}) \text{ .} \end{aligned}$$

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 $\mathcal{R}[\mathbf{T}, \mathbf{E}]$, where $\mathcal{R} = \mathbb{Z}_p[\Sigma]$. We now enlist the coefficients of $\mathbf{T}^i \mathbf{E}^j$ 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):

$(i, j) \ V_1$	$(i, j) \ V_2$
$(2, 1) \ -\delta_4 - \chi_1(\Sigma)$	$(2, 1) \ \delta_4 \mathbf{q}_1(\Sigma)$
$(1, 1) \ -\delta_2 + (1 - \delta_3) \Sigma - \chi_2$	$(1, 1) \ \delta_4 \mathbf{q}_2 + (\delta_2 + \delta_3 \Sigma) \mathbf{q}_1(\Sigma) - \mathbf{Z}_S(\Sigma)$
$(1, 0) \ -\gamma_{11}(\Sigma) - \delta_1$	$(1, 0) \ \delta_1 \mathbf{q}_1(\Sigma) - \gamma_{21}(\Sigma)$
$(0, 0) \ -\gamma_{12}$	$(0, 1) \ (\delta_2 + \delta_3 \Sigma) \mathbf{q}_2$
	$(0, 0) \ -\gamma_{22} + \delta_1 \mathbf{q}_2$

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

From the coefficient of $\mathbf{T}^2 \mathbf{E}$ of V_1 , we get $\chi_1(\Sigma) = -\delta_4$. From the coefficient of $\mathbf{T} \mathbf{E}$ 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 $\mathbf{T}^2 \mathbf{E}$ of V_2 , we get $\delta_4 \mathbf{q}_1(\Sigma) = 0$. Thus, either $\mathbf{q}_1(\Sigma) = 0$ or $\delta_4 = 0$. Taking into account what we already know, from the coefficient of $\mathbf{T} \mathbf{E}$ of V_2 , we get

$$\mathbf{Z}_S(\Sigma) = \delta_4 \mathbf{q}_2 + (\Sigma - \chi_2) \mathbf{q}_1(\Sigma) \text{ .}$$

Recall that we have either $\mathbf{q}_1(\Sigma) = 0$ or $\delta_4 = 0$. If $\mathbf{q}_1(\Sigma) = 0$, then $\mathbf{Z}_S(\Sigma) = \delta_4 \mathbf{q}_2 \in \mathbb{Z}_p$, a contradiction. Hence, $\delta_4 = 0$. Thus, $\mathbf{Z}_S(\Sigma) = (\Sigma - \chi_2) \mathbf{q}_1(\Sigma)$ and $(\Sigma - \chi_2) \mid \mathbf{Z}_S(\Sigma)$, which gives us $\mathbf{Z}_S(\chi_2) = 0$. Moreover, $\chi(\mathbf{X}) = \chi_1(\Sigma) \mathbf{T} + \chi_2 = \chi_2$, 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 $\mathcal{B}(\{\{\sigma^i\}_1\}_{i=0}^{q+1}, \{\{\sigma^i\}_1\}_{i=0}^2)$. \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 $\mathbf{e} \leftarrow \alpha_2 \sigma + \beta_2$. Then, \mathcal{B} creates crs for the DETACM adversary \mathcal{A}_{alg} and calls \mathcal{A}_{alg} with crs . After obtaining π , together with the coefficients of the polynomials like $\chi(\Sigma)$, from \mathcal{A}_{alg} , \mathcal{B} reconstructs the coefficients of the degree- $\leq (q+2)$ polynomial V_2 (which is now univariate since τ and \mathbf{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 $q+2$ roots x_i of V_2 . \mathcal{B} tests which one of them is equal to σ , and returns σ . \square

See Appendix C.1 for the proof of the following result.

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

6 New Set (Non-)Membership NIZK

Next, we use AC^* to construct a succinct set (non-)membership NIZK Π^* . First, Π^* '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 $\psi = \text{AC}^*.P(\text{crs}, \mathcal{S}, \chi)$. 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 Π^* (see Fig. 8). Π^* handles both $\mathcal{L}_{1p}^{\text{sm}}$ (set membership arguments, $mem = \text{Member}$) and $\bar{\mathcal{L}}_{1p}$ (set non-membership arguments, $mem = \text{NotMember}$). The prover of Π^* implements the prover of AC^* but it also additionally encrypts all \mathbb{G}_1 . To make the verification on ciphertexts possible, the prover outputs additional randomizer hints $[z]_2$. The verifier performs AC^* verification on ciphertexts (this relies on the homomorphic properties of Elgamal), taking $[z]_2$ into account. Π^* also defines the simulator algorithm.

Alternatively, Π^* is a version of $\Pi_{\text{clp}\varnothing}$ for the concrete choice of the QDRs (and different CRS). To see the connection between Fig. 8 and Fig. 3, note that $C'(X, Q) = Q + P_1X + P_2Q$, where

$$Q = \begin{pmatrix} \Sigma^\top & -1 \\ -\mathbf{z}_{\mathcal{S}(\Sigma)\top} & 0 \end{pmatrix}, \quad P_1 = \begin{pmatrix} -\tau & 0 \\ 0 & 0 \end{pmatrix}, \quad P_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

For example, starting with Fig. 3,

$$\begin{aligned} [z]_2 &= \sum_{k=1}^{\nu} \varrho_k P_k [\delta]_2 - \varrho_\gamma [1]_2 = \varrho_\chi \begin{pmatrix} -\tau & 0 \\ 0 & 0 \end{pmatrix} [\delta]_2 + \varrho_q \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} [\delta]_2 - \varrho_\gamma [1]_2 \\ &= \varrho_\chi \begin{bmatrix} -\tau e \\ 0 \end{bmatrix}_2 + \varrho_q \begin{bmatrix} 0 \\ \delta \end{bmatrix}_2 - \varrho_\gamma [1]_2 = \begin{pmatrix} -\varrho_\chi [\tau e]_2 \\ \varrho_q [\delta]_2 \end{pmatrix} - \varrho_\gamma [1]_2. \end{aligned}$$

One can represent $\bar{C}'(X, Q, R)$ similarly.

Clearly, Π^* is commit-and-prove, updatable, and universal.

Theorem 3. *The set membership argument Π^* in Fig. 8 is perfectly complete. Assuming Elgamal is IND-CPA secure, it is computationally zero-knowledge.*

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

Membership Argument. Clearly, $b_1 = \text{true}$ iff

$$\text{Enc}_{\text{pk}}([\sigma]_1; 0)\tau e - [\text{ct}_\chi]_1 \tau e - \text{Enc}_{\text{pk}}([\tau]_1; 0)\delta \stackrel{?}{=} [\text{ct}_{\gamma_1}]_1 + z_1 \cdot \text{pk}$$

\iff

$$\text{Enc}((\sigma - \chi)\tau e - \tau\delta; -\varrho_\chi \tau e) \stackrel{?}{=} \text{Enc}(\gamma_1; \varrho_{\gamma_1}) + \text{Enc}_{\text{pk}}(0; z_1).$$

Clearly,

$$(\sigma - \chi)\tau e - \tau\delta = (\sigma - \chi)\tau e - \tau((\sigma - \chi)e - \varrho_\delta) = \varrho_\delta \tau = \gamma_1 \tau,$$

$\text{Pgen}(1^\lambda): \mathbf{p} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, [1]_1, [1]_2) \leftarrow \text{Pgen}(1^\lambda).$
$\text{Kgen}(\mathbf{p}): (\text{crs}, \text{td}) \leftarrow \text{AC}^*. \text{Kgen}(\mathbf{p});$
$\text{Com}(\text{crs}, \text{lp} = (\text{pk}, \mathcal{S})): \text{AC}^*. \text{lp} \leftarrow \mathcal{S}; \text{AC}^*. \text{crs}_{1\mathbf{p}} \leftarrow \text{AC}^*. \text{Com}(\text{crs}, \text{AC}^*. \text{lp}); \text{return } \text{crs}_{1\mathbf{p}} \leftarrow (\text{AC}^*. \text{crs}_{1\mathbf{p}}, \text{pk});$
$\text{P}(\text{crs}_{1\mathbf{p}}, \mathbf{x} = [\text{ct}_\chi]_1, \mathbf{w} = (\chi, \varrho_\chi)): \\ \text{AC}^*. \psi \leftarrow \text{AC}^*. \text{P}(\text{AC}^*. \text{crs}_{1\mathbf{p}}, \chi); \quad // \quad \psi = ([q, \gamma]_1, [\delta]_2) \text{ or } \psi = ([q, s, \gamma]_1, [\delta]_2) \\ \varrho_q \leftarrow \mathbb{Z}_p; [\text{ct}_q]_1 \leftarrow \text{Enc}_{\text{pk}}([q]_1; \varrho_q); \\ \text{If } \chi \in \mathcal{S} \text{ then} \\ 1. \varrho_\gamma \leftarrow \mathbb{Z}_p^2; [\text{ct}_\gamma]_1 \leftarrow \text{Enc}_{\text{pk}}([\gamma]_1; \varrho_\gamma) \in \mathbb{G}_1^{2 \times 2}; [\mathbf{z}]_2 \leftarrow \begin{pmatrix} -\varrho_\chi[\tau e]_2 \\ \varrho_q[\delta]_2 \end{pmatrix} - \varrho_\gamma[1]_2 \in \mathbb{G}_2^2; \\ 2. \pi \leftarrow ([\text{ct}_q, \text{ct}_\gamma]_1, [\delta, \mathbf{z}]_2) \\ \text{else} \\ 1. \varrho_s \leftarrow \mathbb{Z}_p; [\text{ct}_s]_1 \leftarrow \text{Enc}_{\text{pk}}([s]_1; \varrho_s) \in \mathbb{G}_1^{1 \times 2}; \\ 2. \varrho_\gamma \leftarrow \mathbb{Z}_p^3; [\text{ct}_\gamma]_1 \leftarrow \text{Enc}_{\text{pk}}([\gamma]_1; \varrho_\gamma) \in \mathbb{G}_1^{3 \times 2}; [\mathbf{z}]_2 \leftarrow \begin{pmatrix} -\varrho_\chi[\tau e]_2 \\ \varrho_q[\delta]_{12} \\ \varrho_s[\delta]_{22} \end{pmatrix} - \varrho_\gamma[1]_2 \in \mathbb{G}_2^3; \\ 3. \pi \leftarrow ([\text{ct}_q, \text{ct}_s, \text{ct}_\gamma]_1, [\delta, \mathbf{z}]_2); \\ \text{return } \pi; \quad // \quad \text{membership: } 6\mathbf{g}_1 + 3\mathbf{g}_2; \text{ non-membership: } 10\mathbf{g}_1 + 5\mathbf{g}_2$
$\text{Sim}(\text{crs}_{1\mathbf{p}}, \text{td} = (\mathbf{e}, \tau), \mathbf{x} = [\text{ct}_\chi]_1, \text{mem} \in \{\text{Member}, \text{NotMember}\}): \\ \text{If } \text{mem} = \text{Member} \text{ then} \\ 1. \delta \leftarrow \mathbb{Z}_p; \mathbf{z} \leftarrow \mathbb{Z}_p^2; \varrho_q \leftarrow \mathbb{Z}_p; [\text{ct}_q]_1 \leftarrow \text{Enc}_{\text{pk}}(0; \varrho_q); \\ 2. [\text{ct}_\gamma]_1 \leftarrow \begin{pmatrix} \text{Enc}_{\text{pk}}([\sigma\tau]_{1;0}) - [\text{ct}_\chi]_1 \cdot \tau - \text{Enc}_{\text{pk}}([\tau]_{1;0}) & (\frac{\mathbf{e}}{[\text{ct}_q]_1}) \\ -\text{Enc}_{\text{pk}}([\mathcal{C}_S]_{1;0}) & [\text{ct}_q]_1 \end{pmatrix} (\delta) - \text{Enc}_{\text{pk}}(\mathbf{0}; \mathbf{z}); \\ 3. \pi \leftarrow ([\text{ct}_q, \text{ct}_\gamma]_1, [\delta, \mathbf{z}]_2) \\ \text{else} \\ 1. \delta \leftarrow \mathbb{Z}_p^2; \mathbf{z} \leftarrow \mathbb{Z}_p^3; \\ 2. \varrho_q, \varrho_s \leftarrow \mathbb{Z}_p; [\text{ct}_q]_1 \leftarrow \text{Enc}_{\text{pk}}(0; \varrho_q); [\text{ct}_s]_1 \leftarrow \text{Enc}_{\text{pk}}(0; \varrho_s); \\ 3. [\text{ct}_\gamma]_1 \leftarrow - \begin{pmatrix} \text{Enc}_{\text{pk}}([\sigma\tau]_{1;0}) - [\text{ct}_\chi]_1 \cdot \tau - \text{Enc}_{\text{pk}}([\tau]_{1;0}) & \text{Enc}_{\text{pk}}(0;0) \\ -\text{Enc}_{\text{pk}}([\mathcal{C}_S]_{1;0}) & [\text{ct}_q]_1 \\ -\text{Enc}_{\text{pk}}(1;0) & \text{Enc}_{\text{pk}}(0;0) \end{pmatrix} (\frac{\mathbf{e}}{[\text{ct}_s]_1}) - \text{Enc}_{\text{pk}}(\mathbf{0}; \mathbf{z}); \\ 4. \pi \leftarrow ([\text{ct}_q, \text{ct}_s, \text{ct}_\gamma]_1, [\delta, \mathbf{z}]_2); \\ \text{return } \pi;$
$\text{V}(\text{crs}_{1\mathbf{p}}, \mathbf{x} = [\text{ct}_\chi]_1, \pi) : \text{mem} \leftarrow \text{NotMember}; \\ \text{if } \pi \text{ parses as } \pi = ([\text{ct}_q, \text{ct}_\gamma]_1, [\delta, \mathbf{z}]_2) \text{ then } \text{mem} \leftarrow \text{Member}; \\ \text{If } \text{mem} = \text{Member} \text{ then check} \\ 1. \bar{b}_1 \leftarrow \text{Enc}_{\text{pk}}([\sigma\tau]_{1;0}) \bullet [e]_2 - [\text{ct}_\chi]_1 \bullet [\tau e]_2 - \text{Enc}_{\text{pk}}([\tau]_{1;0}) \bullet [\delta]_2 \stackrel{?}{=} [\text{ct}_\gamma]_1 \bullet [1]_2 + [z_1]_2 \bullet \text{pk}; \\ 2. \bar{b}_2 \leftarrow -\text{Enc}_{\text{pk}}([\mathcal{C}_S]_{1;0}) \bullet [e]_2 + [\text{ct}_q]_1 \bullet [\delta]_2 \stackrel{?}{=} [\text{ct}_\gamma]_1 \bullet [1]_2 + [z_2]_2 \bullet \text{pk}; \\ 3. \text{if } \bar{b}_1 \text{ and } \bar{b}_2 \text{ then return Member else return Error}; \\ \text{else check} \\ 1. \bar{b}_1 \leftarrow \text{Enc}_{\text{pk}}([\sigma\tau]_{1;0}) \bullet [e]_2 - [\text{ct}_\chi]_1 \bullet [\tau e]_2 - \text{Enc}_{\text{pk}}([\tau]_{1;0}) \bullet [\delta]_{12} \stackrel{?}{=} [\text{ct}_\gamma]_1 \bullet [1]_2 + [z_1]_2 \bullet \text{pk}; \\ 2. \bar{b}_2 \leftarrow -\text{Enc}_{\text{pk}}([\mathcal{C}_S]_{1;0}) \bullet [e]_2 + [\text{ct}_q]_1 \bullet [\delta]_{12} - \text{Enc}_{\text{pk}}([\tau]_{1;0}) \bullet [\delta]_{22} \stackrel{?}{=} [\text{ct}_\gamma]_1 \bullet [1]_2 + [z_2]_2 \bullet \text{pk}; \\ 3. \bar{b}_3 \leftarrow -\text{Enc}(1;0) \bullet [e]_2 + [\text{ct}_s]_1 \bullet [\delta]_{22} \stackrel{?}{=} [\text{ct}_\gamma]_1 \bullet [1]_2 + [z_3]_2 \bullet \text{pk}; \\ 4. \text{if } \bar{b}_1 \text{ and } \bar{b}_2 \text{ and } \bar{b}_3 \text{ then return NotMember else return Error};$

Fig. 8. The new set (non-)membership NIZK Π^* .

and thus the plaintext part is correct. On the other hand, randomizers are correct by definition.

Similarly, $b_2 = \text{true}$ iff

$$-\text{Enc}_{\text{pk}}(\mathbf{Z}_S(\sigma); 0)\tau\mathbf{e} + [\text{ct}_q]_1\delta = ? [\text{ct}_{\gamma_2}]_1 + z_2 \cdot \text{pk}$$

\iff

$$\text{Enc}(-\mathbf{Z}_S(\sigma)\tau\mathbf{e} + \mathbf{q}\delta; \varrho_q\delta) = ? \text{Enc}(\gamma_2; \varrho_{\gamma_2}) + \text{Enc}_{\text{pk}}(0; z_2) .$$

Consider first the plaintexts. Clearly,

$$-\mathbf{Z}_S(\sigma)\tau\mathbf{e} + \mathbf{q}\delta = -\mathbf{Z}_S(\sigma)\tau\mathbf{e} + \mathbf{q} \cdot ((\sigma - \chi)\mathbf{e} - \varrho\delta) = -\varrho\delta\mathbf{q} = \gamma_2 .$$

On the other hand, randomizers are correct by definition.

Non-Membership Argument. Clearly, $\bar{b}_1 = \text{true}$ iff

$$\text{Enc}_{\text{pk}}([\sigma]_1; 0)\tau\mathbf{e} - [\text{ct}_\chi]_1\tau\mathbf{e} - \text{Enc}_{\text{pk}}([\tau]_1; 0)\delta_1 = ? [\text{ct}_{\gamma_1}]_1 + z_1 \cdot \text{pk}$$

\iff

$$\text{Enc}_{\text{pk}}((\sigma - \chi)\tau\mathbf{e} - \tau\delta_1; \varrho_\chi\tau\mathbf{e}) = ? \text{Enc}_{\text{pk}}(\gamma_1; \varrho_{\gamma_1}) + \text{Enc}_{\text{pk}}(0; z_1) .$$

Consider first the plaintexts. Clearly,

$$(\sigma - \chi)\tau\mathbf{e} - \tau\delta_1 = y_1\tau = \gamma_1\tau .$$

On the other hand, randomizers are correct by definition.

Similarly, $\bar{b}_2 = \text{true}$ iff

$$-\text{Enc}_{\text{pk}}(\mathbf{Z}_S(\sigma); 0)\tau\mathbf{e} + [\text{ct}_q]_1\delta_1 - \text{Enc}_{\text{pk}}(1; 0)\tau\delta_2 = ? [\text{ct}_{\gamma_2}]_1 + z_2 \cdot \text{pk}$$

\iff

$$\text{Enc}_{\text{pk}}(-\mathbf{Z}_S(\sigma)\tau\mathbf{e} + \mathbf{q}\delta_1 - \tau\delta_2; \varrho_q\delta_1) = ? \text{Enc}_{\text{pk}}(\gamma_2; \varrho_{\gamma_2}) + \text{Enc}_{\text{pk}}(0; z_2) .$$

Consider first the plaintexts. Clearly,

$$\begin{aligned} -\mathbf{Z}_S(\sigma)\tau\mathbf{e} + \mathbf{q}\delta_1 - \tau\delta_2 &= -\mathbf{Z}_S(\sigma)\tau\mathbf{e} + \mathbf{q} \cdot ((\sigma - \chi)\mathbf{e} - y_1) - \tau(1/s \cdot \mathbf{e} - y_2) \\ &= -y_1\mathbf{q} + y_2\tau = \gamma_2 . \end{aligned}$$

On the other hand, randomizers are correct by definition.

Finally, $\bar{b}_3 = \text{true}$ iff

$$-\text{Enc}(1; 0)\mathbf{e} + \text{ct}_s\delta_2 = ? [\text{ct}_{\gamma_3}]_1 + z_3 \cdot \text{pk}$$

\iff

$$\text{Enc}_{\text{pk}}(-\mathbf{e} + s\delta_2; \varrho_s\delta_2) = ? \text{Enc}_{\text{pk}}(\gamma_3; \varrho_{\gamma_3}) + \text{Enc}_{\text{pk}}(0; z_3) .$$

Consider first the plaintexts. Clearly,

$$-\mathbf{e} + s\delta_2 = -\mathbf{e} + s(1/s \cdot \mathbf{e} - y_2) = -y_2s = \gamma_3 .$$

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

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

Fig. 9. Reduction \mathcal{B}_{cr} in the soundness proof of Π^*

On the other hand, randomizers are correct by definition.

Computational zero-knowledge: First, consider the membership argument. Fix any λ , and let $(\text{crs}, \text{td}) \in \text{Supp}(\mathcal{K}_{\text{crs}}(1^\lambda))$. Let $\text{lp} = (\text{pk}, \mathcal{S})$ and $(\mathbf{x}, \mathbf{w}) \in \mathcal{R}_{\text{lp}}$. To show zero-knowledge we first define an hybrid simulator Sim_H . The hybrid Sim_H receives as additional input an Elgamal ciphertext $[\text{ct}_{\mathbf{q}}]_1$, that is an encryption of $[\mathbf{q}]_1$ such that $\mathbf{q}(\sigma - \chi) = \mathbf{Z}_{\mathcal{S}}(\sigma)$, where $[\chi]_1 = \text{Dec}_{\text{sk}}([\text{ct}_{\chi}]_1)$. Then Sim_H computes its output as the simulator in Fig. 8, except that it computes $[\text{ct}_{\mathbf{q}}]_1$ as an encryption of $[\mathbf{q}]_1$ 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 prover in [CLPØ21]. For completeness, we state a proof for this concrete case. In the honest prover's algorithm, since \mathbf{g}_{γ} is uniformly random, then also \mathbf{z} is uniformly random. As in Fact 1, δ output by an honest prover is uniformly random. On the other hand, Sim_H also samples uniformly random δ and \mathbf{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. 8) 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. \square

Theorem 4. *Let $\ell = 2$ and $k = 1$. Let \mathcal{D}_k be the distribution of $[\frac{1}{\mathbf{e}}]_2$ for $\mathbf{e} \leftarrow_{\mathcal{S}} \mathbb{Z}_p$. Let $q = \text{poly}(\lambda)$ be an upper bound on $|\mathcal{S}|$. The set membership NIZK Π^* in Fig. 8 is sound, assuming AC^* is $[\cdot]_1$ -collision-resistant.*

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

Decrypting all verification equations, the verifier checks guarantee that the AC^* verifier accepts $[\chi]_1 \leftarrow \text{Dec}_{\text{sk}}(\text{ct}_{\chi})$. Essentially, the constructed adversary \mathcal{B}_{cr} (see Fig. 9), on its input, creates a new Elgamal key-pair. Based on that,

\mathcal{B}_{cr} then creates a correct crs_{1p} for \mathcal{A}_{Π^*} . After obtaining (\mathbb{x}, π) from \mathcal{A}_{Π^*} , \mathcal{B}_{cr} decrypts \mathcal{A}_{Π^*} '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}_{Π^*} succeeds. \square

Efficiency. Π^* 's CRS length is $q + 1$ elements of \mathbb{G}_1 and 4 elements of \mathbb{G}_2 . The set membership argument length is $6\mathbf{g}_1 + 3\mathbf{g}_2$, which comes close to the $\Pi_{\text{clp}\emptyset}$ argument length $4\mathbf{g}_1 + 3\mathbf{g}_2$ for the simple OR language (this corresponds to $\ell = 2$). The difference comes from the fact that here we also 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. P precomputes $f(X)$ ($\Theta(|\mathcal{S}|)$ field operations) and $[\mathbf{q}]_1$ ($|\mathcal{S}|$ scalar multiplications in \mathbb{G}_1). In online computation,

- (1) the membership prover computes 8 scalar multiplications in \mathbb{G}_1 and 6 in \mathbb{G}_2 ($2\mathbf{m}_1 + 2\mathbf{m}_2$ to compute $\text{AC}^*.\psi$ and $6\mathbf{m}_1 + 4\mathbf{m}_2$ in the rest of Π^*), and
- (2) the non-membership prover computes 14 scalar multiplications in \mathbb{G}_1 and 10 in \mathbb{G}_2 ($4\mathbf{m}_1 + 4\mathbf{m}_2$ to compute $\text{AC}^*.\psi$ and $10\mathbf{m}_1 + 6\mathbf{m}_2$ in the rest of Π^*).

(The online computation includes the computation of $[\text{ct}_{\mathbf{q}}]_1$ and other ciphertexts.) The batched membership (resp., non-membership) verifier's computation is dominated by five (resp., six) pairings. Pairings with $[\mathbf{e}]_2$ can be precomputed. (This is replaced with some \mathbb{G}_T exponentiations, so the benefit depends on the implementation.) Online, the verify has to compute four and five pairings, respectively. See Appendix D.1 for the batched verifier. We refer to Table 1 for an efficiency comparison. In Appendix D.2, we compare our construction to [VB22], the most efficient random-oracle based solution.

7 On Handling Group Elements with CLP \emptyset

The CLP \emptyset NIZK [CLP \emptyset 21] works assuming the prover knows all the DR elements as integers. This seems to exclude applications where one needs to prove statements about group elements. In Π^* , we overcome this issue by making the following observation. Consider the case of a single DR $\mathbf{C}(\mathbf{X}) = (h(\mathbf{X})\|T(\mathbf{X}))$, where $h(\mathbf{X})$ is a column vector. Then, for CLP \emptyset to work, it suffices that the prover (1) knows $[\mathbf{C}(\chi)]_1$, and (2) can compute $[\delta]_2$; for this, it suffices to compute $[\mathbf{w}\mathbf{e}]_2$, where \mathbf{w} is such that $h(\mathbf{X}) = T(\mathbf{X})\mathbf{w}$ (this follows from CLP \emptyset 's construction).

In the case of Π^* , (1) means that the prover must be able to compute $[\mathbf{q}, \mathbf{Z}_{\mathcal{S}}(\sigma), \mathbf{s}]_1$ (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 $[\sigma\mathbf{e}, \mathbf{e}]_2$ 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. doi:10.1007/s00145-014-9196-7. 1, 1, 4.1, 2, 4.1, 4.1
- 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. doi:10.1007/978-3-030-45374-9_20. 4.1, 4
- 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. doi:10.1007/978-3-642-19379-8_26. 1, 1, 1, 1, 1, 1, 1, A.2
- 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. doi:10.1007/978-3-642-00862-7_20. 2.1, 3
- 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. 3, 3
- 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. doi:10.1007/978-3-662-64322-8_19. 1
- BCKL07. Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. Non-interactive anonymous credentials. Cryptology ePrint Archive, Report 2007/384, 2007. <https://eprint.iacr.org/2007/384>. A.2
- 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. doi:10.1007/978-3-540-78524-8_20. 1, 1, 1, 1, 1, 5, A.2
- 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. doi:10.1007/978-3-319-16715-2_8. 1, 3, 3
- 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. 1, 2.1
- 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. doi:10.1007/978-3-319-29485-8_8. 1, 3, 3

- 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. doi:10.1145/352600.352604. 1, 1, 2.1
- 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. 1, 2.1
- 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. 1, 1
- 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. doi:10.1007/978-3-540-89255-7_15. 1
- 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. doi:10.1007/978-3-030-56877-1_27. 1, 1, 2, 4.1, 5
- 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. doi:10.1007/978-3-030-92078-4_5. 1, 1, 2, 2.1, 2.2, 2.2, 2.2, 2.2, 2.2, 4, 4.1, 4.1, 1, 5, 6, 7
- 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. doi:10.1007/978-3-030-17253-4_11. 1, 1, 1, 1, A.2
- 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>. 1, 2.1
- 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. doi:10.1007/978-3-319-96881-0_2. 1, 2
- GK16. Shafi Goldwasser and Yael Tauman Kalai. Cryptographic assumptions: A position paper. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 505–522. Springer, Heidelberg, January 2016. doi:10.1007/978-3-662-49096-9_21. 1, A.1
- 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. doi:10.1007/978-3-319-96878-0_24. 1, 2.1
- 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. doi:10.1007/978-3-540-78967-3_24. 1, 1, 1

- GW11. Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011. doi:10.1145/1993636.1993651. A.1
- 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. doi:10.1109/SFCS.2000.892118. 2
- 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. doi:10.1007/3-540-45465-9_22. 2
- Lip12a. Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 169–189. Springer, Heidelberg, March 2012. doi:10.1007/978-3-642-28914-9_10. 1, 2
- Lip12b. 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. doi:10.1007/978-3-642-31284-7_14. 1
- 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. doi:10.1007/978-3-540-72738-5_17. 1, 2.1
- LP23. Helger Lipmaa and Roberto Parisella. Set (Non-)Membership NIZKs from Determinantal Accumulators. In Mehdi Tibouchi and Abdelrahman Aly, editors, *LATINCRYPT 2023*, volume ? of *LNCS*, pages ?–?, Quito, Ecuador, October 4–6, 2023. Springer, Cham. *
- LSZ22. Helger Lipmaa, Janno Siim, and Michal Zajac. Counting vampires: From univariate sumcheck to updatable ZK-SNARK. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part II*, volume 13792 of *LNCS*, pages 249–278. Springer, Heidelberg, December 2022. doi:10.1007/978-3-031-22966-4_9. 1
- Nao03. Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, Heidelberg, August 2003. doi:10.1007/978-3-540-45146-4_6. 1, A.1
- 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. doi:10.1007/978-3-540-30574-3_19. 1, 1, 2.1
- Sta08. Grzegorz Stachowiak. Proofs of knowledge with several challenge values. Cryptology ePrint Archive, Report 2008/181, 2008. <https://eprint.iacr.org/2008/181>. 2
- VB22. Giuseppe Vitto and Alex Biryukov. Dynamic universal accumulator with batch update over bilinear groups. In Steven D. Galbraith, editor, *CT-RSA 2022*, volume 13161 of *LNCS*, pages 395–426. Springer, Heidelberg, March 2022. doi:10.1007/978-3-030-95312-6_17. 1, 6, D.2

- Zha22. Mark Zhandry. To label, or not to label (in generic groups). In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 66–96. Springer, Heidelberg, August 2022. doi: 10.1007/978-3-031-15982-4_3. 1, A.1
- ZZK22. Cong Zhang, Hong-Sheng Zhou, and Jonathan Katz. An analysis of the algebraic group model. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part IV*, volume 13794 of *LNCS*, pages 310–322. Springer, Heidelberg, December 2022. doi:10.1007/978-3-031-22972-5_11. 1, A.1

A Preliminaries

A.1 Discussion on Falsifiable Assumptions

Goldwasser and Kalai [GK16] see constructions based on such models and assumptions as intermediate results towards removing such artificial constraints and constructing provably secure schemes under computational assumptions. While we do not 100% agree with this opinion (many important primitives like adaptively sound zk-SNARKs cannot be constructed under falsifiable assumptions [GW11]), we emphasize the importance of constructing efficient primitives under computational assumptions. Paraphrasing Naor [Nao03], a proof of security under a cryptographic assumption is a proof of the form “the construction is secure *or* the assumption is false”. Relying on falsifiable assumptions means that it is easier to check the veracity of the second clause in the previous sentence. Moreover, a long line of papers has shown that the random oracle model, the GGM, and the AGM [Zha22,ZZK22] have concrete problems.

A.2 Security Assumptions

Next, we enlist some of the security assumptions used in the previous set membership NIZKs. The BCKL [BCKL08] set membership NIZK is secure under the TDH, HSDH, and SXDH assumptions (formally defined in the full version, [BCKL07]). The DGPRS-GS [DGP⁺19] set membership NIZK is secure under the SXDH and the GSDH assumptions; the DGPRS-QA [DGP⁺19] set membership NIZK relies additionally on the QTSDH and KerMDH assumptions. The AN [AN11] set membership NIZK is secure under the SXDH and the ESDH assumptions. We refer the original papers to more discussion on all assumptions. In the following, we only list the least standard assumptions (this excludes, say, SXDH and KerMDH).

The q -TDH (*triple DH*, [BCKL07]) assumption in \mathbb{G}_ℓ holds relative to Pgen if for every λ and PPT \mathcal{A} ,

$$\Pr \left[\begin{array}{l} (\exists \mu \neq 0) \\ (A, B, C) = (\mu\sigma, \mu\tau, \mu\sigma\tau) \end{array} \middle| \begin{array}{l} \mathbf{p} \leftarrow \text{Pgen}(1^\lambda); \sigma, \tau, c_1, \dots, c_q \leftarrow_{\$} \mathbb{F}; \\ \mathbf{ck} \leftarrow (\{c_i\}_{i=1}^q, \{\frac{1}{\sigma+c_i}\}_{i=1}^q, \sigma, \tau)_1, [\sigma, \tau]_2; \\ ([A]_2, [B, C]_1) \leftarrow \mathcal{A}(\mathbf{p}, \mathbf{ck}) \end{array} \right] \approx_\lambda 0 .$$

The q -*HSDH* (*hidden SDH*, [BCKL07]) assumption in \mathbb{G}_ι holds relative to Pgen , if for every λ and PPT \mathcal{A} ,

$$\Pr \left[\begin{array}{l} (\exists c \notin \{c_i\}) \\ (A, B, C) = (\frac{1}{\sigma+c}, c, c\tau) \end{array} \middle| \begin{array}{l} \mathbf{p} \leftarrow \text{Pgen}(1^\lambda); \sigma, \tau, c_1, \dots, c_q \leftarrow_{\mathfrak{s}} \mathbb{F}; \\ \mathbf{ck} \leftarrow (\{\{\frac{1}{\sigma+c_i}, c_i, c_i\tau\}_{i=1}^q, \sigma, \tau\}_1, [\sigma]_2); \\ ([A]_1, [B]_2, [C]_1) \leftarrow \mathcal{A}(\mathbf{p}, \mathbf{ck}) \end{array} \right] \approx_\lambda 0 .$$

Let $\mathcal{Z} \subset \mathbb{Z}_p$ be some set with $|\mathcal{Z}| = q$. The \mathcal{Z} -*GSDH* (*Group SDH*, [DGP⁺19]) assumption in \mathbb{G}_ι holds relative to Pgen , if for every λ and PPT \mathcal{A} ,

$$\Pr \left[\begin{array}{l} A \notin \mathcal{Z} \wedge B = A\tau \wedge \\ C(\sigma - A) = \prod_{z \in \mathcal{Z}} (\sigma - z) \end{array} \middle| \begin{array}{l} \mathbf{p} \leftarrow \text{Pgen}(1^\lambda); \sigma, \tau \leftarrow_{\mathfrak{s}} \mathbb{F}; \\ \mathbf{ck} \leftarrow (\{\{\sigma^i\}_{i=1}^q, \tau\}_1, [\{\sigma^i\}_{i=1}^q, \tau]_2); \\ ([A]_1, [B]_\iota, [C]_2) \leftarrow \mathcal{A}(\mathbf{p}, \mathbf{ck}) \end{array} \right] \approx_\lambda 0 .$$

The q -*STSDH* (*square target SDH*, [DGP⁺19]) assumption in \mathbb{G}_ι holds relative to Pgen , if for every λ and PPT \mathcal{A} ,

$$\Pr \left[\begin{array}{l} A \notin \{-1, 1\} \wedge B = A\tau \wedge \\ C(\sigma - c) = A^2 - 1 \end{array} \middle| \begin{array}{l} \mathbf{p} \leftarrow \text{Pgen}(1^\lambda); \sigma, \tau \leftarrow_{\mathfrak{s}} \mathbb{F}; \\ \mathbf{ck} \leftarrow (\{\{\sigma^i\}_{i=1}^q, 1\}_1, [\{\sigma^i\}_{i=1}^q, \tau]_2); \\ (c, [A]_1, [B]_2, [C]_T) \leftarrow \mathcal{A}(\mathbf{p}, \mathbf{ck}) \end{array} \right] \approx_\lambda 0 .$$

The q -*ESDH* (*extended SDH*, [AN11]) assumption in \mathbb{G}_ι holds relative to Pgen , if for every λ and PPT \mathcal{A} ,

$$\Pr \left[\begin{array}{l} B \neq 0 \wedge \\ A = B(\sigma + c) \wedge C = B\tau \end{array} \middle| \begin{array}{l} \mathbf{p} \leftarrow \text{Pgen}(1^\lambda); \sigma, \tau \leftarrow_{\mathfrak{s}} \mathbb{F}^*; \\ \mathbf{ck} \leftarrow (\{\{\sigma^i\}_{i=1}^{q+1}, \tau\}_1, [1, \sigma]_2); \\ (c, [A]_1, [B]_2, [C]_1) \leftarrow \mathcal{A}(\mathbf{p}, \mathbf{ck}) \end{array} \right] \approx_\lambda 0 .$$

B Supplementary Materials to Section 4

C Missing Proofs

C.1 Proof of Theorem 2

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

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

$$\begin{aligned} \chi(\mathbf{X}) &= \chi_1(\Sigma)\mathbf{T} + \chi_2 , & \mathbf{q}(\mathbf{X}) &= \mathbf{q}_1(\Sigma)\mathbf{T} + \mathbf{q}_2 , \\ \mathbf{s}(\mathbf{X}) &= \mathbf{s}_1(\Sigma)\mathbf{T} + \mathbf{s}_2 , & \gamma_1(\mathbf{X}) &= \gamma_{11}(\Sigma)\mathbf{T} + \gamma_{12} , \\ \gamma_2(\mathbf{X}) &= \gamma_{21}(\Sigma)\mathbf{T} + \gamma_{22} , & \gamma_3(\mathbf{X}) &= \gamma_{31}(\Sigma)\mathbf{T} + \gamma_{32} , \end{aligned}$$

$$\delta_1(\mathbf{X}) = \delta_{11} + \delta_{12}\mathbf{E} + \delta_{13}\Sigma\mathbf{E} + \delta_{14}\mathbf{TE} \quad , \quad \delta_2(\mathbf{X}) = \delta_{21} + \delta_{22}\mathbf{E} + \delta_{23}\Sigma\mathbf{E} + \delta_{24}\mathbf{TE} \quad ,$$

where each polynomial (like \mathbf{q}_1) on the RHS is of degree $\leq q$. That is, the algebraic adversary \mathcal{A}_{alg} also outputs coefficients of all above polynomials.

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

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

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 $\mathcal{R}[\mathbf{T}, \mathbf{E}]$, where $\mathcal{R} = \mathbb{Z}_p[\Sigma]$. We now enlist the non-zero coefficients of all monomials $\mathbf{T}^i\mathbf{E}^j$ of all polynomials, highlighting the coefficients that are actually needed in this proof (we give other coefficients for completeness' sake):

$(i, j) \quad V_1$	$(i, j) \quad V_3$
$(2, 1) \quad -\delta_{14} - \chi_1(\Sigma)$	$(2, 1) \quad \delta_{24}\mathbf{s}_1(\Sigma)$
$(1, 1) \quad -\delta_{12} + (1 - \delta_{13})\Sigma - \chi_2$	$(1, 1) \quad \delta_{22}\mathbf{s}_1(\Sigma) + \delta_{23}\mathbf{s}_1(\Sigma)\Sigma + \mathbf{s}_2\delta_{24}$
$(1, 0) \quad -\delta_{11} - \gamma_{11}(\Sigma)$	$(1, 0) \quad \delta_{21}\mathbf{s}_1(\Sigma) - \gamma_{31}(\Sigma)$
$(0, 0) \quad -\gamma_{12}$	$(0, 1) \quad \mathbf{s}_2\delta_{23}\Sigma + \mathbf{s}_2\delta_{22} - 1$
$(i, j) \quad V_2$	$(0, 0) \quad \mathbf{s}_2\delta_{21} - \gamma_{32}$
$(2, 1) \quad \delta_{14}\mathbf{q}_1(\Sigma) - \delta_{24}$	
$(1, 1) \quad \delta_{14}\mathbf{q}_2 + \delta_{12}\mathbf{q}_1(\Sigma) + (\delta_{13}\mathbf{q}_1(\Sigma) - \delta_{23})\Sigma - \mathbf{Z}_S(\Sigma) - \delta_{22}$	
$(1, 0) \quad \delta_{11}\mathbf{q}_1(\Sigma) - \gamma_{21}(\Sigma) - \delta_{21}$,	
$(0, 1) \quad \delta_{12}\mathbf{q}_2 + \delta_{13}\mathbf{q}_2\Sigma$	
$(0, 0) \quad \delta_{11}\mathbf{q}_2 - \gamma_{22}$	

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

From the coefficient of $\mathbf{T}^2\mathbf{E}$ of V_1 , we get $\chi_1(\Sigma) = -\delta_{14}$. From the coefficient of \mathbf{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 :

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

Consider the coefficients of V_2 :

- \mathbf{TE} : $\mathbf{Z}_S(\Sigma) - \delta_{14}\mathbf{q}_2 + 1/\mathbf{s}_2 = (\Sigma - \chi_2)\mathbf{q}_1(\Sigma)$.
Since $\mathbf{Z}_S(\Sigma)$ is non-constant, $\mathbf{q}_1(\Sigma) \neq 0$.

– T²E: $\delta_{14}\mathbf{q}_1(\Sigma) = 0$. Since $\mathbf{q}_1(\Sigma) \neq 0$, we get $\delta_{14} = 0$.

Hence, the coefficient of TE of V_2 gives $\mathbf{Z}_S(\Sigma) + 1/s_2 = (\Sigma - \chi_2)\mathbf{q}_1(\Sigma)$. Thus, $(\Sigma - \chi_2) \mid \mathbf{Z}_S(\Sigma) + 1/s_2$. Since $\chi(\mathbf{X}) = \chi_2$, $\mathbf{Z}_S(\chi_2) = -1/s_2 \neq 0$.

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 $\mathcal{B}(\{\{\sigma^i\}_1\}_{i=0}^{q+1}, \{\{\sigma^i\}_1\}_{i=0}^2)$. \mathcal{B} samples $\alpha_1, \alpha_2, \beta_1, \beta_2 \leftarrow_{\mathbb{S}} \mathbb{Z}_p$ and sets implicitly $\tau \leftarrow \alpha_1\sigma + \beta_1$ and $\mathbf{e} \leftarrow \alpha_2\sigma + \beta_2$. Then, \mathcal{B} creates \mathbf{crs} for an DETACNM adversary \mathcal{A}_{alg} , and calls \mathcal{A}_{alg} with \mathbf{crs} . After obtaining π , together with the coefficients of the polynomials like $\chi(\Sigma)$, from $\text{Ext}_{\mathcal{A}_{\text{alg}}}$, \mathcal{B} reconstructs the coefficients of the degree- $\leq (q+2)$ polynomial V_2 (which is now univariate since τ and \mathbf{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 $q+2$ roots x_i of V_2 . \mathcal{B} tests which one of them is equal to σ , and returns σ . \square

D Supplementary Material to Section 6

D.1 Batched Verifier of Π^*

Next, we describe the batched verifier of Π^* . Here, we denote the i th component of Elgamal ciphertext ct by $\text{ct}^{\langle\langle i \rangle\rangle}$.

$V(\mathbf{crs}_{1p}, \mathbb{X} = [\text{ct}_\chi]_1, \pi) : \text{mem} \leftarrow \text{NotMember}; \eta \leftarrow_{\mathbb{S}} \mathbb{Z}_p;$
 if π parses as $\pi = ([\text{ct}_q, \mathbf{ct}_\gamma]_1, [\delta, \mathbf{z}]_2)$ then $\text{mem} \leftarrow \text{Member};$
 If $\text{mem} = \text{Member}$ then check

1. $b \leftarrow (\eta[\sigma\tau]_1 - \eta^3[\mathbf{C}_S]_1) \bullet [\mathbf{e}]_2 - ([\text{ct}_\chi^{\langle\langle 1 \rangle\rangle}]_1 + \eta[\text{ct}_\chi^{\langle\langle 2 \rangle\rangle}]_1) \bullet [\tau\mathbf{e}]_2 + (-\eta[\tau]_1 + \eta^2[\text{ct}_q^{\langle\langle 1 \rangle\rangle}]_1 + \eta^3[\text{ct}_q^{\langle\langle 2 \rangle\rangle}]_1) \bullet [\delta]_2 \stackrel{?}{=} ([\text{ct}_{\gamma_1}^{\langle\langle 1 \rangle\rangle}]_1 + \eta[\text{ct}_{\gamma_1}^{\langle\langle 2 \rangle\rangle}]_1 + \eta^2[\text{ct}_{\gamma_2}^{\langle\langle 1 \rangle\rangle}]_1 + \eta^3[\text{ct}_{\gamma_2}^{\langle\langle 2 \rangle\rangle}]_1) \bullet [1]_2 + ([1]_1 + \eta[\mathbf{sk}]_1) \bullet ([z_1]_2 + \eta^2[z_2]_2);$
2. if b then return **Member** else return **Error**;

else check

1. $\bar{b} \leftarrow (\eta[\sigma\tau]_1 - \eta^3[\mathbf{C}_S]_1 - \eta^5[1]_1) \bullet [\mathbf{e}]_2 - ([\text{ct}_\chi^{\langle\langle 1 \rangle\rangle}]_1 + \eta[\text{ct}_\chi^{\langle\langle 2 \rangle\rangle}]_1) \bullet [\tau\mathbf{e}]_2 + (-\eta[\tau]_1 + \eta^2[\text{ct}_q^{\langle\langle 1 \rangle\rangle}]_1 + \eta^3[\text{ct}_q^{\langle\langle 2 \rangle\rangle}]_1) \bullet [\delta_1]_2 + (-\eta^3[\tau]_1 + [\eta^4\text{ct}_s^{\langle\langle 1 \rangle\rangle}]_1 + \eta^5[\text{ct}_s^{\langle\langle 2 \rangle\rangle}]_1) \bullet [\delta_2]_2 \stackrel{?}{=} ([\text{ct}_{\gamma_1}^{\langle\langle 1 \rangle\rangle}]_1 + \eta[\text{ct}_{\gamma_1}^{\langle\langle 2 \rangle\rangle}]_1 + [\eta^2\text{ct}_{\gamma_2}^{\langle\langle 1 \rangle\rangle}]_1 + \eta^3[\text{ct}_{\gamma_2}^{\langle\langle 2 \rangle\rangle}]_1 + \eta^4[\text{ct}_{\gamma_3}^{\langle\langle 1 \rangle\rangle}]_1 + \eta^5[\text{ct}_{\gamma_3}^{\langle\langle 2 \rangle\rangle}]_1) \bullet [1]_2 + ([1]_1 + \eta[\mathbf{sk}]_1) \bullet ([z_1]_2 + \eta^2[z_2]_2 + \eta^4[z_3]_2);$
2. if \bar{b} then return **NotMember** else return **Error**;

D.2 Comparison to [VB22]

The most efficient random-oracle-model universal set (resp., non-)membership NIZK [VB22] has proof size of 3 (resp., 5) elements in \mathbb{G}_1 , and 6 (resp., 11) elements in \mathbb{Z}_p . The prover needs 9 (resp., 15) scalar-point multiplications in \mathbb{G}_1 , 4 (resp., 5) exponentiations in \mathbb{G}_T , and 1 pairing to compute a (resp., non-)membership proof. Lastly, the verifier checks proof with 2 pairings for a non-membership and 1 pairing for a membership. Unsurprisingly, [VB22] is mostly more efficient than our construction since it uses the random oracle model. However, surprisingly, our prover is actually faster.