



This is a repository copy of *Shuffle arguments based on subset-checking*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/219573/>

Version: Accepted Version

Proceedings Paper:

Abdolmaleki, B. orcid.org/0009-0008-8335-2787, Fauzi, P. orcid.org/0000-0001-6800-7821, Krips, T. orcid.org/0000-0003-0981-3553 et al. (1 more author) (2024) Shuffle arguments based on subset-checking. In: Galdi, C. and Phan, D.H., (eds.) Security and Cryptography for Networks. 14th International Conference, SCN 2024, 11-13 Sep 2024, Amalfi, Italy. Lecture Notes in Computer Science, 14973 . Springer Nature Switzerland , pp. 345-366. ISBN 9783031710698

https://doi.org/10.1007/978-3-031-71070-4_16

© 2024 The Authors. Except as otherwise noted, this author-accepted version of a paper published in Security and Cryptography for Networks is made available via the University of Sheffield Research Publications and Copyright Policy under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here:

<https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

Shuffle Arguments Based on Subset-Checking

Behzad Abdolmaleki¹, Prastudy Fauzi², Toomas Krips³, Janno Siim⁴

¹ University of Sheffield, UK

² Nanyang Technological University, Singapore

³ University of Tartu, Estonia

⁴ Simula UiB, Norway

Abstract. Zero-knowledge shuffle arguments are a useful tool for constructing mix-nets which enable anonymous communication. We propose a new shuffle argument using a novel technique that probabilistically checks that each weighted set of input elements corresponds to some weighted set of output elements, with weights from the same set as the input element weights. We achieve this using standard discrete log assumptions and the shortest integer solution (SIS) assumption. Our shuffle argument has prover and verifier complexity linear in the size of the shuffled set, and communication complexity logarithmic both in the shuffled set size and security parameter.

Keywords: Shuffle argument · electronic voting · zero-knowledge.

1 Introduction

Mix-networks are protocols used to hide relations between message senders and their messages. At the heart of many widely used mix-networks are shuffle arguments that prove that a set of output ciphertexts are a rerandomized permutation of given input ciphertexts. Shuffle arguments are an essential tool for preserving anonymity in cryptographic protocols such as electronic voting systems [1], anonymous messaging systems [28], anonymous cryptocurrencies [15], and single secret leader elections [7].

Shuffle arguments can be constructed in various ways, such as Neff’s permutation of roots method [24,4], permutation matrices [17], and rerandomizable CCA-secure cryptosystems [14]. Recent techniques such as Bulletproofs [10,12], Hoffman et al. [22], and Curdleproofs [27] achieve shuffle arguments with logarithmic communication complexity and linear prover and verifier complexity. However, the concrete complexity of Bulletproofs and Hoffman et al. depend on which sorting circuit is implemented, while Curdleproofs shuffles public keys $(g_i, g_i^{x_i})_{i=1}^n$ into $(g_{\pi(i)}^r, (g_{\pi(i)}^r)^{x_{\pi(i)}})_{i=1}^n$, for some randomly sampled r , and as such do not shuffle ciphertexts. See [21] for an overview of known techniques.

A recent work by Fleischhacker and Simkin [16] uses a novel technique to prove a shuffle of commitments by performing a shuffle as a composition of v shuffles, then letting the verifier open any $v - 1$ of them. The technique leads to very efficient shuffles based on mostly symmetric primitives without any setup

assumptions, but with a noticeable soundness error. While noticeable soundness error is not ideal in many situations, it can be useful in cases where detected cheating provers are heavily penalized. Moreover, their shuffle can achieve negligible soundness by repetition, although communication becomes linear in the security parameter.

A natural question to ask is whether or not one can achieve shuffle arguments under standard assumptions that are comparable to Fleischhacker and Simkin [16] if we allow noticeable soundness error, but can still have logarithmic proof size if we require negligible soundness.

1.1 Our Contributions

We propose a framework for communication-efficient shuffle arguments using a simple random subset checking method. In the basic version, the verifier chooses a random non-empty subset of input commitments (resp., ciphertexts) as a challenge, and the prover proves knowledge of a set of output commitments (resp., ciphertexts) and its randomizers that correspond to the input commitments (resp., ciphertexts). A cheating prover in our protocol will fail to answer this challenge with some noticeable probability δ , thus by repeating the protocol a small number of times, a cheating prover can only succeed with a negligible chance. We are using techniques which will make repeated applications very cheap by batching. We obtain two versions of a public-coin zero-knowledge shuffle argument either for commitments or ciphertexts using standard assumptions.

- The protocol Π_{scs}^{lite} assumes that it is hard to find any non-trivial linear relations between the input commitments. Π_{scs}^{lite} achieves excellent efficiency under the discrete logarithm (DL) assumption in the random oracle model (ROM).
- The protocol Π_{scs} is somewhat less efficient, but does not make any assumption about the input commitments. However, it additionally requires the Short Integer Solution (SIS) assumption.

We have minimal setup assumptions, as we only require a Pedersen commitment key as CRS. See Table 1 for comparisons of our scheme to some recent shuffle arguments.

1.2 Technical Overview

Many previous shuffle arguments commit to some representation of a permutation, then prove that it is indeed a permutation, and that the shuffled commitments were obtained from the original commitments by applying the permutation inside the commitment. We take a slightly different approach that does not try to include some representation that describes a permutation. The prover has to show for a randomly chosen subset of input commitments, that it knows a corresponding subset of output commitments. We will mainly focus our attention on

	Prover	Verifier	Decrypt	Communicat.	CRS size	CRS	Assumption
[10]	$O(N)$ exp.	$O(N)$ exp.	N exp.	$2N + O(\log N)$ $\times \mathbb{G}$	$2N \times \mathbb{G}$	Uniform	ROM, DL
[22]	$30N$ exp.	$10N$ exp.	N exp.	$2N + O(\log N)$ $\times \mathbb{G}$	$N \times \mathbb{G}$	Uniform	Kernel-MDH
[27]*	$30N$ exp.	$5N$ exp.	- *	$2N + 10 \log N$ $\times \mathbb{G}$	$N \times \mathbb{G}$	Uniform	DDH
[14]	$72N$ exp., $5N$ pair.	$22N$ pair.	$2N$ exp., $46N$ pair.	$12N \times \mathbb{G}_1$, $11N \times \mathbb{G}_2$, $4N \times \mathbb{G}_T$	$2m \times \mathbb{G}_1$, $2m \times \mathbb{G}_2$	Uniform	Falsifiable
[2]	$11N$ exp.	$7N$ exp., $3N$ pair.	N exp.	$4N \times \mathbb{G}_1$, $3N \times \mathbb{G}_2$	$5N \times \mathbb{G}_1$, $N \times \mathbb{G}_2$	Verifiable	AGM
Π_{scs}^{lite}	$12N\lambda + 2$ exp.	$2N + 2$ exp.	N exp.	$2N + 2 \log(N\lambda)$ $2 \times \mathbb{G}$	$(N + 1) \times \mathbb{G}$	Uniform	ROM, DL, Trusted input ciphertexts
Π_{scs}	$35N\lambda + 5N$ exp.	$26N$ exp.	N exp.	$2N + 28 \log(N\lambda/2)$ $\times \mathbb{G}$	$2 \times \mathbb{G}$	Uniform	ROM, DL, SIS

Table 1: Comparison of our shuffle argument against state-of-the-art. Exp. stands for exponentiations, pair. for pairings, N is the number of input ciphertexts, m is the number of mixers, and λ is the security parameter. Constant terms are neglected, shuffling is included to prover's efficiency, and shuffled ciphertexts are included to proof size.
 (*) Note that Curdleproofs [27] does not shuffle ciphertexts, so N depicts the number of public keys it shuffles; here, no decryption is performed.

constructing a proof of shuffle for Pedersen commitments, but later show that it can be modified to a proof of shuffle for ElGamal ciphertexts.

Subset-checking idea

Denote a Pedersen commitment to a message w by $[w] = h^r g^w$ for some randomness r and group generators g and h . Let $\{[w_j]\}_{j=1}^N$ be a set of input commitments and $\{[\hat{w}_j]\}_{j=1}^N$ a set of shuffled output commitments (i.e., there exists a permutation π and a vector \mathbf{r} of randomizers such that for each i , $[w_i] = [\hat{w}_{\pi(i)}] h^{r_{\pi(i)}}$, we also assume that the prover knows the π and \mathbf{r}).

First, we note that one can treat the commitments $\{[w_j]\}_{j=1}^N$ as a public key for a Pedersen multicommittment, assuming that nobody knows any nontrivial linear discrete logarithm relations between them. More formally, the adversary cannot efficiently find a non-zero $\mathbf{a} \in \mathbb{Z}_q^N$ such that $\prod_{j=1}^N [w_j]^{a_j} = 1$.⁵ This makes it easy to apply efficient proof techniques such as Bulletproofs to products of commitments, because products of commitments will correspond to Pedersen multicommittments in this perspective. We will expand more on this later on.

Let $I \subseteq [1, N]$ be a subset chosen randomly by the verifier and let $\mathbf{e} \in \{0, 1\}^N$ denote the characteristic vector of I . If there exists \mathbf{r} and a permutation π such

⁵ Brands [9] called it the FindRep assumption and its known to be tightly equivalent to the discrete logarithm assumption.

that $[w_i] = h^{r_{\pi(i)}}[\hat{w}_{\pi(i)}]$, then for the subset I ,

$$\prod_{i \in I} [w_i] = \prod_{i \in I} h^{r_{\pi(i)}}[\hat{w}_{\pi(i)}] = h^{\sum_{i \in I} r_{\pi(i)}} \prod_{i=1}^N [\hat{w}_{\pi(i)}]^{e_{\pi(i)}} .$$

The right-hand side of this can be considered a Pedersen multicommithment with the public key $h, [\hat{w}_1], \dots, [\hat{w}_N]$.

We will show that if the prover cheats, there exists many subsets I such that the prover does not know a randomizer r or a binary vector \mathbf{e} satisfying $\prod_{i \in I} [w_i] = h^r \prod_{j=1}^N [\hat{w}_j]^{e_j}$. We let the prover prove knowledge of r and \mathbf{e} for this relation using an aggregated range proof (e.g., Bulletproofs), which means a cheating prover will fail for some challenge subsets I . In the most basic case, as we will explain later, the prover will not know a suitable r and \mathbf{e} for at least $\frac{1}{4}$ of subsets. We can make the soundness error negligible by repeating the protocol a small number of times.

We will show that the soundness error of the basic protocol can be decreased, which reduces the number of repetitions that we need in the protocol. The strategies we exploit are as follows.

1. The prover additionally proves that the size of I is equal to $\sum_j e_j$, which will increase the number of challenge subsets for which a cheating prover will fail.
2. Instead of having $\mathbf{e} \in \{0, 1\}^N$, we can use some larger challenge set F for which the verifier can also easily check that the messages in a multicommithment belong to this set. More precisely, the verifier can sample a random vector \mathbf{e} from F and the prover shows knowledge of how to open $\prod_{i=1}^N [w_i]^{e_i}$ in the basis $h, [\hat{w}_1], \dots, [\hat{w}_N]$ such that every element of the message vector in that multicommithment is a member of F .

One can also flip the roles of $\{[w_i]\}_{i=1}^N$ and $\{[\hat{w}_j]\}_{j=1}^N$ and obtain the symmetrical case. In fact, we will proceed with the flipped case, due to it being more realistic that no non-trivial DL relations (or DLREs, as we will denote them) are known between the $\{[w_i]\}_{i=1}^N$. In a mix-net, a mixer (a server that shuffles the ciphertexts) receives $\{[w_i]\}_{i=1}^N$ from the previous mixer and produces suffled $\{[\hat{w}_j]\}_{j=1}^N$ and a proof of the shuffle. If the previous mixer does not collude with the current mixer, then the current mixer does not know any non-trivial DL relations for $\{[w_i]\}_{i=1}^N$. The same is not true for $\{[\hat{w}_j]\}_{j=1}^N$, which the mixer itself generates.

In the basic subset-checking shuffle Π_{scs}^{lite} , we assume that the prover knows no nontrivial DL Relations between the input.⁶ If a prover is able to pass with at least a certain non-negligible probability, we will be able to extract values $d_{i,j}$ such that $[w_i] = h^{r_i} \prod_{j=1}^N [\hat{w}_j]^{d_{i,j}}$. We will then show that all responses have to be consistent with these values and that if $(d_{i,j})_{i,j=1}^N$ is not a permutation matrix, the success rate of the prover is negligible.

⁶ If non-trivial DLREs are known between the input elements, there are attacks that succeed with overwhelming probability.

The full subset-checking shuffle Π_{scs}

For the more realistic case, we may face an adversary that knows DL relations between the input elements. To overcome such an adversary, our general strategy is as follows.

1. Use range proofs to ensure that with overwhelming probability, the known DL relation is linear in nature, with small constants in the linear equation. We will call such DL relations *small*. (Note that the range proofs serve a dual purpose, as they are additionally used for the main argument to show that the elements of the message vector are members of the challenge set. Thus, in a sense, we get this first proof "for free".)
2. Ask the prover to pass the basic subset checking argument for various rerandomizations $\{h^{a_i}[w_i]\}_{i=1}^N$ instead of $\{[w_i]\}_{i=1}^N$, where the values $\{a_i\}$ are chosen by the verifier. If the verifier accepts, then either the extractor can extract a valid permutation and randomness values for the shuffle, or the prover knows *small* DL relations for all such given rerandomizations of $\{[w_i]\}_{i=1}^N$.
3. Show that a prover that knows *small* DL relations for all given rerandomization values will break a variant of the SIS assumption.

Firstly, note that not all known non-trivial DLRELS will be useful to a cheating prover. Due to a clever use of range proofs, the only useful DLRELS for the prover are those of the form $\prod [w_i]^{a_i} = 1$, where all the a_i are smaller than a certain bound. Intuitively, the basic attack vector that arises from known DLRELS is that the cheating prover can use one or more shuffled elements to play the part of another, and hence substitute elements into the output shuffle undetected.

For example, suppose $N = 3$ and suppose the prover knows that $[w_1][w_2]^2 = [w_3]$. Then a cheating prover can set $[\hat{w}_1] = [w_1]$, $[\hat{w}_2] = [w_2]$ and $[\hat{w}_3] = [0]$. Suppose that we are using the set $F = [0, 10]$ for the range proof. Then, when the verifier sends a challenge $I = \{1, 2, 3\}$ (i.e., asks for the value corresponding to $[w_1][w_2][w_3]$ in the basis $[\hat{w}_1], [\hat{w}_2], [\hat{w}_3]$), the prover can say that $[w_1][w_2][w_3] = [\hat{w}_1]^2[\hat{w}_2]^3[\hat{w}_3]^0$ and pass verification since $2, 3, 0 \in [0, 10]$, and thus the range check passes. However, if the adversary instead knew a relation of the form $[w_1][w_2]^{100} = [w_3]$, then the above attack would not work since the analogous resulting equation would be $[w_1][w_2][w_3] = [\hat{w}_1]^2[\hat{w}_2]^{101}[\hat{w}_3]^0$ and $101 \notin [0, 10]$. Consequently, the corrected scheme will work even if the cheating prover is allowed to know some nontrivial DL relations between the input elements, provided that at least one of the elements describing the DL relation is large enough (i.e., not in F) so a range proof will fail.

Secondly, note that the scheme we have currently described, has not paid attention to the exponent of h . However, in a correct challenge-response, the exponent of the h must have a specific value. More specifically, we will let the prover provide v commitments $\{C_k\}_{k=1}^v$ to all the randomizers $\{r_i\}$. For each response to the challenge, the prover must additionally show that her response is consistent to these commitments $\{C_k\}_{k=1}^v$. In essence, we will be able to extract a matrix $(d_{i,j})_{i,j=1}^N$ whose coefficients will satisfy $[w_i] = \prod_{j=1}^N \left(\frac{[\hat{w}_j]}{h^{r_j}}\right)^{d_{i,j}}$ where the

r'_j must be consistent with the commitment of rerandomization factors $\{C_k\}_{k=1}^v$. Since the commitment is binding, we can conclude that $r'_j = r_j$ for $1 \leq j \leq N$.

Hence, a cheating prover must not only know a non-trivial DLREL between the messages, but also a non-trivial DLREL between the commitments that can be denoted by some matrix $(d_{i,j})_{i,j=1}^N$. Moreover, the matrix d must contain small elements. Roughly speaking, this is because in order to do the same attack, the same DLRELS that hold between the messages must now also hold between the h^{r_i} . A more precise reasoning will be given in the security proofs. We can use rerandomization with public rerandomization factors to destroy small non-trivial discrete logarithm relations between the commitments. That is, we would publicly randomly sample a_1, \dots, a_N , denote $[w_i]' \leftarrow h^{a_i}[w_i]$ for all i , and do the shuffle proof between the $\{[w_i]'\}_{i=1}^N$ and the $\{[\hat{w}_j]'\}_{j=1}^N$. After all, as the honest prover knows these a_i , she can as well do the new proof as the old proof. However, as the values a_1, \dots, a_N are sent after the adversary has chosen $\{[w_i]\}_{i=1}^N$ and $\{[\hat{w}_i]\}_{i=1}^N$, the adversary's advantage will significantly decrease.

However, the question of how good one public rerandomization is in getting rid of small non-trivial DLRELS, is tricky. Here the rerandomization factors will have to be still known to the prover. Thus, there will still be many nontrivial DLRELS that might be known to the prover if she happens to know some on the original commitments.⁷

Fortunately, we will see that the vast majority of these DLRELS will have at least some coefficients that are large which will cause them to not be usable as attack vectors, due to reasons that were roughly explained above and will be explained more precisely later on. The small amount of DLRELS that might be usable, turn out to be hard to find due to lattice assumptions. However, the precise values of “too large” and what the lattice assumption gives us, is dependent on a number of parameters. Thus it might happen that if we rerandomize only once, the adversary might still plausibly find medium-size DLRELS that are useful for breaking the scheme. Thus it might be necessary to use several public rerandomizations, say v times, with rerandomization values $\{a_{i,k}\}_{i=1,k=1}^{N,v}$. The challenge I will be the same for all the rerandomizations and the prover has to answer them all in a consistent way. Suppose now that the cheating prover will be able to come up with a small discrete logarithm relation characterized by vector $\{c_i\}_{i=1}^N$. To cheat, the $\{c_i\}_{i=1}^N$ has to be a DLREL in all the rerandomizations, that is, for all $k \in [1, v]$, we would have that $\prod_{i=1}^N (h^{a_{i,k}}[w_i])^{c_i} = 1$.

This turns out to be equivalent to \mathbf{c} being a vector with small coefficients that gives a scalar product of 0 with v random vectors, which we show to be equivalent to solving a SIS-problem over an exponentially large field. Note that the soundness of a single test will depend on what set F we can use for the range proof and on the number of rerandomizations v that we do. However, since the size of the group \mathbb{G} is a part of the SIS assumption, just one rerandomization will be sufficient as long as \mathbb{G} is large enough.

⁷ Essentially finding a DLREL that holds both between the original commitments and the rerandomized ones, is related to solving a linear equation with N variables with two equations.

2 Preliminaries

Let λ denote the statistical security parameter. Let \mathbb{F} denote a finite field. We denote uniform sampling from a set S by $x \leftarrow S$. Let $[1..N]$ denote $\{1, \dots, N\}$. We write $f(\lambda) \leq \text{negl}(\lambda)$ if f is a negligible function in λ . By $y \leftarrow \mathcal{A}(x; r)$ we denote that an algorithm \mathcal{A} takes an input x and uses random coins r to produce an output y . We denote by \mathbb{S}_N the set of permutations on N elements. For a predicate P , let $\exists^n x P(x)$ denote that there exists exactly n distinct values x such that $P(x)$ holds.

We denote a commitment of x by $[x]$, and a commitment of x with explicit randomness r by $[x; r]$. We use Pedersen multicommitsments, where $[x_1, \dots, x_N; r] = h^r \prod_{i=1}^N g_i^{x_i}$. For a shuffle argument, we assume that we have two committed vectors $\{[w_1], \dots, [w_N]\}$ and $\{[\hat{w}_1], \dots, [\hat{w}_N]\}$ so that the second vector is a permutation-and-rerandomization of the first. We assume that the prover knows the permutation π and the rerandomization vector \mathbf{r} . Thus, in the case of Pedersen multicommitsments, $[w_i] = [\hat{w}_{\pi(i)}] h^{r_{\pi(i)}}$ for $i \in [1..N]$.

Notation: for an element a of \mathbb{Z}_q , we denote with $|a|$ the non-negative integer that measures the distance of this element from 0, that is, if $a \in [0, \frac{q}{2})$, $|a| = a$, if $a \in [-\frac{q}{2}, 0)$, then $|a| = -a$. For a vector $\mathbf{a} = \{a_i\}_{i=1}^N$, we denote with $|\mathbf{a}| := \max_i(|a_i|)$. For any set A , and an element k , we denote by $k+A := \{k+a \mid a \in A\}$ and $k \cdot A := \{k \cdot a \mid a \in A\}$. Also $A - A := \{a_1 - a_2 \mid a_1, a_2 \in A\}$. Let $P_{p,f} = \{a \in \mathbb{Z}_q \mid \exists a_0, \dots, a_f \in \{0, 1\}, a = \sum_{i=0}^f a_i p^i\}$ where p is an integer and $3 < p < \sqrt[2]{q}$.

For a D that is a $n \times n$ square matrix with elements in \mathbb{Z}_q with rows $D^{(1)}, \dots, D^{(n)}$ and $E \subset \mathbb{Z}_q$, we denote

$$p_{D,E} := \Pr[(D^{(i)}, \mathbf{b}) \in E, \forall i \in [1, n] \mid b_1, b_2, \dots, b_n \xleftarrow{\$} E, \mathbf{b} = (b_1, \dots, b_n)].$$

We defer to Appendix A for more definitions of basic concepts.

2.1 Zero-Knowledge Argument

Let Pgen be a PPT parameter generation algorithm that on input 1^λ outputs \mathbf{p} (e.g., a description of the group or some other setup parameters). A zero-knowledge argument of knowledge for a relation \mathcal{R} is a tuple of efficient algorithms $(\text{Pgen}, \text{P}, \text{V})$ that satisfies properties of perfect completeness, computational witness-extended emulation, and special honest verifier zero-knowledge, defined in Appendix A.2. The prover algorithm P and verifier algorithm V are interactive algorithms and we denote their protocol transcript by $\text{tr} \leftarrow \langle \text{P}(\mathbf{p}, \mathbf{x}, \mathbf{w}), \text{V}(\mathbf{x}) \rangle$ where $(\mathbf{x}, \mathbf{w}) \in \mathcal{R}$ and $\mathbf{p} \in \text{Pgen}(1^\lambda)$. We write $\langle \text{P}(\mathbf{p}, \mathbf{x}, \mathbf{w}), \text{V}(\mathbf{x}) \rangle = 1$ to denote that verifier outputs 1 at the end of the interaction. Interactions with an adversary follow a similar notation. See Appendix A.2 for more detail of definitions related to zero-knowledge arguments of knowledge.

Our argument becomes non-interactive zero-knowledge when applying the Fiat-Shamir transform [6].

2.2 Pedersen Multicommitment

Let \mathbb{G} be a cyclic multiplicative group of prime order p where the discrete logarithm assumption holds. A Pedersen multicommitment, also known as Extended Pedersen commitment, consists of a key generation \mathbf{KGen} and a commitment algorithm \mathbf{Com} . Key generation algorithm $\mathbf{KGen}(1^\lambda, n)$ samples $h, g_1, \dots, g_n \leftarrow \mathbb{G}$ and outputs $\mathbf{ck} = (h, g_1, \dots, g_n)$. The commitment algorithm $\mathbf{Com}_{\mathbf{ck}}(\mathbf{m}; r)$ takes in a message $\mathbf{m} \in \mathbb{Z}_p^n$ and a randomness $r \leftarrow \mathbb{Z}_p$ and outputs a commitment $c = h^r g_1^{m_1} \dots g_n^{m_n}$. The commitment is opened by revealing \mathbf{m} and r which allows to verify the commitment.

We obtain the standard Pedersen commitment if $n = 1$. Pedersen multicommitment is perfectly hiding, i.e., \mathbf{m} is information-theoretically hidden, and if the discrete logarithm assumption holds in \mathbb{G} it is also binding, i.e., an efficient adversary cannot open a commitment to two different values.

2.3 Assumptions

Let \mathbf{GGen} be an algorithm that takes as an input the security parameter 1^λ and outputs a multiplicative group \mathbb{G} .

Definition 1 (NoDLRel [10]). *We say that the N -NoDLRel assumption holds respect to \mathbf{GGen} if for any PPT adversary \mathcal{A} and for all $N \geq 2$ there exists a negligible function $\mu(\lambda)$ such that*

$$\Pr \left[\begin{array}{l} \mathbb{G} \leftarrow \mathbf{GGen}(1^\lambda), g_1, \dots, g_N \leftarrow \mathbb{G}, \\ \mathbf{a} = (a_1, \dots, a_N) \leftarrow \mathcal{A}(\mathbb{G}, g_1, \dots, g_N) : \mathbf{a} \neq \mathbf{0} \wedge \prod_{i=1}^N g_i^{a_i} = 1. \end{array} \right] = \mu(\lambda).$$

The above is known to reduce to the standard discrete logarithm assumption [9,10].

2.4 Proof of knowledge of Pedersen commitment opening.

It is well known how to do the proof of knowledge of a Pedersen multicommitment with a logarithmic-size proof using the compressed Sigma protocol approach [5]. The full relation is defined as follows.

$$\mathcal{R}_{\mathbf{K}o\mathbf{E}}^{\mathbf{ck}, N} = \left\{ \begin{array}{l} (x = c \in \mathbb{G}, \mathbf{w} = (e, \{e_i\}_{i=1}^N) \in \mathbb{Z}_p^{N+1}) : \\ c = h^e g_1^{e_1} \dots g_N^{e_N} \wedge \mathbf{ck} = (h, g_1, \dots, g_N) \in \mathbb{G}^{N+1} \end{array} \right\}.$$

2.5 Inner product argument

An *inner product argument* is an argument for the following relation.

$$\mathcal{R}_{\mathbf{in-prod}}^{\mathbf{ck}, N} := \left\{ \begin{array}{l} (x = (c \in \mathbb{G}, \hat{c} \in \mathbb{G}, H \in \mathbb{G}, \mathbf{b} \in \mathbb{Z}_p^N), \mathbf{w} = (\mathbf{x} \in \mathbb{Z}_p^N, \delta \in \mathbb{Z}_p)) : \\ c = h^\delta \prod_{i=1}^N g_i^{x_i} \wedge \hat{c} = H^{\sum_{i=1}^N b_i x_i}. \end{array} \right\},$$

where $\mathbf{ck} = (h, g_1, \dots, g_N)$. One can use known sigma protocol theory [13] to describe a sigma protocol for this language with $\mathcal{O}(N)$ size last message, then use the compression technique to make the proof size $\mathcal{O}(\log N)$. See Appendix B.2 for the full argument.

2.6 Set proofs and range proofs

A *set-proof* shows that committed elements belong to some structured set F .

$$\mathcal{R}_{set}^{\mathbb{Z}_q, N, F} = \left\{ \left((h, g_1, \dots, g_N \in \mathbb{G}, \tau_I), (r, e_1, \dots, e_N \in \mathbb{Z}_q) \right) : \right. \\ \left. h^r \prod_{j=1}^N g_j^{e_j} = \tau_I \wedge e_j \in F, \forall j \in [1, N] \right\}.$$

Note that while showing that the elements of a multicommitment belongs to some arbitrary set can be quite expensive, for some particular types of sets, the proofs are very efficient.

A more specific example of a set-proof is a *range proof*, i.e., the case when the set is the interval $[0, 2^K - 1]$ for some fixed positive integer K . One example of how to do it is Bulletproofs [10,12].

We note that while $F = [0, 2^K - 1]$ is one particular example of an effective set proof using the Bulletproofs construction, the Bulletproofs construction actually allows to use the same technique for somewhat more general sets. In particular, the idea in Bulletproofs is that we prove that we can represent a committed value a as $a = \sum_{i=0}^{K-1} a_i 2^i$ and then show that the a_i are bits. However, the same idea could be used, for example, to show that one can represent a committed value a as $a = \sum_{i=0}^v a_i p^i$ for some public value p and then show that the a_i are bits.

3 New sub-arguments and assumptions

Before describing our main construction, we introduce some tools that will be used in the main protocol. Due to size constraints, we will not be able to present them in the main body and they will be given in the appendices along with the necessary proofs.

3.1 Showing that the values and randomness in two commitments must be the same

Let us call a set of elements $h, \{g_j\}_{j=1}^N$ in some group a *trusted basis* if we assume that no nontrivial DLRELS are known between them. (For example, they were sampled randomly from a group where a corresponding hardness assumption is believed to hold.) We call the above set of elements an *untrusted basis* if we do not have this assumption about them.

Given a trusted basis $\bar{h}, \{g_j\}_{j=1}^N$ and an untrusted basis $h, \{w_j\}_{j=1}^N$ the argument $\Pi_{samecom}^N$ proves that you know the witness in the relation $\mathcal{R}_{samecom}^N$ given below. The full argument can be seen in Appendix B.3.

$$\mathcal{R}_{samecom}^N = \left\{ \left((h, \{w_j\}_{j=1}^N, \bar{h}, \{g_j\}_{j=1}^N \in \mathbb{G}, \rho, \tau), (r, e_1, \dots, e_N \in \mathbb{Z}_q) \right) : \right. \\ \left. h^r \prod_{j=1}^N w_j^{e_j} = \rho \wedge \bar{h}^r \prod_{j=1}^N g_j^{e_j} = \tau \right\}.$$

The relation is also known as the linking relation, see Lego SNARK [11].

3.2 Showing that commitment's randomizer is the same value in the exponent of H

Given an element λ , a randomly generated group element H , commitment key \hat{h}, g_1, \dots, g_N , and a value τ_I , we need to prove knowledge of $r, \{e_j\}_{j=1}^N$, such that $\hat{h}^r \prod_{j=1}^N g_j^{e_j} = \tau$, and $H^r = \lambda$. We assume that no nontrivial DLREs are known between the $H, \hat{h}, \{g_i\}_{i=1}^N$. The relation $\mathcal{R}_{comrand}$ is shown below. Note that it is essentially $\mathcal{R}_{samecom}^N$ with $\{w_j\}_{j=1}^N = \mathbf{1}$. The full argument for this relation is given in Fig. 8 in Appendix B.4.

$$\mathcal{R}_{comrand}^N = \left\{ \left((H, \hat{h}, g_1, \dots, g_N \in \mathbb{G}, \lambda \in \mathbb{Z}_q, \tau), (r, e_1, \dots, e_N \in \mathbb{Z}_q) \right) : \right. \\ \left. \hat{h}^r \prod_{j=1}^N g_j^{e_j} = \tau \wedge H^r = \lambda \right\}.$$

3.3 Showing that the (weighted) sum of committed elements is equal to v

Given a trusted basis \hat{h}, g_1, \dots, g_N , publicly known integers a_1, \dots, a_N , public value v and a commitment in that basis τ_I , we need to show that you know $r, \{e_j\}_{j=1}^N$ such that $\hat{h}^r \prod_{j=1}^N g_j^{e_j} = \tau_I$ and that $\sum_{j=1}^N a_j e_j = v$. The relation \mathcal{R}_{comsum} is given below. The full argument is given in Fig. 9 in Appendix B.5.

$$\mathcal{R}_{comsum}^N = \left\{ \left((\hat{h}, g_1, \dots, g_N \in \mathbb{G}, a_1, \dots, a_N, v \in \mathbb{Z}_q, \tau), (r, e_1, \dots, e_N \in \mathbb{Z}_q) \right) : \right. \\ \left. \hat{h}^r \prod_{j=1}^N g_j^{e_j} = \tau \wedge \sum_{j=1}^N a_j e_j = v \right\}.$$

3.4 Showing that the messages in several commitments are the same

This is a slight variation of a previous primitive. Here, we are given v commitments $\{\rho_i\}_{i=1}^v$, $vN + v$ basis elements $\{g_{i,j}\}_{i=1,j=1}^{v,N}, \{h_i\}_{i=1}^v$ where no nontrivial DLREL is known between these. We are asked to show that the prover knows $\{r_i\}_{i=1}^v$ and $\{e_j\}_{j=1}^N$ such that $\rho_i = h_i^{r_i} \prod_{j=1}^N g_{i,j}^{e_j}$. We assume that the prover has already shown that she knows how to open the ρ_i in these respective bases, we just need to show that the elements are the same. Note that it is essentially a generalization of $\mathcal{R}_{samecom}^N$. The protocol for showing the following relation is discussed in Appendix B.6.

$$\mathcal{R}_{samemes}^N = \left\{ \left((h_1, \dots, h_v, g_{1,1}, g_{1,2}, \dots, g_{v,N} \in \mathbb{G}, \rho_1, \dots, \rho_v), \right. \right. \\ \left. \left. (r_1, \dots, r_N, e_1, \dots, e_N \in \mathbb{Z}_q) \right) : \right. \\ \left. \rho_i = h_i^{r_i} \prod_{j=1}^N g_{i,j}^{e_j}, \forall i \in [1, v] \right\}.$$

3.5 Additional assumptions

Definition 2 (Shortest Integer Solution [3]). Let $\mathbf{a}_1, \dots, \mathbf{a}_v \in \mathbb{Z}_q^N$ be uniformly randomly sampled. The $SIS_{q,N,v,L}$ -assumption states that it is hard for an adversary to find a vector $\mathbf{b} = (b_1, \dots, b_N) \in \mathbb{Z}^N$ such that $\langle \mathbf{b}, \mathbf{a}_j \rangle = 0 \pmod{q}$ for all $j \in [1, v]$ and $|b_i| \leq L$ for all $i \in [1, N]$.

We note here that usually in this definition the field size q is relatively small (usually quadratic in the security parameter) and thus v needs to be approximately the size of the desired security parameter. In our case, however, the field size is exponentially big, and N may be large. The case of a big q and N has been studied before, and it has been shown [18,8], that even if $v = 1$ (i.e., 1-dimensional SIS), one can show it to be as good as classic lattice assumptions, provided that $q \geq L \cdot \sqrt{N\lambda \log \lambda}$ and $N \geq \lambda \log q$. Thus it seems plausible that even in our case, if we picked a large-enough prime q and assume a large N , $v = 1$ would suffice. However, our design also allows to choose a larger v without much extra cost. Thus the choice of v is a matter of preference, depending on which version of SIS-in-a-large-field one uses.

To describe our final assumption, we must first define "small" discrete logarithm relations. Thus, we come to the following definition:

Definition 3 (NoBounded-L-DLRel). *For a set of elements $\{g_{i,j}\}_{i=1,j=1}^{N,v}$ we say that the (N, v) -NoBounded-L-DLRel property holds for an adversary \mathcal{A} if there exists a negligible function $\mu(\lambda)$ such that*

$$\Pr \left[\mathbf{a} = (\{a_i\}_{i=1}^N) \leftarrow \mathcal{A}(\mathbb{G}, \{g_{i,j}\}_{i=1,j=1}^{N,v}) : \mathbf{a} \neq \mathbf{0} \wedge \|\mathbf{a}\| \leq L \wedge \prod_{i=1}^N g_{i,j}^{a_i} = 1, \forall j \in [1, v] \right] = \mu(\lambda).$$

Note that this is actually not yet an assumption, but a property of a set. We are merely introducing the language so that we would be able to say later that a given set has this property.

Proposition 1. *Let \mathbb{G} be a group with order q and assume $SIS_{q,N,v,L}$ holds. Let the set $\{g_{1,1}, \dots, g_{N,v}\} \subset \mathbb{G}$ be obtained the following way. Let $g, h \in \mathbb{G}$ be chosen in such a way that no DLRELS are known between g and h . The adversary \mathcal{A} picks $a'_1, \dots, a'_N, b_1, \dots, b_N$. Then, $c_{1,1}, \dots, c_{1,N}, c_{2,1}, \dots, c_{v,N}$ are picked uniformly at random from $[0, \dots, |\mathbb{G}| - 1]$. We set $g_{i,j} \leftarrow h^{a'_i - c_{i,j}} g^{b_i}$. Then (N, v) -NoBounded-L-DLRel property holds for \mathcal{A} .*

Note that here we essentially assume that the prover not only knows all the committed messages and randomnesses, but has picked them herself. In practice, the prover likely has much less knowledge and power over the situation, thus she will not be able to find short DLRELS, if our assumptions hold.

4 The main technique

4.1 Basic shuffle

If we assume that the prover knows no DL relations between input ciphertexts, then we can obtain the basic shuffle argument Π_{scs}^{lite} depicted in Fig. 1. The argument is very efficient, as we mostly need κ range proofs that can be batched. Using Bulletproofs, the verifier only needs to perform $2N + 1$ exponentiations.

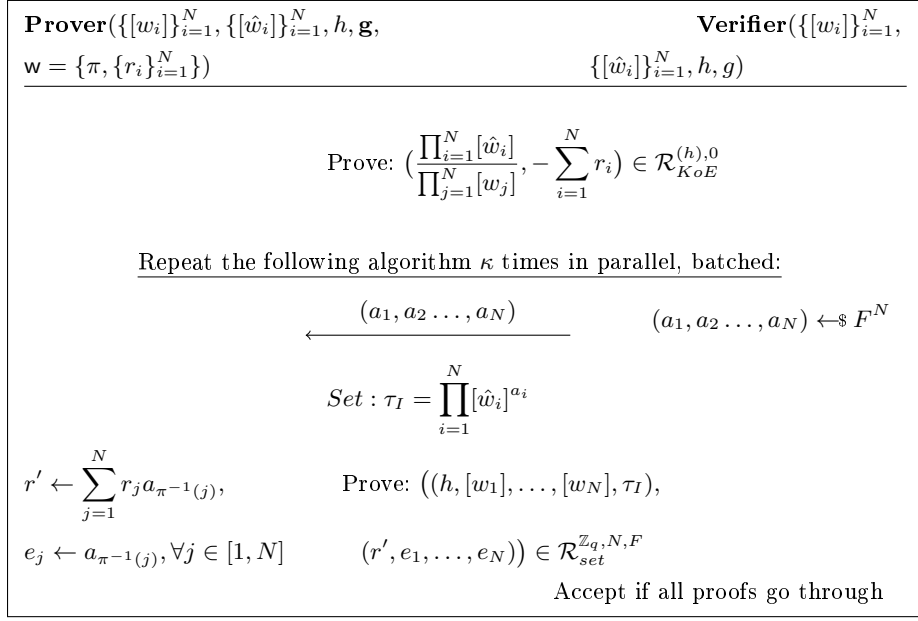


Fig. 1: The basic shuffle argument Π_{scs}^{lite} .

Theorem 1. *Let F be $\{0, 1\}$ and $\hat{T} = \frac{3}{4}$. Suppose that it is hard for the prover to find a nontrivial DLREL between $h, \{[w_i]\}_{i=1}^N$. Then if the verifier accepts in Π_{scs}^{lite} with probability at least $(\hat{T} + \varepsilon)^\kappa$ for a non-negligible ε , there exists an extractor that extracts a valid permutation and rerandomization factors.*

Proof (Sketch). It is easy to see that the theorem claim is equivalent to the claim that one single test has a success chance that is $\hat{T} + \varepsilon$ for a non-negligible ε . Thus, let us consider the following rewinding experiment.

For all $t = 1, \dots, N$, we argue that with polynomially many queries we can obtain two challenge vectors $a_{1,t}, \dots, a_{t,t}, \dots, a_{N,t}$ and $a_{1,t}, \dots, a_{t,t} + 1, \dots, a_{N,t}$ for which the verifier accepts, that only differ at the t -th coordinate. This follows from Lemma 3 from which we have that we can obtain two challenge vectors that only differ on the t th position, and as the challenge elements are bits, they can only differ by 1. We can extract for both of challenges the values $(r'_t, \{e'_{j,t}\}_{j=1}^N)$ and $(r''_t, \{e''_{j,t}\}_{j=1}^N)$ respectively by using the extractors of the set proofs.

Thus $\prod_{i=1}^N [\hat{w}_i]^{a_{i,t}} = h^{r'_t} \prod_{j=1}^N [w_j]^{e'_{j,t}}$ and $[\hat{w}_t] \prod_{i=1}^N [\hat{w}_i]^{a_{i,t}} = h^{r''_t} \prod_{j=1}^N [w_j]^{e''_{j,t}}$. Denote $d_{t,j} := e''_{j,t} - e'_{j,t}$ and $r_t := r''_t - r'_t$. Dividing the second equation by the first one we obtain $[\hat{w}_t] = h^{r_t} \prod_{j=1}^N [w_j]^{d_{t,j}}$. Let D be the matrix $\{d_{t,j}\}_{t=1, j=1}^{N,N}$. Note that all elements of D are either $-1, 0$ or 1 .

Now, take any successful response to a challenge $\{a_i\}_{i=1}^N$. We can extract $r, \{e_j\}_{j=1}^N$ from it such that $\prod_{i=1}^N [\hat{w}_i]^{a_i} = h^r \prod_{j=1}^N [w_j]^{e_j}$. On the other hand $\prod_{i=1}^N [\hat{w}_i]^{a_i} = \prod_{i=1}^N (h^{r_i} \prod_{j=1}^N [w_j]^{d_{i,j}})^{a_i} = h^{\sum_{i=1}^N a_i r_i} \prod_{j=1}^N [w_j]^{\sum_{i=1}^N a_i d_{i,j}}$.

Now, because we assumed that no nontrivial DLREs are known between the $[w_j]$ it follows that for all j , we have that $e_j = \sum_{i=1}^N a_i d_{i,j}$. On the other hand $e_j \in F$. Thus the prerequisite for a successful answer to the verifier accepting is that $\sum_{i=1}^N a_i d_{i,j} \in F$. Thus, if the verifier happens to pick $\{a_i\}_{i=1}^N$ in such a way that $\sum_{i=1}^N a_i d_{i,j} \notin F$, then the verifier will not accept. Thus it must hold that $\hat{T} \leq p_{D,F}$ by the definition of $p_{D,F}$.

Now, it will suffice to show that if D is not a permutation matrix, we will have that $p_{D,F} < \hat{T}$. We note that as the first step, in the KoE-proof, the prover has to show that she knows a R such that $h^R = \frac{\prod_{i=1}^N [\hat{w}_i]}{\prod_{j=1}^N [w_j]}$. This implies that $h^R \prod_{j=1}^N [w_j] = \prod_{i=1}^N [\hat{w}_i] = \prod_{i=1}^N (h^{r_i} \prod_{j=1}^N [w_j]^{d_{i,j}}) = h^{\sum_{i=1}^N r_i} \prod_{j=1}^N [w_j]^{\sum_{i=1}^N d_{i,j}}$. Thus, we have that for all $j \in [1, N]$ we have that $\sum_{i=1}^N d_{i,j} = 1$. Now the result follows directly from Lemma 13, as that gives us $p_{D,F} \leq \frac{3}{4} < \hat{T}$. \square

4.2 Challenge-Response subargument

Our full shuffle argument will have the following part — the main part, where some preparations are done for the subargument ChalResp, which is then called κ times with two variations. The argument will essentially be a random challenge from a challenge set and a response, with the response being mostly proofs of different types. The argument will allow us to extract a unique response. We will run the argument for two different challenge sets.

Lemma 1. *Consider the $\Pi_{ChalResp}$ argument depicted in Fig. 2, and assume that no nontrivial DLOG relations are known between H, h, g, ck and the $\{\text{ck}_k\}_{k=1}^v$.*

Let there also be an extractor who is given $\{r'_{t,k}\}_{k=1, t=1}^v, \{\delta_k\}_{k=1}^v$ such that these satisfy $C_k = h^{\delta_k} \prod_{t=1}^v g_t^{r'_{t,k}}$. Denote $[\hat{w}_{t,k}]' := \frac{[\hat{w}_t]}{h^{r'_{t,k}}}$. Let the first message of the Verifier be some (b_1, \dots, b_N) where each $b_i \in F$. Assume that the Verifier accepts. Then the extractor can extract $\{e_j\}_{j=1}^N$, which satisfy $\prod_{j=1}^N [w'_{j,k}]^{e'_j} = \prod_{t=1}^v [\hat{w}_{t,k}]'^{b_t}$. Additionally, $e_j \in F$ for all j and $\sum_{j=1}^N e_j = \sum_{t=1}^v b_t$.

Proof. (Sketch). Assume that the verifier accepts the argument $\Pi_{ChalResp}$. Firstly, the verifier accepts the arguments $\Pi_{comrand}^N$, i.e., for $k \in [1, v]$ the prover provides accepting arguments for the relation $(H, \hat{h}_k, g_{1,k}, \dots, g_{N,k}, \lambda_{I,k}, \tau_{I,k}) \in \mathcal{R}_{comrand}^N$. Thus we can extract r'_k and $(e_{1,1}, \dots, e_{v,N})$ such that $\tau_{I,k} = \hat{h}_k^{r'_k} \prod_{j=1}^N g_{j,k}^{e_{j,k}}$ for all k and $H^{r'_k} = \lambda_{I,k}$. Similarly, since the verifier accepts $\Pi_{samemes}^N$, for $(\hat{h}_1, \dots, \hat{h}_v, g_{1,1}, \dots, g_{N,v}, \tau_{I,1}, \dots, \tau_{I,k})$, we can extract r''_k and (e'_1, \dots, e'_N) such that $\tau_{I,k} = \hat{h}_k^{r''_k} \prod_{j=1}^N g_{j,k}^{e'_j}$ for all $k \in [1, v]$. Combining these we have that if for some $e_{j,k} \neq e'_j$ or $r'_k \neq r''_k$, then we have broken the DLOG-assumption for the $g_{j,k}$. Thus we have that $\tau_{I,k} = \hat{h}_k^{r'_k} \prod_{j=1}^N g_{j,k}^{e'_j}$ for all k and $H^{r'_k} = \lambda_{I,k}$.

Next, we have that for all k , the verifier accepts $\Pi_{in-prod}^{ck,N}$ went through, i.e., for $(C_k, \lambda_{I,k}, H, (b_1, \dots, b_N)) \in \mathcal{R}_{in-prod}^{ck,N}$ for $k \in [1, v]$, we can extract $\delta_{k'}$ and $R_{t,k}$ such that $C_k = h^{\delta'_k} \prod_{t=1}^N g_t^{R'_{t,k}}$, and $\lambda_{I,k} = H^{\sum_{t=1}^N b_t R'_{t,k}}$. Our extractor knows the values $\{r'_{t,k}\}_{k=1, t=1}^{v,N}, \{\delta_k\}_{k=1}^v$ such that $C_k = h^{\delta_k} \prod_{t=1}^N g_t^{r'_{t,k}}$. Due to the nontrivial DLOG property of $(h, \{g_t\}_{t=1}^N)$ we have that $R'_{t,k} = r'_{t,k}$ and $\delta_k = \delta'_k$ for all k and t . Hence we have that $H^{r'_k} = \lambda_{I,k} = H^{\sum_{t=1}^N b_t r'_{t,k}}$ and thus

$$r'_k = \sum_{t=1}^N b_t r'_{t,k}. \quad (1)$$

We also have that the $\Pi_{samecom}^N$ argument for the relation

$$(h, \{[w'_{j,k}]\}_{j=1}^N, \hat{h}, \{g_{j,k}\}_{j=1}^N, \prod_{t=1}^N [\hat{w}_t]^{b_t}, \tau_{I,k}) \in \mathcal{R}_{samecom}^N$$

is accepted. Thus we can extract the \bar{r}_k and $\bar{d}_{k,j}$ such that

$$h^{\bar{r}_k} \prod_{j=1}^N [w'_{j,k}]^{\bar{d}_{k,j}} = \prod_{t=1}^N [\hat{w}_t]^{b_t}$$

and

$$\hat{h}^{\bar{r}_k} \prod_{j=1}^N g_{j,k}^{\bar{d}_{k,j}} = \tau_{I,k}.$$

Thus

$$\hat{h}^{\bar{r}_k} \prod_{j=1}^N g_{j,k}^{\bar{d}_{k,j}} = \tau_{I,k} = \hat{h}_k^{r'_k} \prod_{j=1}^N g_{j,k}^{e'_j}$$

for all k . Hence $\bar{r}_k = r'_k$ and $e'_j = \bar{d}_{k,j}$ for all k . Thus

$$h^{r'_k} \prod_{j=1}^N [w'_{j,k}]^{e'_j} = \prod_{t=1}^N [\hat{w}_t]^{b_t}.$$

We have that $r'_k = \sum_{t=1}^N b_t r'_{t,k}$ and thus this is equivalent to

$$h^{\sum_{t=1}^N b_t r'_{t,k}} \prod_{j=1}^N [w'_{j,k}]^{e'_j} = \prod_{t=1}^N [\hat{w}_t]^{b_t}$$

or

$$\prod_{j=1}^N [w'_{j,k}]^{e'_j} = \prod_{t=1}^N \left(\frac{[\hat{w}_t]}{h^{r'_{t,k}}} \right)^{b_t} = \prod_{t=1}^N [\hat{w}_{t,k}]^{b_t}.$$

We have thus shown the first part of the claim. Now we have to show that the $e_i \in F$.

We also have that the range proofs are accepted. More specifically,

$$((\hat{h}_1, g_{1,1}, \dots, g_{N,1}, \tau_{I,1})) \in \mathcal{R}_{set}^{\mathbb{Z}_1, N, F}.$$

Thus we can extract $\{\hat{r}\}, \{d'_j\}_{j=1}^N$ such that

$$\hat{h}_1^{\hat{r}} \prod_{j=1}^N g_{j,1}^{d'_j} = \tau_{I,1} \wedge d'_j \in F .$$

We have that

$$\hat{h}_1^{\hat{r}} \prod_{j=1}^N g_{j,1}^{d'_j} = \tau_{I,1} = \hat{h}_1^{\tau_1} \prod_{j=1}^N g_{j,1}^{e'_j}$$

Thus $e'_j = d'_j$ where $d_j \in F$. Hence the second claim is proven.

Also, the verifier accepts Π_{consum}^N , i.e.,

$$(\hat{h}_1, (g_{1,1}, \dots, g_{N,1}), (1, \dots, 1), \sum_{t=1}^N b_t, \tau_{I,1}) \in \mathcal{R}_{consum}^N .$$

Thus we can extract \bar{r}' and \bar{d}'_j such that $\tau_{I,1} = \hat{h}_1^{\bar{r}'_1} \prod_{j=1}^N g_{j,1}^{\bar{d}'_j}$ and $\sum_j \bar{d}'_j = \sum_{t=1}^N b_t$. By previous equalities, we have that

$$\hat{h}_1^{\hat{r}} \prod_{j=1}^N g_{j,1}^{d'_j} = \tau_{I,1} = \hat{h}_1^{\bar{r}'_1} \prod_{j=1}^N g_{j,1}^{\bar{d}'_j},$$

thus we have that $d'_j = \bar{d}'_j$. Thus we also have that

$$\sum_j d'_j = \sum_{t=1}^N b_t.$$

4.3 Main body of the shuffle argument

We will now describe our full construction, along with explaining why we are using each part. Essentially our construction is a generalization of the simple version. We first have to do some preparations, and then we will run two sets of the ChalResp protocol κ times using different types of challenge sets. For preparation, we will need to compute intermediate commitment keys on v parallel public rerandomizations.⁸

⁸ The number v here will depend on our SIS-assumption. Essentially, if a cheating prover wants to use some known discrete non-trivial relation between the inputs, the relation has to be "short" and same across every rerandomization, meaning that effectively using it would break the SIS-assumption. We will expand on this more formally later.

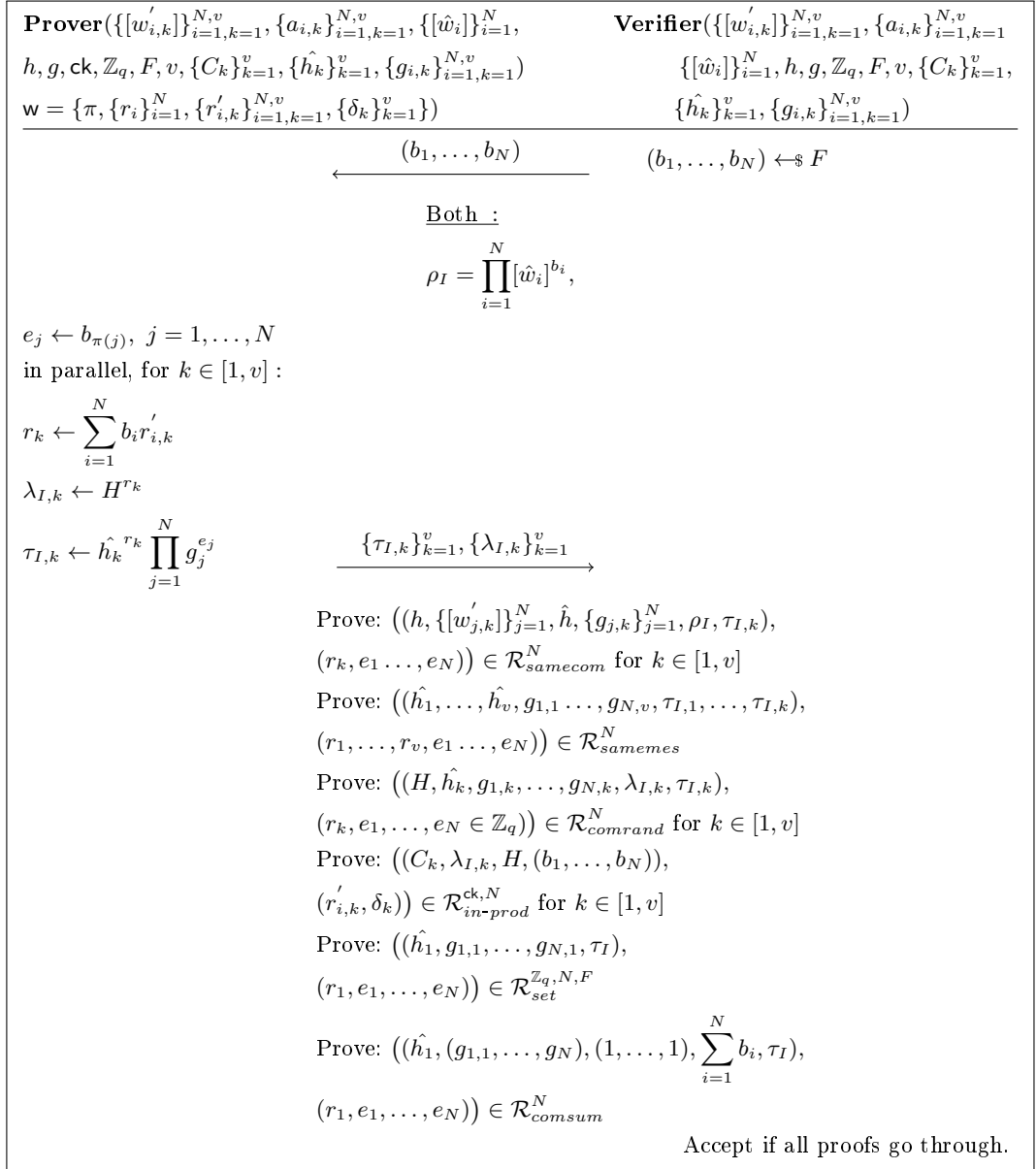


Fig. 2: One round of the shuffling argument ChalResp (without the random oracle, linear communication of random elements).

Next, the Verifier picks rerandomization factors $\{a_{i,1}\}_{i=1}^N, \dots, \{a_{i,v}\}_{i=1}^N$ and a number of random trusted bases sends them to the prover. The Prover and

Verifier both compute $[w'_{i,k}] \leftarrow h^{a_{i,k}}[w_i]$. The prover also computes $r_{i,k} \leftarrow r_i - a_{\pi^{-1}(i),k}$ for $i = 1, \dots, N, k = 1, \dots, v$ and commits to these with commitments C_1, \dots, C_v which she sends to the Verifier. The Prover shows that $\prod_{j=1}^N [w'_{j,k}] = \prod_{t=1}^N [\hat{w}_{t,k}]'$ for all k where $[\hat{w}_{t,k}]' := \frac{[\hat{w}_t]}{h^{r'_{t,k}}}$. The $[w'_{i,k}]$ and the $[\hat{w}_{t,k}]'$ will be our main building blocks. The fact that their products are the same will be an important tool as it will allow us to show that in the extracted matrix, the sum of all rows will be 1. The point of the commitments C_i is that they will help us to argue about $[\hat{w}_{t,k}]'$ and $[w_{t,k}]'$ instead of $[\hat{w}_t]$ and $[w_t]$.

After that they will do two types of ChalResp protocols, one where the challenge set will be an interval and another where it will be $P_{p,f}$. The reason for the two types of tests is that it is much easier to extract from the interval, however, the $P_{p,f}$ will give much more efficient results.

Fine-tuning the ChalResp protocol

The prover will send N random elements $\{b_i\}_{i=1}^N$ from the challenge set F . In the simpler version, the prover would show that she knows how to open the product $\rho_I = \prod_{i=1}^N [\hat{w}_i]^{b_i}$ in the basis $h, \{[w_i]\}_{i=1}^N$. However, here we will want to do all our subproofs on a trusted basis, because a number of them simply would not work on an untrusted basis. Thus, at the first step, we want to move to that trusted basis $(\hat{h}, \{g_{j,k}\}_{j=1}^N)$. An honest prover would know $e_j = b_{\pi(j)}$ and randomizers $r_k = \sum_{i=1}^N b_i r'_{i,k}$ such that $h^{r_k} \prod_{j=1}^N [w'_{j,k}] = \rho_I$. Instead of showing that, she will produce $\tau_{I,k} = \hat{h}_k^{r_k} \prod_{j=1}^N g_j^{e_j}$, show using the $\Pi_{samecom}^N$ argument that the values of $\tau_{I,k}$ when opened in basis $\hat{h}, \{g_{j,k}\}_{j=1}^N$ are the same as the values when ρ opened in the basis $h, \{[w'_{j,k}]\}_{j=1}^N$ would be and then proceed working with $\tau_{I,k}$ and the trusted basis.

For the SIS-arguments to work, the response to the challenge $\{b_i\}_{i=1}^N$ must be the same in every public rerandomization, which can be guaranteed using the argument $\Pi_{samemes}^N$.

A cheating prover might be tempted to pick the rerandomization factors $r'_{i,k}$ in a dishonest way. Remember that because a prover might know the contents of the messages, the fact that the secret rerandomization factors are a crucial part of the protocol. The arguments $\Pi_{comrand}^N$ and $\Pi_{in-prod}^N$ force the rerandomization factors to be exactly what is in the commitments C_k .

Now the conditions are set that we may run the actual range proof on the challenge value τ_I and the basis $(\hat{h}_1, g_{1,1}, \dots, g_{N,1}, \tau_I)$. Note that we do not actually have to run it on v parallel rerandomizations because it has been previously proven that the committed values will be same for all of them, thus we only need to do this for one of them.

Finally, the Π_{comsum} argument makes it sure that $\sum_{i=1}^N b_i = \sum_{i=1}^N e_i$, which will allow us to show that in the extracted matrix, all columns must sum to 1. The fact that both all rows and all columns must sum to 1 will be an important part of the proof that the extracted matrix must be a permutation matrix.

Theorem 2. *Let $p, 2^{K+1} < N$. Suppose that the SIS $_{q,N,v,(N+1)\cdot(\sum_{i=0}^f p^i)}$ holds. If p is odd, then let $2^K \leq \frac{p}{2}$. If p is even, then let $2^K \leq p - 2$. Let $T = \min\{2^K, (\frac{8}{5})^f\}$. The protocol depicted in Fig. 3 has $(\frac{1}{T} + \epsilon_1)^\kappa + \epsilon_2$ -witness-extended-emulation where ϵ_1 and ϵ_2 are negligible.*

Proof given in Appendix C.

One can pick p and f accordingly to the SIS-bound and get the respective soundness error of one round according to that.

For example, if $(N + 1)2^{26}$ -SIS holds, then one can take $p = 18$, $K = 4$, $f = 6$, check that $(N + 1)\sum_{i=0}^6 18^i \leq (N + 1)2^{26}$ and obtain that one round of ChalResp has a soundness error of $\approx \frac{1}{16}$. Thus then, one can take $\kappa = \frac{1^\lambda}{4}$. If $(N + 1)2^{55}$ -SIS holds, then one can take $p = 66$, $K = 6$, $f = 9$, check that $(N + 1)\sum_{i=0}^9 66^i \leq (N + 1)2^{55}$ and obtain that one round of ChalResp has a soundness error of $\approx \frac{1}{64}$. Thus then, one can take $\kappa = \frac{1^\lambda}{6}$. To simplify the analysis of efficiency, we take κ to be $\lambda/4$.

Finally, by the completeness and HVZK of the sub-arguments, we obtain security of the full shuffle argument.

Theorem 3. *Let $\kappa = \lambda/4$. Then the shuffle argument in Fig. 3 is perfectly complete, $(2^{-\lambda})$ -WEE and special honest-verifier zero-knowledge.*

4.4 ElGamal shuffle

Suppose that the prover instead wants to prove the correctness of an ElGamal shuffle. Let $\mathcal{R}_{Elg}^{N,g,h}$ be the shuffle relation for ElGamal ciphertexts defines as

$$\mathcal{R}_{Elg}^{N,g,h} = \left\{ \left((C = \{(c_{i,1}, c_{i,2})\}_{i=1}^N, \hat{C} = \{(\hat{c}_{i,1}, \hat{c}_{i,2})\}_{i=1}^N), (\pi, \{r_i\}_{i=1}^N) \right) : \pi \in \mathbb{S} \wedge \begin{array}{l} c_{i,1} = \hat{c}_{\pi(i),1} \cdot g^{r_{\pi(i)}} \wedge c_{i,2} = \hat{c}_{\pi(i),2} \cdot h^{r_{\pi(i)}} \text{ for } i = 1, \dots, N \end{array} \right\}.$$

We will in the appendix show that using an extra random challenge η and setting $[w_i] := c_{i,1}c_{i,2}^\eta$ and $[\hat{w}_i] := \hat{c}_{i,1}\hat{c}_{i,2}^\eta$, and the new \hat{h} as gh^η , we can obtain a proof-of-shuffle of ciphertexts with practically no extra costs. For more details, see Appendix C.2.

Theorem 4. *Let $v \geq 2$. Let $\kappa \geq 2$. If the underlying shuffle argument for commitments has α -WEE, then the argument given in Figure 10 is a $(\alpha + \frac{N^2}{q})$ -WEE proof-of-shuffle for ElGamal ciphertexts.*

4.5 Efficiency

We discuss the efficiency of the sub-arguments, one round of the shuffle argument (with noticeable soundness error), and the full shuffle argument (with negligible soundness error) in Section 4.5. For a shuffle argument that achieves noticeable soundness with soundness error $1/16$, one can take Π_{scs} with $v = \kappa = 1$. Based on the discussion following Definition 2, to achieve negligible soundness with soundness error $2^{-\lambda}$ we can take $v = 1$ and $\kappa = \lambda/4$ assuming N is large.

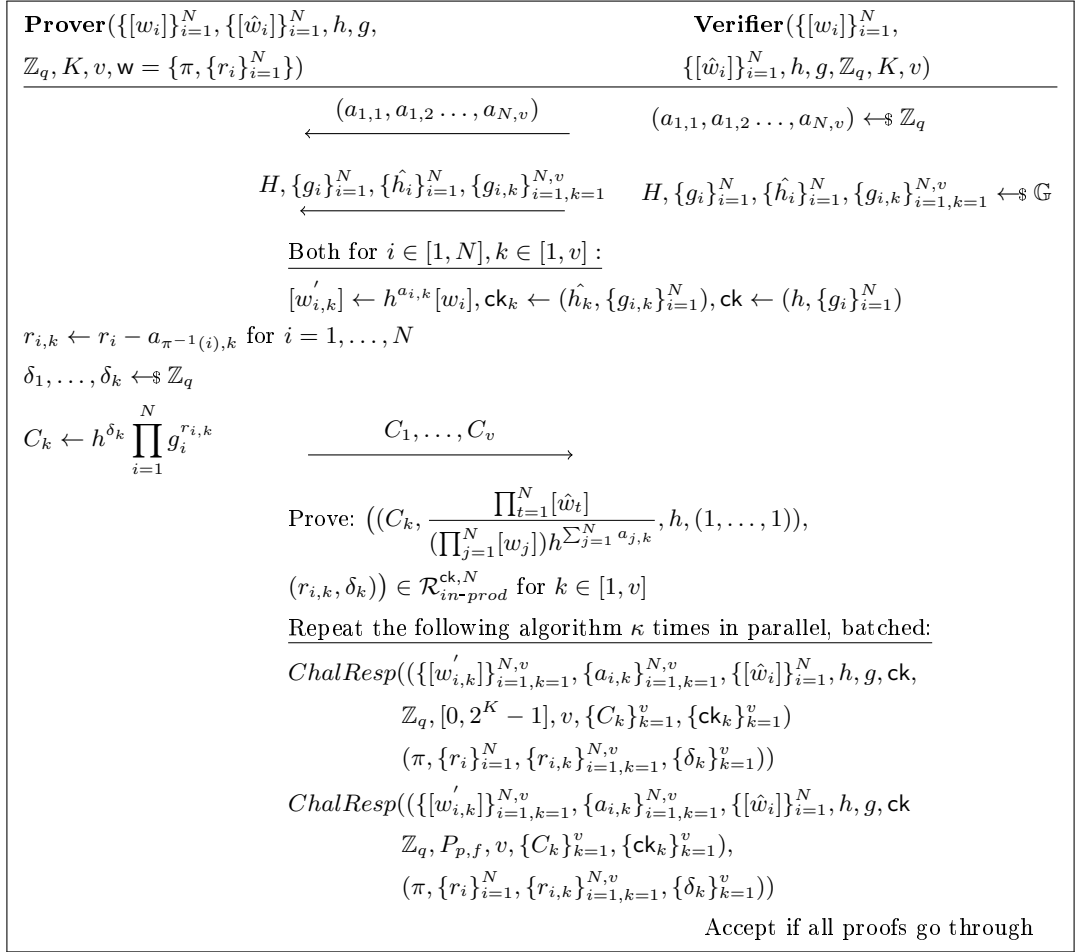


Fig. 3: The full shuffle argument Π_{scs} .

However, to directly use Theorem 4 we will conservatively take $v = 2$ and $\kappa = \lambda/4$. We leave as an open problem how to prove soundness for smaller values of v and κ .

Note that the prover and verifier complexity is dominated by multi-exponentiations of width N (or more), which can be optimized using Pippenger's algorithm [25] to get a $\log N$ speedup. Verifier's complexity is much more efficient than prover's due to the use of batching techniques on repetitions of the same arguments. Additionally, random integers and group elements sent by the verifier can be replaced by a single random seed that is then fed to a pseudorandom generator.

Argument	Prover (exp)	Verifier (exp)	Communication	CRS size
Π_{KoE}^N	$3Nv$	$2N$	$2 \log(Nv) \times \mathbb{G}$	$(N+1) \times \mathbb{G}$
$\Pi_{samecom}^N$	$6Nv$	$2N$	$2 \log(Nv) \times \mathbb{G}$	$(N+1) \times \mathbb{G}$
$\Pi_{samemes}^N$	$3Nv$	$2N$	$2 \log(Nv) \times \mathbb{G}$	$(Nv+N+1) \times \mathbb{G}$
$\Pi_{comrand}^N$	$3Nv$	$2N$	$2 \log(Nv) \times \mathbb{G}$	$(N+1) \times \mathbb{G}$
$\Pi_{in-prod}$	$3Nv$	$2N$	$2 \log(Nv) \times \mathbb{G}$	$(N+1) \times \mathbb{G}$
$\Pi_{set}^{z_q, N, F}$ [10]	$12Nv$	$2N$	$2 \log(Nv) \times \mathbb{G}$	$(N+1) \times \mathbb{G}$
Π_{consum}^N	$8Nv$	$2N$	$2 \log(Nv) \times \mathbb{G}$	$(N+1) \times \mathbb{G}$
ChalResp	$35Nv + N$	$12N$	$14 \log(Nv) \times \mathbb{G}$	$(Nv+N+1) \times \mathbb{G}$
Basic shuffle Π_{scs}^{lite}	$12N\kappa + 1$ exp.	$2N + 2$ exp.	$2N + 2 \log(N\kappa) + 2 \times \mathbb{G}$	$(N+1) \times \mathbb{G}$
Full shuffle Π_{scs}	$70Nv\kappa + N(v+3)$ exp.	$26N$ exp.	$2N + 28 \log(Nv\kappa) \times \mathbb{G}$	$2 \times \mathbb{G}$

Table 2: Efficiency of our shuffle argument and v runs of the sub-arguments. Exp. stands for exponentiations, pair. for pairings, N is the number of input ciphertexts, and κ is the number of repetitions of ChalResp. Constant terms are neglected, shuffling is included to prover’s efficiency, and shuffled ciphertexts are included to proof size.

5 Acknowledgment

This research is supported by the National Research Foundation, Singapore under its Strategic Capability Research Centres Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

Author Toomas Krips was partly supported by the Estonian Research Council, ETAG, through grant PRG 946.

References

- [1] Adida, B.: Helios: Web-based open-audit voting. In: van Oorschot, P.C. (ed.) *USENIX Security 2008*. pp. 335–348. USENIX Association (Jul / Aug 2008)
- [2] Aggelakis, A., Fauzi, P., Korfiatis, G., Louridas, P., Mergoupis-Anagnostou, F., Siim, J., Zajac, M.: A non-interactive shuffle argument with low trust assumptions. In: Jarecki, S. (ed.) *CT-RSA 2020*. LNCS, vol. 12006, pp. 667–692. Springer, Heidelberg (Feb 2020). https://doi.org/10.1007/978-3-030-40186-3_28
- [3] Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: *28th ACM STOC*. pp. 99–108. ACM Press (May 1996). <https://doi.org/10.1145/237814.237838>
- [4] Aranha, D.F., Baum, C., Gjøsteen, K., Silde, T., Tunge, T.: Lattice-based proof of shuffle and applications to electronic voting. In: Paterson, K.G. (ed.) *CT-RSA 2021*. LNCS, vol. 12704, pp. 227–251. Springer, Heidelberg (May 2021). https://doi.org/10.1007/978-3-030-75539-3_10
- [5] Attema, T., Cramer, R.: Compressed Σ -protocol theory and practical application to plug & play secure algorithmics. In: Micciancio, D., Ristenpart, T. (eds.) *CRYPTO 2020, Part III*. LNCS, vol. 12172, pp. 513–543. Springer, Heidelberg (Aug 2020). https://doi.org/10.1007/978-3-030-56877-1_18
- [6] Attema, T., Cramer, R., Kohl, L.: A compressed Σ -protocol theory for lattices. In: Malkin, T., Peikert, C. (eds.) *CRYPTO 2021, Part II*. LNCS, vol. 12826, pp. 549–579. Springer, Heidelberg, Virtual Event (Aug 2021). https://doi.org/10.1007/978-3-030-84245-1_19
- [7] Boneh, D., Eskandarian, S., Hanzlik, L., Greco, N.: Single secret leader election. *Cryptology ePrint Archive, Report 2020/025* (2020), <https://eprint.iacr.org/2020/025>
- [8] Brakerski, Z., Vaikuntanathan, V.: Constrained key-homomorphic PRFs from standard lattice assumptions - or: How to secretly embed a circuit in your PRF. In: Dodis, Y., Nielsen, J.B. (eds.) *TCC 2015, Part II*. LNCS, vol. 9015, pp. 1–30. Springer, Heidelberg (Mar 2015). https://doi.org/10.1007/978-3-662-46497-7_1
- [9] Brands, S.: Untraceable off-line cash in wallets with observers (extended abstract). In: Stinson, D.R. (ed.) *CRYPTO'93*. LNCS, vol. 773, pp. 302–318. Springer, Heidelberg (Aug 1994). https://doi.org/10.1007/3-540-48329-2_26
- [10] Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: *2018 IEEE Symposium on Security and Privacy*. pp. 315–334. IEEE Computer Society Press (May 2018). <https://doi.org/10.1109/SP.2018.00020>
- [11] Campanelli, M., Fiore, D., Querol, A.: LegoSNARK: Modular design and composition of succinct zero-knowledge proofs. In: Cavallaro, L., Kinder,

- J., Wang, X., Katz, J. (eds.) ACM CCS 2019. pp. 2075–2092. ACM Press (Nov 2019). <https://doi.org/10.1145/3319535.3339820>
- [12] Chung, H., Han, K., Ju, C., Kim, M., Seo, J.H.: Bulletproofs+: Shorter proofs for privacy-enhanced distributed ledger. Cryptology ePrint Archive, Report 2020/735 (2020), <https://eprint.iacr.org/2020/735>
- [13] Damgård, I.: On σ -protocols. Lecture Notes, University of Aarhus, Department for Computer Science p. 84 (2002)
- [14] Faonio, A., Fiore, D., Herranz, J., Ràfols, C.: Structure-preserving and re-randomizable RCCA-secure public key encryption and its applications. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 159–190. Springer, Heidelberg (Dec 2019). https://doi.org/10.1007/978-3-030-34618-8_6
- [15] Fauzi, P., Meiklejohn, S., Mercer, R., Orlandi, C.: Quisquis: A new design for anonymous cryptocurrencies. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part I. LNCS, vol. 11921, pp. 649–678. Springer, Heidelberg (Dec 2019). https://doi.org/10.1007/978-3-030-34578-5_23
- [16] Fleischhacker, N., Simkin, M.: On publicly-accountable zero-knowledge and small shuffle arguments. In: Garay, J. (ed.) PKC 2021, Part II. LNCS, vol. 12711, pp. 618–648. Springer, Heidelberg (May 2021). https://doi.org/10.1007/978-3-030-75248-4_22
- [17] Furukawa, J., Sako, K.: An efficient scheme for proving a shuffle. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 368–387. Springer, Heidelberg (Aug 2001). https://doi.org/10.1007/3-540-44647-8_22
- [18] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 197–206. ACM Press (May 2008). <https://doi.org/10.1145/1374376.1374407>
- [19] Gradshteyn, I.S., Ryzhik, I.M.: Table of integrals, series, and products. Academic press (2014)
- [20] Groth, J.: Evaluating security of voting schemes in the universal composability framework. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 04. LNCS, vol. 3089, pp. 46–60. Springer, Heidelberg (Jun 2004). https://doi.org/10.1007/978-3-540-24852-1_4
- [21] Haines, T., Müller, J.: SoK: Techniques for verifiable mix nets. In: Jia, L., Küsters, R. (eds.) CSF 2020 Computer Security Foundations Symposium. pp. 49–64. IEEE Computer Society Press (2020). <https://doi.org/10.1109/CSF49147.2020.00012>
- [22] Hoffmann, M., Kloof, M., Rupp, A.: Efficient zero-knowledge arguments in the discrete log setting, revisited. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) ACM CCS 2019. pp. 2093–2110. ACM Press (Nov 2019). <https://doi.org/10.1145/3319535.3354251>
- [23] Lindell, Y.: Parallel coin-tossing and constant-round secure two-party computation. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 171–189. Springer, Heidelberg (Aug 2001). https://doi.org/10.1007/3-540-44647-8_10

- [24] Neff, C.A.: A verifiable secret shuffle and its application to e-voting. In: Reiter, M.K., Samarati, P. (eds.) ACM CCS 2001. pp. 116–125. ACM Press (Nov 2001). <https://doi.org/10.1145/501983.502000>
- [25] Pippenger, N.: On the evaluation of powers and monomials. *SIAM J. Comput.* **9**(2), 230–250 (1980). <https://doi.org/10.1137/0209022>, <https://doi.org/10.1137/0209022>
- [26] Schwartz, J.T.: Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM (JACM)* **27**(4), 701–717 (1980)
- [27] Team, T.E.F.C.R.: Curdleproofs: A shuffle argument protocol (2022), <https://github.com/asn-d6/curdleproofs>
- [28] Tyagi, N., Gilad, Y., Zaharia, M., Zeldovich, N.: Stadium: A distributed metadata-private messaging system. *Cryptology ePrint Archive, Report 2016/943* (2016), <https://eprint.iacr.org/2016/943>
- [29] Zippel, R.: Probabilistic algorithms for sparse polynomials. In: *Symbolic and Algebraic Computation: EUROSM'79, An International Symposium on Symbolic and Algebraic Manipulation, Marseille, France, June 1979*. pp. 216–226. Springer (1979)

A Omitted Preliminaries

A.1 Schwartz-Zippel Lemma

Lemma 2 (Schwartz-Zippel lemma [26,29]). *Let $f(X_1, \dots, X_n)$ be a non-zero polynomial of degree d over a finite field \mathbb{F} and let $S \subseteq \mathbb{F}$. Then,*

$$\Pr[f(x_1, \dots, x_n) = 0 : x_1, \dots, x_n \leftarrow S] \leq \frac{d}{|S|}.$$

A.2 Zero-Knowledge Argument

Let Pgen be a PPT parameter generation algorithm that on input 1^λ outputs \mathbf{p} (e.g., a description of the group or some other setup parameters). A zero-knowledge argument of knowledge for a relation \mathcal{R} is a tuple of efficient algorithms $(\text{Pgen}, \text{P}, \text{V})$ that satisfies properties of perfect completeness, computational witness-extended emulation, and perfect special honest verifier zero-knowledge, defined below. Prover algorithm P and verifier algorithm V are interactive algorithms and we denote their protocol transcript by $\text{tr} \leftarrow \langle \text{P}(\mathbf{p}, \mathbf{x}, \mathbf{w}), \text{V}(\mathbf{x}) \rangle$ where $(\mathbf{x}, \mathbf{w}) \in \mathcal{R}$ and $\mathbf{p} \in \text{Pgen}(1^\lambda)$. We write $\langle \text{P}(\mathbf{p}, \mathbf{x}, \mathbf{w}), \text{V}(\mathbf{x}) \rangle = 1$ to denote that verifier outputs 1 at the end of the interaction. Interactions with an adversary follow a similar notation.

Definition 4 (Perfect completeness). *An argument is perfectly complete if for any $(\mathbf{x}, \mathbf{w}) \in \mathcal{R}$,*

$$\Pr[\mathbf{p} \leftarrow \text{Pgen}(1^\lambda) : \langle \text{P}(\mathbf{p}, \mathbf{x}, \mathbf{w}), \text{V}(\mathbf{p}, \mathbf{x}) \rangle = 1] = 1.$$

Definition 5 (Witness-extended emulation). *An argument has witness-extended emulation with knowledge error κ , denoted κ -WEE, if there exists a PPT extractor Ext such that for any PPT \mathcal{A} , $|\varepsilon_0^{\text{we}} - \varepsilon_1^{\text{we}}| \leq \kappa(1^\lambda)$, where*

$$\varepsilon_0^{\text{we}} = \Pr[\mathbf{p} \leftarrow \text{Pgen}(1^\lambda), (\mathbf{x}, \text{state}) \leftarrow \mathcal{A}(\mathbf{p}), \text{tr} \leftarrow \langle \mathcal{A}(\text{state}), \text{V}(\mathbf{p}, \mathbf{x}) \rangle : \mathcal{A}(\text{state}, \text{tr}) = 1],$$

$$\varepsilon_1^{\text{we}} = \Pr\left[\mathbf{p} \leftarrow \text{Pgen}(1^\lambda), (\mathbf{x}, \text{state}) \leftarrow \mathcal{A}(\mathbf{p}), (\text{tr}, \mathbf{w}) \leftarrow \text{Ext}^{\mathcal{A}(\text{state})}(\mathbf{p}, \mathbf{x}) : \mathcal{A}(\text{state}, \text{tr}) = 1 \wedge \text{if tr is accepting then } (\mathbf{x}, \mathbf{w}) \in \mathcal{R}\right].$$

Definition 6. *An argument has special honest-verifier zero-knowledge (SHVZK) if there exists PPT simulator Sim such that for any adversary \mathcal{A} , $|\varepsilon_0^{\text{zk}} - \varepsilon_1^{\text{zk}}| \leq \text{negl}(\lambda)$, where*

$$\varepsilon_0^{\text{zk}} = \Pr\left[\mathbf{p} \leftarrow \text{Pgen}(1^\lambda), (\mathbf{x}, \mathbf{w}, r, \text{state}) \leftarrow \mathcal{A}(\mathbf{p}), \text{tr} \leftarrow \langle \text{P}(\mathbf{p}, \mathbf{x}, \mathbf{w}), \text{V}(\mathbf{p}, \mathbf{x}; r) \rangle : (\mathbf{x}, \mathbf{w}) \in \mathcal{R} \wedge \mathcal{A}(\text{tr}, \text{state}) = 1\right],$$

$$\varepsilon_1^{\text{zk}} = \Pr\left[\mathbf{p} \leftarrow \text{Pgen}(1^\lambda), (\mathbf{x}, \mathbf{w}, r, \text{state}) \leftarrow \mathcal{A}(\mathbf{p}), \text{tr} \leftarrow \text{Sim}(\mathbf{p}, \mathbf{x}, r) : (\mathbf{x}, \mathbf{w}) \in \mathcal{R} \wedge \mathcal{A}(\text{tr}, \text{state}) = 1\right].$$

Note that we allow \mathcal{A} to choose random coins r of the verifier.

Moreover, we say that an argument is public coin if all of verifier’s messages are uniformly random bit-strings. This is useful since it allows to make the argument non-interactive with the Fiat-Shamir heuristic.

Note that our definition of witness-extended emulation can be seen as a generalization of the more standard definitions of knowledge soundness and special soundness. In particular, Lindell [23] showed that if an argument is knowledge sound with negligible knowledge error, then there exists a witness-extended emulator for the argument. Additionally, Groth [20] showed that witness-extended emulation is implied by special soundness.

A.3 Compressed Sigma Protocols

We can use compressed sigma protocol theory [5] to transform a sigma protocol with $\mathcal{O}(N)$ size last message to one with $\mathcal{O}(\log N)$ size. The main idea is that in many Sigma protocols the verification equation applies some linear map to the last message \mathbf{z} . However, instead of sending \mathbf{z} , the prover can prove knowledge of \mathbf{z} which satisfies verification. The latter can be done in $\mathcal{O}(\log N)$ communication complexity using the recursion techniques introduced in Bulletproofs [10].

Let $\text{ck} = (h, g_1, \dots, g_n)$ be a Pedersen commitment key. Let $\mathcal{M} : \mathbb{Z}_p^N \rightarrow \mathbb{Z}_p$ be a linear map. We define a relation for linear maps as follows:

$$\mathcal{R}_{Lin}^{\text{ck}, N} = \left\{ \left((c, \mathcal{M}), \mathbf{e} \in \mathbb{Z}_p^N \right) : c = h^{\mathcal{M}(\mathbf{e})} \prod_{i=1}^N g_i^{e_i} \right\} .$$

Linear map argument of [5].

Let $\mathcal{M}_L : \mathbb{Z}_p^{N/2} \rightarrow \mathbb{Z}_p$ and $\mathcal{M}_R : \mathbb{Z}_p^{N/2} \rightarrow \mathbb{Z}_p$ be the linear maps $\mathcal{M}_L(\mathbf{X}) = \mathcal{M}(\mathbf{X} \parallel \mathbf{0}_{N/2})$ and $\mathcal{M}_R(\mathbf{X}) = \mathcal{M}(\mathbf{0}_{N/2} \parallel \mathbf{X})$. Clearly, for any $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_p^{N/2}$, it holds that $\mathcal{M}_L(\mathbf{x}) + \mathcal{M}_R(\mathbf{y}) = \mathcal{M}(\mathbf{x} \parallel \mathbf{y})$.

In Fig. 4, we recall the linear map argument of [5] with proof size of $\mathcal{O}(\log N)$. The argument is not zero-knowledge (or even honest verifier zero-knowledge), but it has knowledge soundness and logarithmic proof size. In our case, zero-knowledge is not important since the last message \mathbf{z} is public in a Sigma protocol.

B Full Description of Sub-arguments

B.1 Knowledge of committed message argument

The argument depicted in Fig. 5 proves knowledge of the message and randomness used in a commitment, formally defined as relation $\mathcal{R}_{K \circ E}^{\text{ck}, N}$.

It is well-known that the argument $\Pi_{K \circ E}^{\text{ck}, N}$ is perfectly complete, special sound, and perfectly SHVZK. In particular, the simulator works as follows:

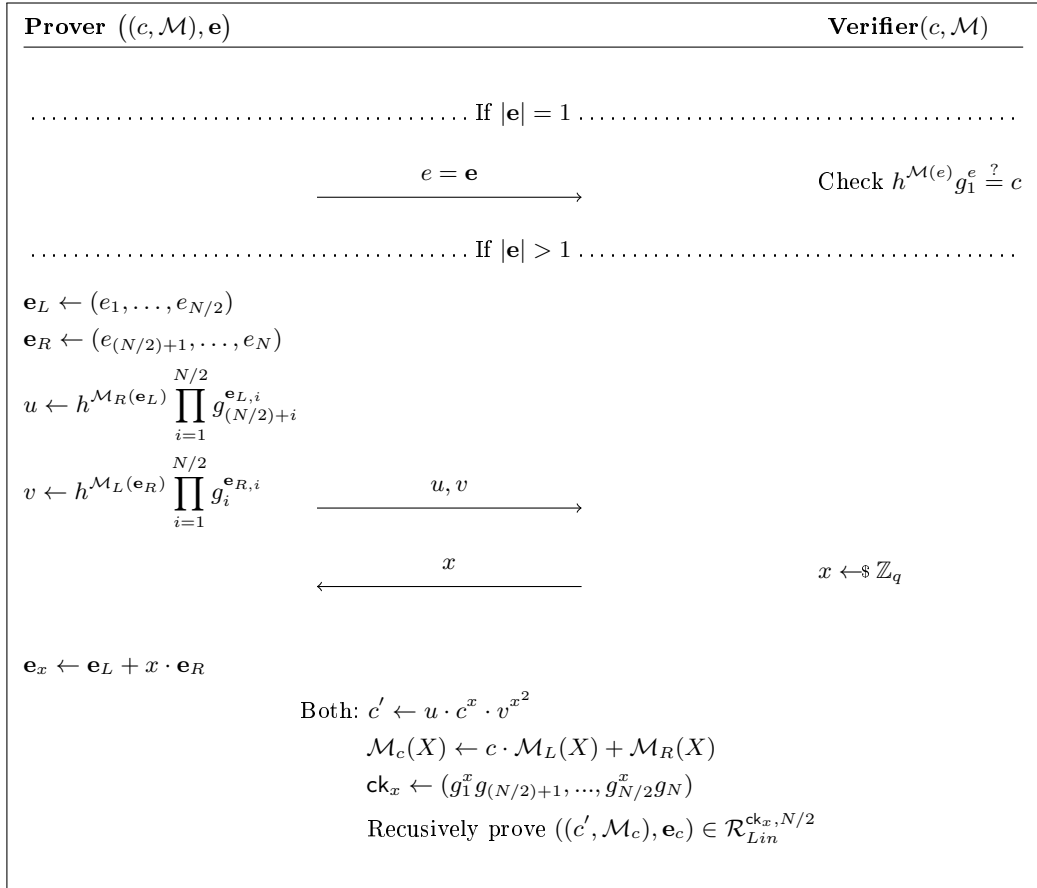


Fig. 4: Argument $\Pi_{Lin}^{\mathbf{ck}, N}$ for relation $\mathcal{R}_{Lin}^{\mathbf{ck}, N}$.

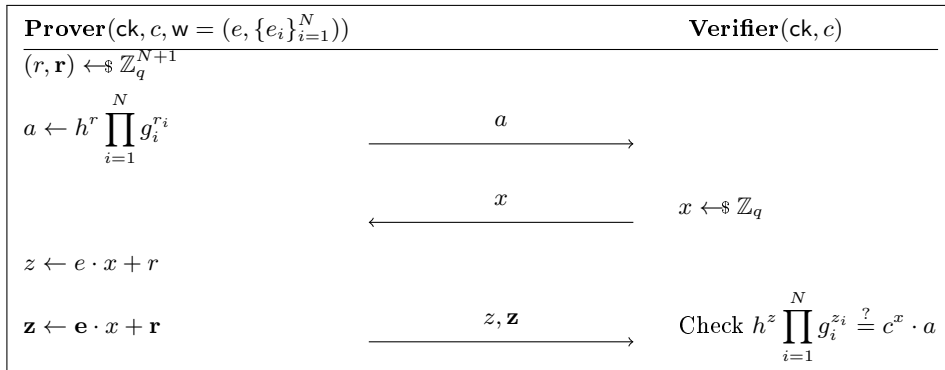


Fig. 5: Argument $\Pi_{KOE}^{\mathbf{ck}, N}$ for relation $\mathcal{R}_{KOE}^{\mathbf{ck}, N}$.

$\text{Sim}_{K \circ E}(\text{ck} = (h, \mathbf{g}), c)$
$(c, z, \mathbf{z}) \leftarrow_{\$} \mathbb{Z}_q^{N+2}$
$a \leftarrow (h^z \prod_{i=1}^N g_i^{z_i}) / c^x$
Return (a, c, z, \mathbf{z})

B.2 Inner product argument

The inner product argument is shown in Fig. 6.

Theorem 5. *Argument in Fig. 6 for the relation $\mathcal{R}_{in-prod}^{\text{ck}, N}$ is complete, specially sound, and special honest verifier zero-knowledge.*

Proof. (Perfect completeness). The completeness of the equation $h^z \prod_{i=1}^N g_i^{z_i} = ac^y$ is the same as in $\Pi_{K \circ E}$. If the prover is honest, the other equality is also easy to prove:

$$\hat{a} \hat{c}^y = H^{\sum_{i=1}^N b_i r_i} \cdot (H^{\sum_{i=1}^N b_i x_i})^y = H^{\sum_{i=1}^N b_i (r_i + y x_i)} = H^{\sum_{i=1}^N b_i z_i}.$$

(Special Soundness). Let $(a, \hat{a}, y, \mathbf{z}, z)$ and $(a, \hat{a}, y', \mathbf{z}', z')$ be two accepting transcripts for $y \neq y'$. That is

$$h^z \prod_{i=1}^N g_i^{z_i} = ac^y, \quad h^{z'} \prod_{i=1}^N g_i^{z'_i} = ac^{y'}$$

and

$$H^{\sum_{i=1}^N b_i z_i} = \hat{a} \hat{c}^y, \quad H^{\sum_{i=1}^N b_i z'_i} = \hat{a} \hat{c}^{y'}.$$

Then by dividing the respective equations, we obtain

$$h^{z-z'} \prod_{i=1}^N g_i^{z_i - z'_i} = c^{y-y'}, \quad H^{\sum_{i=1}^N b_i (z_i - z'_i)} = \hat{c}^{y-y'}.$$

This is equivalent to

$$h^{\frac{z-z'}{y-y'}} \prod_{i=1}^N g_i^{\frac{z_i - z'_i}{y-y'}} = c, \quad H^{\sum_{i=1}^N b_i (\frac{z_i - z'_i}{y-y'})} = \hat{c}.$$

Hence, it is possible to extract a witness $\delta = \frac{z-z'}{y-y'}$, $x_i = \frac{z_i - z'_i}{y-y'}$ for $i = 1, \dots, N$.

(Perfect SHVZK). We describe the simulator in the following. Let $y \leftarrow_{\$} \mathbb{Z}_p$. The simulator picks $\mathbf{z} \leftarrow_{\$} \mathbb{Z}_p^N$ and $z \leftarrow_{\$} \mathbb{Z}_p$. It then sets $a \leftarrow (h^z \prod_{i=1}^N g_i^{z_i}) / c^y$ and $\hat{a} \leftarrow (H^{\sum_{i=1}^N b_i z_i}) / \hat{c}^y$.

Elements \mathbf{z} , z are chosen uniformly randomly and independently just as in the real protocol. Now there are unique a, \hat{a} that pass the verification. Thus, simulated proof is indistinguishable from the real proof. \square

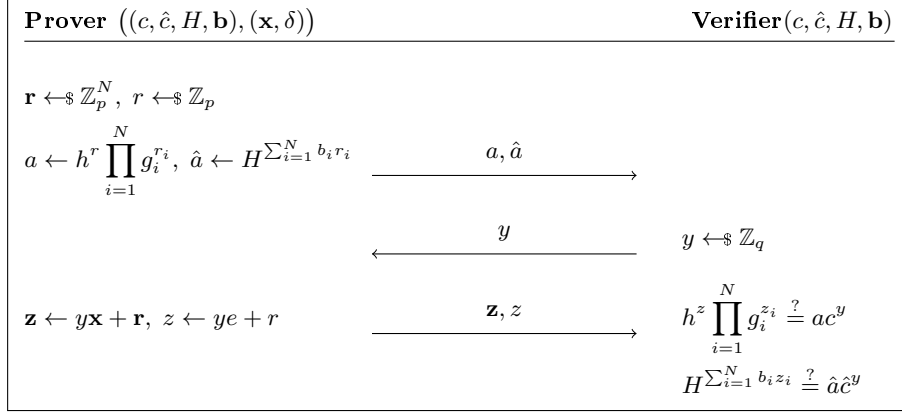


Fig. 6: Argument $\Pi_{in-prod}^{ck,N}$ for relation $\mathcal{R}_{in-prod}^{ck,N}$.

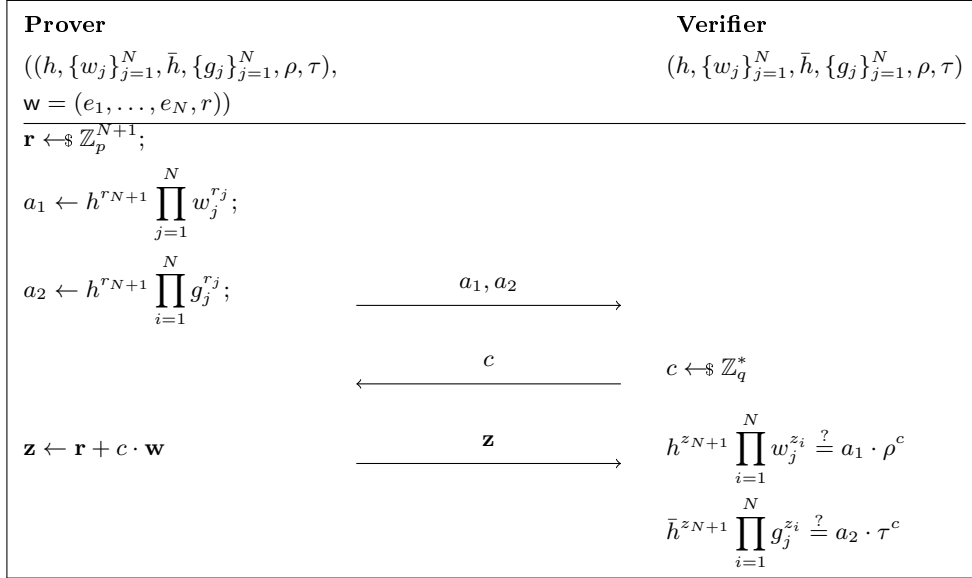


Fig. 7: The argument $\Pi_{samecom}^N$ for relation $\mathcal{R}_{samecom}^N$ showing that two commitments are equal.

B.3 Same-Message Argument

Proposition 2. *The argument Π_{samecom}^N depicted in Fig. 7 is perfectly complete, has witness-extended emulation with negligible knowledge error and is a perfect SHVZK argument of the relation $\mathcal{R}_{\text{samecom}}^N$.*

Proof. Perfect completeness of the argument is trivial.

(*Perfect SHVZK*). Perfect SHVZK follows from the perfect SHVZK of Π_{KoE} . In particular, let Sim_{KoE} be the simulator for Π_{KoE} as defined in Section 2.4. Then the simulator for Π_{samecom}^N works as follows:

$$\begin{aligned} & \text{Sim}_{\text{samecom}}(\text{ck} = (h, \mathbf{w}, \bar{h}, \mathbf{g}), \rho, \tau) \\ & \mathbf{z} \leftarrow \mathbb{Z}_q^{N+1} \\ & c \leftarrow \mathbb{Z}_q^* \\ & a_1 \leftarrow (h^{z_{N+1}} \prod_{i=1}^N w_j^{z_i}) / \rho^c \\ & a_2 \leftarrow (\bar{h}^{z_{N+1}} \prod_{i=1}^N g_j^{z_i}) / \tau^c \\ & \text{Return } (a_1, a_2, c, \mathbf{z}) \end{aligned}$$

The output of $\text{Sim}_{\text{samecom}}$ has identical distribution to Π_{samecom}^N .

(*Witness extended emulation*). By rewinding, an extractor can obtain two accepting transcripts $(a_1, a_2, c, \mathbf{z})$ and $(a_1, a_2, c', \mathbf{z}')$, where $\mathbf{z} = \mathbf{r} + c \cdot \mathbf{w}$, $\mathbf{z}' = \mathbf{r}' + c' \cdot \mathbf{w}$, and $c \neq c'$. Hence the extractor obtains $\mathbf{w} = \frac{\mathbf{z} - \mathbf{z}'}{c - c'}$. \square

B.4 Same Randomness Argument

As we mentioned in Section 3.2, $\mathcal{R}_{\text{comrand}}^N$ is a special case of $\mathcal{R}_{\text{samecom}}^N$ with $\{w_j\}_{j=1}^N = \mathbf{1}$. Hence, we omit the security proof for Π_{comrand}^N , and show it in Fig. 8 for completeness sake.

Proposition 3. *The argument Π_{comrand}^N depicted in Fig. 8 is perfectly complete, has witness-extended emulation with negligible knowledge error and is a perfect SHVZK argument of the relation $\mathcal{R}_{\text{comrand}}^N$.*

B.5 Weighted Sum Argument

Proposition 4. *Let $\hat{h}, g_1, \dots, g_N, \hat{H}, \hat{G} \in \mathbb{G}$ be such that finding any nontrivial DLREs between them is hard. Let Π_{samecom}^N and Π_{KoE}^N have witness-extended emulation with negligible knowledge error and be special honest-verifier zero-knowledge. Then the argument Π_{comsum}^N depicted in Fig. 9 is also SHVZK and has witness-extended emulation with negligible knowledge error.*

Proof. Let us extract the values (r', e'_1, \dots, e'_N) from the $\mathcal{R}_{\text{samecom}}^N$ protocol and the value r'' from the KoE protocol. We thus have that

$$\frac{\sigma}{\hat{G}^v} = \hat{H}^{r''}$$

Prover	Verifier
$((H, \bar{h}, \{g_j\}_{j=1}^N, \rho, \tau),$ $\mathbf{w} = (e_1, \dots, e_N, r))$	$(H, \bar{h}, \{g_j\}_{j=1}^N, \rho, \tau)$
$\mathbf{r} \leftarrow \mathbb{Z}_p^{N+1};$	
$a_1 \leftarrow H^{r_{N+1}};$	
$a_2 \leftarrow h^{r_{N+1}} \prod_{i=1}^N g_i^{r_i};$	$\xrightarrow{a_1, a_2}$
	\xleftarrow{c}
$\mathbf{z} \leftarrow \mathbf{r} + c \cdot \mathbf{w}$	$c \leftarrow \mathbb{Z}_q^*$
	$\xrightarrow{\mathbf{z}}$
	$H^{z_{N+1}} \stackrel{?}{=} a_1 \cdot \rho^c$ $\bar{h}^{z_{N+1}} \prod_{i=1}^N g_i^{z_i} \stackrel{?}{=} a_2 \cdot \tau^c$

Fig. 8: The argument $\Pi_{comrand}^N$ for relation $\mathcal{R}_{comrand}^N$ showing that the randomness of a commitment is the same as the logarithm of a given value.

Prover $((\hat{h}, g_1, \dots, g_N \in \mathbb{G}, a_1, \dots, a_N,$	Verifier $((\hat{h}, g_1, \dots, g_N \in \mathbb{G},$
$v \in \mathbb{Z}_q, \tau, \hat{H}, \hat{G}), \mathbf{w} = \{r, e_1, \dots, e_N \in \mathbb{Z}_q\})$	$a_1, \dots, a_N, v \in \mathbb{Z}_q, \tau, \hat{H}, \hat{G})$
$\sigma \leftarrow \hat{H}^r \prod_{j=1}^N (\hat{G}^{a_j})^{e_j}$	
$\xrightarrow{\sigma}$	
Prove: $((\hat{H}, \hat{G}^{a_1}, \dots, \hat{G}^{a_N}, \hat{h}, g_1, \dots, g_N, \sigma, \tau),$ $(r, e_1, \dots, e_N \in \mathbb{Z}_q)) \in \mathcal{R}_{samecom}^N$	
Prove: $((\hat{H}, \frac{\sigma}{\hat{G}^v}, (r)) \in \mathcal{R}_{K \circ E}$	
Accept if the proofs go through	

Fig. 9: The argument showing that the (weighted) sum of commitments is equal to a given value.

and

$$\sigma = \hat{H}^{r'} \prod_{j=1}^N (\hat{G}^{a_j})^{e_j} = \hat{H}^{r'} \hat{G}^{\sum_{j=1}^N a_j e_j}.$$

Thus we get that

$$\hat{H}^{r''} \hat{G}^v = \hat{H}^{r'} \hat{G}^{\sum_{j=1}^N a_j e_j}.$$

By the assumption that no nontrivial DLOG relations are known, we have that $r'' = r'$ and $v = \sum_{j=1}^N a_j e_j$. \square

B.6 Argument showing same message in several commitments

As we mentioned in Section 3.4, $\mathcal{R}_{samemes}^N$ is a general case of $\mathcal{R}_{samecom}^N$ where 2 commitments becomes v commitments. Hence, we omit the argument depiction and the security proof for $\Pi_{samemes}^N$.

C Additional Lemmas and Proofs

C.1 Full proofs and missing proofs from the main body

Lemma 3. *Let $F \subset \mathbb{Z}_q$ have polynomial size, and let k be a positive integer. Let \mathcal{P} be a protocol between prover P and verifier V where the first message of V is a randomly chosen challenge $(a_1, \dots, a_k) \in F^k$. Let the probability that V accepts be at least $\frac{1}{|F|} + \varepsilon$ where ε is non-negligible. Suppose that for a given challenge (a_1, \dots, a_k) the verifier either accepts or rejects in expected time t . Then there is an algorithm \mathcal{E}' that for any $t \in [1, k]$, runs in expected time $\frac{|F|^2}{\varepsilon} t$ that finds two challenges (a_1, \dots, a_k) and (a'_1, \dots, a'_k) where the verifier accepts for both of them and where (a_1, \dots, a_k) and (a'_1, \dots, a'_k) differ only at position t .*

Proof. The strategy of \mathcal{E}' would be simply picking $(a_1, \dots, a_{t-1}, a_{t+1}, \dots, a_k)$ at random, and then testing all possible values in F to see whether there are at least two possible accepting values for a_t . If this is not the case, \mathcal{E}' would pick a new set $(a_1, \dots, a_{t-1}, a_{t+1}, \dots, a_k)$ at random, and repeat, until the desired object is obtained. Let us now analyze the expected running time of this algorithm.

Let us fix the random coin used by P and V after (a_1, \dots, a_k) is chosen. Then, for any (a_1, \dots, a_k) , the verifier accepts with probability either 0 or 1.

Let us denote the number of $(k-1)$ -tuples of elements $(a_1, \dots, a_{t-1}, a_{t+1}, \dots, a_k)$ from F where there exist exactly r values for a_t such that in \mathcal{P} , $(a_1, \dots, a_{t-1}, a_t, a_{t+1}, \dots, a_k)$ is an accepting argument, by v_r . That is,

$$v_r := |\{(a_1, \dots, a_{t-1}, a_{t+1}, \dots, a_k) \in F^{k-1} : (\exists^r a_t \text{ s.t. } (a_1, \dots, a_k) \text{ is accepted})\}|.$$

We now note that $\sum_{r=0}^{|F|} v_r = F^{k-1}$ and that

$$\frac{\sum_{r=0}^{|F|} r v_r}{|F|^k} = \frac{1}{|F|} + \varepsilon,$$

as we are simply counting the number of accepting inputs in the numerator.

Suppose that $\frac{\sum_{r=2}^{|F|} v_r}{\sum_{r=0}^{|F|} v_r} < \frac{\varepsilon}{|F|}$. Then we have that

$$\frac{1}{|F|} + \varepsilon = \frac{\sum_{r=0}^{|F|} r v_r}{|F|^k} \leq \frac{0 \cdot v_0 + 1 \cdot v_1}{|F|^k} + \frac{\sum_{r=2}^{|F|} |F| v_r}{|F|^k} < \frac{|F|^{k-1}}{|F|^k} + |F| \frac{\varepsilon}{|F|} = \frac{1}{|F|} + \varepsilon,$$

a contradiction. Hence $\frac{\sum_{r=2}^{|F|} v_r}{\sum_{r=0}^{|F|} v_r} \geq \frac{\varepsilon}{|F|}$

We thus note that the probability of choosing such a set $(a_1, \dots, a_{t-1}, a_{t+1}, \dots, a_k)$, such that there are at least 2 possible values for a_t such that (a_1, \dots, a_k) is accepted, is $\frac{\sum_{r=2}^{|F|} v_r}{\sum_{r=0}^{|F|} v_r} \geq \frac{\varepsilon}{|F|}$. Thus, \mathcal{E}' would obtain such a pair of values \mathbf{a}, \mathbf{a}' in expected number of tries $\frac{|F|}{\varepsilon}$ with each try taking $|F|t$ time, giving us the result.

Before restating Theorem 2 we will prove the following helpful lemma.

Lemma 4. *Let $a, b \in P_{p,f} - P_{p,f}$. Let $(|c| + |d|) \cdot (\sum_{i=0}^f p^i) \leq q$. Let p be even. Let $ac = bd$. Let $|c|, |d| < p - 1$. Let $c, d \neq 0$. Then $a = \pm b$.*

Proof. Assume that this is not the case. W.l.o.g, let $a, b > 0$, and $a > b$. Let $a = \sum_{i=0}^l a_i p^i$ and $b = \sum_{i=0}^{l'} b_i p^i$ where $a_l = b_{l'} = 1$ and all a_i and b_i are in $\{-1, 0, 1\}$. Let us split the proof into two cases — either $l = l'$ or not.

First, we will consider the case where $l = l'$. If $l = 0$, then $a = 1$ and $b = 1$ and thus we have a contradiction. Thus we will assume that $l > 0$. We have that $ac = bd$, thus $cp^l + \sum_{i=0}^{l-1} ca_i p^i = dp^l + \sum_{i=0}^{l-1} db_i p^i$, i.e $p^l(d - c) = \sum_{i=0}^{l-1} (ca_i - db_i) p^i$. If $d - c > 1$, then we will show that $\sum_{i=0}^{l-1} (ca_i - db_i) p^i < 2p^l$ which is a contradiction.

More precisely, note that $\sum_{i=0}^{l-1} (ca_i - db_i) p^i \leq \sum_{i=0}^{l-1} (|c| + |d|) p^i \leq (2p - 4) \sum_{i=0}^{l-1} p^i$. If $l = 1$, then $(2p - 4) \sum_{i=0}^{l-1} p^i = 2p - 4 < 2p$.

Otherwise we note that $(2p - 4) \sum_{i=0}^{l-1} p^i = \sum_{i=1}^l 2p^i - \sum_{i=0}^{l-1} 4p^i = 2p^l - 2 \sum_{i=1}^{l-1} 2p^i - 4 < 2p^l$. Hence, in the case of $d - c > 1$ we have a contradiction.

If $d - c = 1$, then note that $c \not\equiv \pm d \pmod{p}$, because c and d have different parities. Thus we can apply the second clause of Lemma 6, and obtain that $a = b = 0$, a contradiction.

Secondly, we will consider the case where $l > l'$. We will show that $\frac{a}{b} > p - 2$. Denote by $a' := p^l - \sum_{i=0}^{l-1} p^i$ and by $b' := \sum_{i=0}^{l-1} p^i$. Clearly $a' \leq a$ and $b' \geq b$. Thus $\frac{a}{b} \geq \frac{a'}{b'}$. Now

$$\frac{a'}{b'} = \frac{p^l - \sum_{i=0}^{l-1} p^i}{\sum_{i=0}^{l-1} p^i} = \frac{p^l - \frac{p^l - 1}{p - 1}}{\frac{p^l - 1}{p - 1}} = \frac{p^l(p - 1)}{p^l - 1} - 1 > \frac{p^l(p - 1)}{p^l} - 1 = p - 2.$$

Thus $\frac{a}{b} > p - 2$, which leads to a contradiction.

Thus the result holds.

Here we restate Theorem 2 again and prove it.

Theorem 6. *Let $p, 2^{K+1} < N$. Suppose that the $SIS_{q,N,v,(N+1) \cdot (\sum_{i=0}^f p^i)}$ holds. If p is odd, then let $2^K \leq \frac{p}{2}$. If p is even, then let $2^K \leq p - 2$. Let $T = \min\{2^K, (\frac{8}{5})^f\}$. The protocol depicted in 3 has $(\frac{1}{T} + \epsilon_1)^K + \epsilon_2$ -witness-extended-emulation where ϵ_1 and ϵ_2 are negligible.*

Proof. From the $(N + 1) \cdot (\sum_{i=0}^f p^i)$ -SIS-assumption, it follows that for the $\{[w'_{j,k}]\}_{j=1,k=1}^{N,v}$ the **NoBounded**- $(N + 1) \cdot (\sum_{i=0}^f p^i)$ -*DLRel* holds. Suppose that there is a prover who has success rate more than $(\frac{1}{T} + \epsilon)^K$ where ϵ is non-negligible. By soundness amplification principles, it follows that the probability to pass both the ChalResp protocols, for randomly picked challenges, is at least $\frac{1}{T} + \epsilon$.

Now, thus both the individual ChalResp protocols must also have at least $\frac{1}{T} + \epsilon$ chance of passing. Consider first the ChalResp with the challenge set $[0, 2^K - 1]$. Now for every $i \in [1, N]$ we do the following. Fix at random $\{b_{i,j}\}_{j \in [1, N] \setminus \{i\}}$ from the elements of $[0, 2^K - 1]$. Now for $b_{i,i}$ there are in expectation, $\frac{1}{\frac{1}{T} + \epsilon}$ values for which the prover is successful for the challenge $\{b_{i,j}\}_{j=1}^N$. Thus for every consecutive T values, at least with a chance ϵT^2 , there are two values for $b_{i,i}$ for which the proof is successful. Let those values be $b_{i,i}$ and $b_{i,i} + a_i$, w.l.o.g $a_i > 0$. Let the responses we extract to these challenges, as per Lemma 1 be $\{d'_{i,j}\}_{j=1}^N$ and $\{d''_{i,j}\}_{j=1}^N$, with all $d'_{i,j}, d''_{i,j} \in [0, 2^K - 1]$. We get that $\prod_{j=1}^N [w'_{j,k}]^{d'_{i,j}} = \prod_{t=1}^N [\hat{w}_{t,k}]^{b_{i,t}}$ and $\prod_{j=1}^N [w'_{j,k}]^{d''_{i,j}} = [\hat{w}'_{i,k}]^{a_i} \prod_{t=1}^N [\hat{w}_{t,k}]^{b_{i,t}}$ and where $\sum_{t=1}^N b_{i,t} = \sum_{j=1}^N d'_{i,j}$ and $a_i + \sum_{t=1}^N b_{i,t} = \sum_{j=1}^N d''_{i,j}$. Deducting those equations from each other and denoting $d_{i,j} := d''_{i,j} - d'_{i,j}$, we have that

$$[\hat{w}'_{i,k}]^{a_i} = \prod_{j=1}^N [w'_{j,k}]^{d_{i,j}}. \quad (2)$$

If for some i , $\gcd(a_i, \{d_{i,j}\}_{j=1}^N) = A_i \neq 1$ over \mathbb{Z} , then let us divide all those elements with that common divisor A_i . Thus we will assume that $\gcd(a_i, \{d_{i,j}\}_{j=1}^N) = 1$. Note that thus the "original" extracted values will be $a_i A_i$ and $\{A_i d_{i,j}\}_{j=1}^N$.

Analogously, from the ChalResp for the other extraction principles we are able to extract $\{\bar{d}_{i,j}\}_{i=1,j=1}^{N,N}$ and \bar{a}_i such that

$$[\hat{w}'_{i,k}]^{\bar{a}_i} = \prod_{j=1}^N [w'_{j,k}]^{\bar{d}_{i,j}} \quad (3)$$

where $\bar{a}_i, \bar{d}_{i,j} \in P_{p,f} - P_{p,f}$.

Analogously, if for some i , $\gcd(\bar{a}_i, \{\bar{d}_{i,j}\}_{j=1}^N) = \bar{A}_i \neq 1$ over \mathbb{Z} , then let us divide all those elements with that common divisor \bar{A}_i , thus we will assume from now on that $\gcd(\bar{a}_i, \{\bar{d}_{i,j}\}_{j=1}^N) = 1$. Likewise, note here as well that the "original" values in the set $P_{p,f} - P_{p,f}$ will be $\bar{A}_i \bar{a}_i, \{\bar{A}_i \bar{d}_{i,j}\}_{j=1}^N$.

Let $a_i = \alpha_i \cdot c_i$ and $\bar{a}_i = \alpha_i \cdot \bar{c}_i$ where α_i is the greatest common denominator of a_i and \bar{a}_i . Then c_i and \bar{c}_i are coprime and $\bar{a}_i c_i = \bar{c}_i a_i$.

Now take the equations 2 and 3 to the powers \bar{c}_i and c_i respectively. We will obtain that

$$\prod_{j=1}^N [w'_{j,k}]^{d_{i,j} c_i} = [\hat{w}'_{i,k}]^{\bar{a}_i c_i} = [\hat{w}'_{i,k}]^{\bar{c}_i a_i} = \prod_{j=1}^N [w'_{j,k}]^{d_{i,j} \bar{c}_i}$$

and hence $\prod_{j=1}^N [w'_{j,k}]^{d_{i,j} c_i - d_{i,j} \bar{c}_i} = 1$. Because $|d_{i,j} c_i - d_{i,j} \bar{c}_i| \leq 2^{K+1} (\sum_{i=0}^f p^i) \leq (N+1) \cdot (\sum_{i=0}^f p^i)$, then it follows that $d_{i,j} c_i - d_{i,j} \bar{c}_i = 0$ for all i and j .

Now, because $d_{i,j} c_i = d_{i,j} \bar{c}_i$ and c_i and \bar{c}_i are coprime, it follows that \bar{c}_i must divide $d_{i,j}$ and c_i must divide $d_{i,j}$. However, this means that $\gcd(a_i, \{d_{i,j}\}_{j=1}^N) = c_i$ and $\gcd(\bar{a}_i, \{d_{i,j}\}_{j=1}^N) = \bar{c}_i$. Hence $c_i = 1 = \bar{c}_i$ for all i . This means that actually $a_i = \alpha_i = \bar{a}_i$ and also $d_{i,j} = \bar{d}_{i,j}$ for all i, j .

We have that $a_i A_i \in [1, p-2]$ and $\bar{a}_i \bar{A}_i \in P_{p,f} - P_{p,f}$. Note that also $a_i \in [1, p-2]$. We now want to show that $a_i = 1$. Now, if $\bar{A}_i = 1$, then we have that $a_i \in [1, p-2] \cap P_{p,f} - P_{p,f} = \{1\}$ and thus $a_i = 1$. However, suppose that $\bar{A}_i \neq 1$. Consider some $d_{i,j} \neq 0$ which must exist, otherwise $\gcd(a_i A_i, \{d_{i,j}\}_{j=1}^N) = a_i A_i$ and thus $a_i = 1$. W.l.o.g let it be $d_{i,1}$. Then, we have that $a_i, d_{i,1} \in [-2^K + 1, 2^K - 1] \subseteq [-\frac{p}{2}, \frac{p}{2})$ and $a_i \bar{A}_i, d_{i,1} \bar{A}_i \in P_{p,f} - P_{p,f}$.

We have that $\frac{a_i}{d_{i,1}} = \frac{a_i \bar{A}_i}{d_{i,1} \bar{A}_i}$. Denote $x_1 := a_i \bar{A}_i$ and $x_2 := d_{i,1} \bar{A}_i$ we get that $a_i x_2 = d_{i,1} x_1$ with $x_1, x_2 \in P_{p,f} - P_{p,f}$. We have that $a_i, d_{i,1} \neq 0$. Thus, if p is odd, then by Lemma 9, and if p is even then by Lemma 4, we get that $|a_i| = |d_{i,1}|$. Note however, that the same argument can be made for any nonzero $d_{i,j}$. Hence, all nonzero $d_{i,j}$ divide a_i . However, by construction $\gcd(a_i, \{d_{i,j}\}_{j=1}^N) = 1$. Thus it follows that $|a_i| = 1$, and because we chose it so that $a_i > 0$, $a_i = 1$. Additionally, all $d_{i,j} \in \{-1, 0, 1\}$.

From the last claim of Lemma 1, we also get that for any extracted equation, the sums of the exponents must match on both sides. From there on it is not difficult to obtain that $\sum_j d_{i,j} = a_i$. Because $a_i = 1$, $\sum_j d_{i,j} = 1$ for all i .

Thus $a_i = 1$ for all i . Additionally, it follows that $d_{i,j} = \bar{d}_{i,j}$ and thus all $d_{i,j} \in [-2^K + 1, 2^K - 1] \cap P_{p,f} - P_{p,f}$.

Thus we get that

$$[\hat{w}_{i,k}]' = \prod_{j=1}^N [w'_{j,k}]^{d_{i,j}} \quad (4)$$

for all i, k .

From the first proof of the $\mathcal{R}_{in-prod}^{ck,N}$, we can get that $\prod_{i=1}^N [\hat{w}'_{i,k}] = \prod_{j=1}^N [w'_{j,k}]$. This is because we can extract from it values δ'_k and $r''_{i,k}$ such that $C_k = h^{\delta'_k} \prod_{i=1}^N g_i^{r''_{i,k}}$ and that $\frac{\prod_{i=1}^N [\hat{w}_i]}{(\prod_{j=1}^N [w_j]) h^{\sum_{j=1}^N a_{j,k}}} = h^{\sum_{i=1}^N r''_{i,k}}$. We note that these values extracted from C_k must be the same values that are extracted from it in other parts of the proof, thus we must have that $[\hat{w}'_{i,k}] = \frac{[\hat{w}_i]}{h^{r''_{i,k}}}$. Also $[w'_{j,k}] = [w_j] h^{a_{j,k}}$.

Now when we reorganize $\frac{\prod_{t=1}^N [\hat{w}_t]}{(\prod_{j=1}^N [w_j]) h^{\sum_{j=1}^N a_{j,k}}} = h^{\sum_{i=1}^N r''_{i,k}}$, then $\prod_{i=1}^N [\hat{w}'_{i,k}] = \prod_{j=1}^N [w'_{j,k}]$ follows.
 Note now that

$$\prod_{j=1}^N [w_{j,k}]' = \prod_{t=1}^N [\hat{w}_{t,k}]' = \prod_{t=1}^N \left(\prod_{j=1}^N [w'_{j,k}]^{d_{t,j}} \right) = \prod_{j=1}^N [w_{j,k}]'^{\sum_{t=1}^N d_{t,j}}.$$

Thus it must also hold for all j that

$$\sum_{t=1}^N d_{t,j} = 1.$$

Previously we had that for all t ,

$$\sum_{j=1}^N d_{t,j} = 1.$$

Now, suppose that the ChalResp with the challenge set $[0, 2^K - 1]$ one gets a challenge (b_1, \dots, b_N) where all $b_i \in [0, 2^K - 1]$, and successfully answers it with some (e_1, \dots, e_N) where all $e_j \in [0, 2^K - 1]$. Thus

$$\prod_{i=1}^N [w_{j,k}]'^{e_j} = \prod_{i=1}^N [\hat{w}_{i,k}]'^{b_i} = \prod_{i=1}^N \left(\prod_{j=1}^N ([w'_{j,k}]^{d_{i,j}})^{b_i} \right) = \prod_{i=1}^N [w_{j,k}]'^{\sum_{i=1}^N b_i d_{i,j}}.$$

We have that

$$\left| e_j - \sum_{i=1}^N b_i d_{i,j} \right| \leq |e_j| + \sum_{i=1}^N |b_i| |d_{i,j}| \leq 2^K + \sum_{i=1}^N 2^K \cdot 1 \leq (N+1)2^K.$$

Because $(N+1)2^K \leq N \cdot p^{f+1}$ the **NoBounded- $N \cdot p^{f+1}$ -DLRel** property, it must hold that for all j , $e_j = \sum_{i=1}^N b_i d_{i,j}$. This implies that $\sum_{i=1}^N b_i d_{i,j} \in [0, 2^K - 1]$. Thus, if for a randomly chosen (b_1, \dots, b_N) it happens that for some j , $\sum_{i=1}^N b_i d_{i,j} \notin [0, 2^K - 1]$, then the prover is unable to answer this challenge. Analogously, for the challenge set $P_{p,f}$, one does the same argument and obtains that if for some randomly chosen $(\bar{b}_1, \dots, \bar{b}_N)$ where all $\bar{b}_i \in P_{p,f}$, if for some j , $\sum_{i=1}^N \bar{b}_i d_{i,j} \notin P_{p,f}$, then the prover is unable to answer this challenge. (In this case the corresponding $|\bar{e}_j - \sum_{i=1}^N \bar{b}_i d_{i,j}|$ in the exponent will analogously be upper-bounded by $(N+1) \sum_{i=0}^f p^i$)

Denote $D := \{d_{i,j}\}_{i=1,j=1}^{N,N}$. We have that all rows and columns of D must sum to 1. Now consider Theorem 8. Suppose that there is some column vector of D that is not a unit vector. In that case, if b_1, \dots, b_n are randomly chosen from $P_{p,f}$, then the probability that all the $\sum_{i=1}^N \bar{b}_i d_{i,j} \notin P_{p,f}$ is at most $(\frac{5}{8})^f$. This implies that the probability of the prover passing is upper-bounded by $(\frac{5}{8})^f > \frac{1}{T}$

which contradicts our assumption. Thus all columns of D must be unit vectors. On the other hand, because each row must sum to 1, these values 1 must be in separate rows. Thus D is a permutation matrix. Thus we have successfully extracted the permutation matrix. Let it describe a permutation π' such that $D_{\pi'(j),j} = 1$ and all the other entries are zeroes. Now let us also extract the rerandomization factors.

Plugging the equation into the equation 4 we obtain that

$$[\hat{w}_{i,k}]' = \prod_{j=1}^N [w'_{j,k}]^{d_{i,j}} = \left(\prod_{i \neq \pi'(j)} [w'_{j,k}]^0 \right) [w'_{\pi'^{-1}(i),k}]^1 = [w'_{\pi'^{-1}(i),k}] \quad (5)$$

Taking $\pi'^{-1}(i) =: j$, we have that $[w'_{j,k}] = [\hat{w}_{\pi'(j),k}]'$ for all j, k . We have by definition that $[w'_{j,k}] = h^{a_{j,k}} [w_j]$. On the other hand, we denoted $[\hat{w}_{t,k}]' := \frac{[\hat{w}_t]}{h^{r'_{t,k}}}$. Thus the equation becomes

$$[\hat{w}_{\pi'(j)}] h^{-r'_{\pi'(j),k}} h^{-a_{j,k}} = [w_j].$$

We note that from this it follows that for a fixed j , for all k , the value $r'_{\pi'(j),k} + a_{j,k}$ is the same. We will denote this value by r'_j . Thus we have extracted $\{r'_j\}_{j=1}^N$ and a π' such that

$$[\hat{w}_{\pi'(j)}] = h^{r'_j} [w_j].$$

such that $D_{\pi'(j),j} = 1$ and all the other entries are zeroes. Plugging this into the equation 4 we obtain that

$$[\hat{w}_{i,k}]' = \prod_{j=1}^N [w'_{j,k}]^{d_{i,j}} = \left(\prod_{i \neq \pi'(j)} [w'_{j,k}]^0 \right) [w'_{\pi'^{-1}(i),k}]^1 = [w'_{\pi'^{-1}(i),k}] \quad (6)$$

Taking $\pi'^{-1}(i) =: j$, we have that $[w'_{j,k}] = [\hat{w}_{\pi'(j),k}]'$ for all j, k . We have by definition that $[w'_{j,k}] = h^{a_{j,k}} [w_j]$. On the other hand, we denoted $[\hat{w}_{t,k}]' := \frac{[\hat{w}_t]}{h^{r'_{t,k}}}$. Thus the equation becomes

$$[\hat{w}_{\pi'(j)}] h^{-r'_{\pi'(j),k}} h^{-a_{j,k}} = [w_j].$$

We note that from this it follows that for a fixed j , for all k , the value $r'_{\pi'(j),k} + a_{j,k}$ is the same. We will denote this value by r'_j . Thus we have extracted $\{r'_j\}_{j=1}^N$ and a π' such that

$$[\hat{w}_{\pi'(j)}] = h^{r'_j} [w_j].$$

We thus have successfully extracted a permutation along with rerandomization factors. \square

Prover ($\{c_i\}_{i=1}^N, \{\hat{c}_i\}_{i=1}^N, h, g,$ $\mathbb{Z}_q, K, v, \mathbf{w} = \{\pi, \{r_i\}_{i=1}^N\}$)	Verifier ($\{c_i\}_{i=1}^N,$ $\{\hat{c}_i\}_{i=1}^N, h, g, \mathbb{Z}_q, K, v$)
	$(a_{1,1}, a_{1,2}, \dots, a_{N,v}) \leftarrow \mathbb{Z}_q$
$\longleftarrow (a_{1,1}, a_{1,2}, \dots, a_{N,v})$	
	$H \leftarrow \mathbb{G}$
$\longleftarrow H$	
$r_{i,k} \leftarrow r_i - a_{\pi^{-1}(i),k}$ for $i = 1, \dots, N$	
	$g_1, \dots, g_N \leftarrow \mathbb{G}$
$\longleftarrow \mathbf{ck} = (h, g_1, \dots, g_N)$	
$\delta_1, \dots, \delta_k \leftarrow \mathbb{Z}_q$ $C_k \leftarrow h^{\delta_k} \prod_{i=1}^N g_i^{r_{i,k}}$	
$\longrightarrow C_1, \dots, C_v$	
	$\eta \leftarrow \mathbb{Z}_q$
$\longleftarrow \eta$	
<u>Both: $[w_i] \leftarrow c_{i,1} c_{i,2}^\eta, [\hat{w}_i] \leftarrow \hat{c}_{i,1} \hat{c}_{i,2}^\eta, \forall i \in [1, N], \hat{h} \leftarrow gh^\eta$</u>	
Prove: $((C_k, \frac{\prod_{t=1}^N [\hat{w}_t]}{(\prod_{j=1}^N [w_j])^{\hat{h}^{\sum_{j=1}^N a_{j,k}}}}, \hat{h}, (1, \dots, 1)),$ $(r_{i,k}, \delta_k) \in \mathcal{R}_{in-prod}^{ck,N}$ for $k \in [1, v]$	
$\longleftarrow \hat{h}_1, \dots, \hat{h}_v, g_{1,1}, \dots, g_{N,v}$	$\hat{h}_1, \dots, \hat{h}_v, g_{1,1}, \dots, g_{N,v} \leftarrow \mathbb{G}$
<u>Both for $i = 1, \dots, N, k = 1, \dots, v$:</u>	
$[w'_{i,k}] \leftarrow \hat{h}^{a_{i,k}} [w_i],$ Repeat the following algorithm κ times in parallel, batched:	
$ChalResp((\{[w'_{i,k}]\}_{i=1,k=1}^{N,v}, \{a_{i,k}\}_{i=1,k=1}^{N,v}, \{[\hat{w}_i]\}_{i=1}^N, \hat{h}, g, \mathbf{ck},$ $\mathbb{Z}_q, [0, 2^K - 1], v, \{C_k\}_{k=1}^v, \{\hat{h}_k\}_{k=1}^v, \{g_{i,k}\}_{i=1,k=1}^{N,v})$ $(\pi, \{r_i\}_{i=1}^N, \{r_{i,k}\}_{i=1,k=1}^{N,v}, \{\delta_k\}_{k=1}^v)$	
$ChalResp((\{[w'_{i,k}]\}_{i=1,k=1}^{N,v}, \{a_{i,k}\}_{i=1,k=1}^{N,v}, \{[\hat{w}_i]\}_{i=1}^N, \hat{h}, g, \mathbf{ck},$ $\mathbb{Z}_q, P_{p,f}, v, \{C_k\}_{k=1}^v, \{\hat{h}_k\}_{k=1}^v, \{g_{i,k}\}_{i=1,k=1}^{N,v})$ $(\pi, \{r_i\}_{i=1}^N, \{r_{i,k}\}_{i=1,k=1}^{N,v}, \{\delta_k\}_{k=1}^v)$	
Accept if all proofs go through	

Fig. 10: The shuffling protocol for ElGamal.

C.2 Proof of ElGamal shuffle

The protocol for proving an ElGamal shuffle will be the following.

Theorem 7. *Let $p, 2^{K+1} < N$. Suppose that the $SIS_{q,N,v,(N+1) \cdot (\sum_{i=0}^f p^i)}$ holds. If p is odd, then let $2^K \leq \frac{p}{2}$. If p is even, then let $2^K \leq p - 2$. Let $T = \min\{2^K, (\frac{8}{5})^f\}$. Let $v \geq 2$. The protocol in 10 is $(\frac{1}{T} + \varepsilon)^\kappa + \varepsilon_2$ -sound.*

Proof. Fix a value η . We carry out the proof similarly to the one of the previous theorem. Denote with the subscript η the extracted values in the extraction where η was used. We see thus that we are able to extract values $(r'_i)_\eta, (r'_{t,k})_\eta, (\delta_k)_\eta$ and $(\pi')_\eta$ such that, among other things

$$[\hat{w}_{\pi'(j)}]_\eta = h^{r'_{j,\eta}}[w_j]_\eta$$

and

$$r'_{j,\eta} = a_{j,k} + r_{\pi(j),k,\eta}$$

and

$$C_k = h^{\delta'_{k,\eta}} \prod_{t=1}^N g_t^{r'_{t,k,\eta}}.$$

Now let's rewind until before η is chosen. We have that the C_k do not change. Thus we have that $r'_{t,k,\eta} = r'_{t,k,\eta'}$ and $\delta'_{k,\eta} = \delta'_{k,\eta'}$ for all η, η' . Let us denote $r'_{t,k,\eta} =: r'_{t,k}$.

We have that the $a_{i,k}$ were randomly chosen from \mathbb{Z}_q . We can assume that $\frac{N^2}{q}$ is negligibly small. Thus, for all k , with overwhelming probability, there do not exist such i_1, j_1, i_2, j_2 where $i_1 \neq j_1$ and $i_2 \neq j_2$ but $a_{i_1,k} - a_{j_1,k} = a_{i_2,k} - a_{j_2,k}$.

We now have that

$$r'_{j,\eta} = a_{j,1} + r_{\pi_\eta(j),1} = \dots = a_{j,v} + r_{\pi_\eta(j),v}.$$

Suppose now that for some η_1, η_2 , the extracted permutations are not equal, i.e. there is a u such that $\pi_{\eta_1}(u) \neq \pi_{\eta_2}(u)$. Let v be such an element that $\pi_{\eta_2}(v) = \pi_{\eta_1}(u)$, by our assumption $u \neq v$.

We have that

$$a_{u,1} + r_{\pi_{\eta_1}(u),1} = a_{u,2} + r_{\pi_{\eta_1}(u),2},$$

i.e.

$$a_{u,1} - a_{u,2} = r_{\pi_{\eta_1}(u),2} - r_{\pi_{\eta_1}(u),1} = r_{\pi_{\eta_2}(v),2} - r_{\pi_{\eta_2}(v),1} = a_{v,1} - a_{v,2},$$

where the last equation holds because

$$a_{v,1} + r_{\pi_{\eta_2}(v),1} = a_{v,2} + r_{\pi_{\eta_2}(v),2} = r_{v,\eta_2}.$$

We have now obtained that $a_{u,1} - a_{u,2} = a_{v,1} - a_{v,2}$, where $u \neq v$, which contradicts our assumption that this happens only with a negligible probability. Thus we have that $\pi_{\eta_1} = \pi_{\eta_2}$ for all η . We will thus denote $r'_{j,\eta} =: r'_j$.

From the WEE-proof of the commitments-proof we have that $c_i = \hat{c}_{\pi(i)} \hat{h}^{r'_{\pi(i)}}$. Therefore,

$$c_{i,1} \cdot c_{i,2}^\eta = \hat{c}_{\pi(i),1} \cdot \hat{c}_{\pi(i),2}^\eta \cdot g^{r'_{\pi(i)}} \cdot h^{\eta r'_{\pi(i)}},$$

or equivalently

$$\frac{\hat{c}_{\pi(i),1} \cdot g^{r'_{\pi(i)}}}{c_{i,1}} \cdot \left(\frac{\hat{c}_{\pi(i),2} h^{r'_{\pi(i)}}}{c_{i,2}} \right)^\eta = 1.$$

We saw that π and $\{r'_i\}$ do not depend on η . Thus by Schwarz-Zippel over η we have that $\frac{\hat{c}_{\pi(i),1} \cdot g^{r'_{\pi(i)}}}{c_{i,1}} = 1$ and $\frac{\hat{c}_{\pi(i),2} h^{r'_{\pi(i)}}}{c_{i,2}} = 1$, hence the claim also holds for ElGamal-shuffles.

C.3 Necessary lemmas to prove Theorem 8

It turns out that the proof of Theorem 8 is rather involved and includes many special cases. Thus we start by proving some lemmas essential to the proof. Moreover, it seems likely that the Theorem can be strengthened in a way that will be discussed later, but we will leave the details for future work.

Lemma 5. *Let D be a $n \times n$ square matrix with elements in \mathbb{Z}_q with rows $D^{(1)}, \dots, D^{(n)}$. Let $E \subset \mathbb{Z}_q$ be a set. Consider the probability*

$$p_{D,E} := \Pr[\langle D^{(i)}, \mathbf{b} \rangle \in E, \forall i \in [1, n] | b_1, b_2, \dots, b_n \stackrel{\$}{\leftarrow} E, \mathbf{b} = (b_1, \dots, b_n)].$$

Let a be the largest absolute value of an element of D . Let E be an interval $[0, m-1]$ with $ma < q$. Let m be even. Then $p_{D,E} \leq \frac{\lfloor \frac{m}{a} \rfloor}{m}$.

Proof. Let the row where an element with the largest absolute value is stored be w.l.o.g $D^{(1)}$. Let one such element, w.l.o.g, be $d_{1,n}$. Let us consider the probability that $\langle D^{(1)}, \mathbf{b} \rangle \in [0, m-1]$. This probability is equal to

$$\sum_{j=0}^{q-1} \Pr[\sum_{i=1}^{n-1} d_{1,i} b_i = j] \cdot \Pr[\sum_{i=1}^n d_{1,i} b_i \in [0, m-1] | \sum_{i=1}^{n-1} d_{1,i} b_i = j]. \quad (7)$$

Let us try to upper-bound $\Pr[\sum_{i=1}^n d_{1,i} b_i \in [0, m-1] | \sum_{i=1}^{n-1} d_{1,i} b_i = j]$. We note that if the b_1, \dots, b_{n-1} have been fixed, then the function $f(b_n) = \sum_{i=1}^n d_{1,i} b_i = j + d_{1,n} b_n$ is injective for $b_n \in [0, m-1]$ because $ma < q$. This means that the probability

$$\Pr_{b_n \in [0, m-1]} [j + d_{1,n} b_n \in [0, m-1]] = \frac{(j + [0, m-1]d_{1,n}) \cap [0, m-1]}{(j + [0, m-1]d_{1,n})}.$$

We see that in the set $(j + [0, m-1]d_{1,n})$, the distance between each two consecutive elements is $d_{1,n}$. Thus, the size of $(j + [0, m-1]d_{1,n}) \cap [0, m-1]$ can be at

most $\lceil \frac{m}{a} \rceil$. From this it follows that $\Pr_{b_n \in [0, m-1]}[j + d_{1,n} b_n \in [0, m-1]] \leq \frac{\lceil \frac{m}{a} \rceil}{m}$. Thus the probability in Equation 8 is no bigger than

$$\sum_{j=0}^{q-1} \Pr\left[\sum_{i=1}^{n-1} d_{1,i} b_i = j\right] \cdot \frac{\lceil \frac{m}{a} \rceil}{m} = \frac{\lceil \frac{m}{a} \rceil}{m}.$$

Thus the result is proven.

- Lemma 6.** 1. Let $\alpha \in [2, p-2] \pmod{p}$ with $(|\alpha| + 1) \cdot (\sum_{i=0}^f p^i) < q$. Then if $\alpha c = d$ and $c, d \in P_{p,f} - P_{p,f}$, then $c = d = 0$.
2. Let $\alpha, \beta \in \mathbb{Z}_q$ be such that $(|\alpha| + |\beta|) \cdot (\sum_{i=0}^f p^i) \leq q$, and where $\alpha \neq \beta, p - \beta \pmod{p}$, and where $\alpha, \beta \neq 0 \pmod{p}$. Then if $\alpha c = \beta d$ and $c, d \in P_{p,f} - P_{p,f}$, then $c = d = 0$.

Proof. 1. We note that $P_{p,f} - P_{p,f} = \{b \in \mathbb{Z}_q \mid \exists b_0, \dots, b_f \in \{-1, 0, 1\}, b = \sum_{i=0}^f b_i p^i\}$. Let $c = \sum_{i=0}^f c_i p^i$ and $d = \sum_{i=0}^f d_i p^i$ where all c_i and d_i are in $\{-1, 0, 1\}$. We have thus that $\alpha \sum_{i=0}^f c_i p^i = \sum_{i=0}^f d_i p^i$, i.e

$$\sum_{i=0}^f (\alpha c_i - d_i) p^i = 0.$$

Let j be the smallest index where either of c_i or d_i is nonzero. Thus the equation becomes

$$\sum_{i=j}^f (\alpha c_i - d_i) p^i = 0.$$

Let us now move this equation from \mathbb{Z}_q to \mathbb{Z} . Thus, using for each variable the representative in $[-\frac{q+1}{2}, \frac{q+1}{2})$, we would have that for some integer v we would have

$$\sum_{i=j}^f (\alpha c_i - d_i) p^i = vq$$

over the integers, where $\alpha, c_j, \dots, c_f, d_j, \dots, d_f \in [-\frac{q+1}{2}, \frac{q+1}{2})$. We note that

$$\sum_{i=j}^f (\alpha c_i - d_i) p^i \leq \sum_{i=j}^f (|\alpha| |c_i| + |d_i|) p^i \leq \sum_{i=j}^f (|\alpha| + 1) p^i \leq (|\alpha| + 1) |p^f| < q$$

and analogously that $\sum_{i=j}^f (\alpha c_i - d_i) p^i > -q$. Thus it must be that $v = 0$ and we have that $\sum_{i=j}^f (\alpha c_i - d_i) p^i = 0$ holds over \mathbb{Z} .

Over \mathbb{Z} we can consider the equation modulo p^{j+1} . We then obtain that $(\alpha c_j - d_j) p^j = 0 \pmod{p^{j+1}}$. From this follows that $\alpha c_j - d_j = 0 \pmod{p}$. Let $\alpha = p\alpha_1 + \alpha_2$ with $\alpha_2 \in [2, p-2]$. Then the equation becomes $\alpha_2 c_j - d_j = 0 \pmod{p}$. Considering the nine possible values for the pair (c_j, d_j) we see that the only case when it is possible that $\alpha_2 c_j - d_j = 0 \pmod{p}$, is when $c_j = d_j = 0$. However, this contradicts our assumption that either c_j or d_j is nonzero. Thus, by contradiction the claim is proven.

2. Analogously to the last case, we reach the equation $\alpha c_j - \beta d_j = 0 \pmod{p}$. Analyzing the nine cases, we see that this is possible only when $c_j = d_j = 0$. The result again follows.

Lemma 7. *Let $\alpha \in [2, p-2] \pmod{p}$ with $(|\alpha| + 1) \cdot (\sum_{i=0}^f p^i) < q$. Then, for any $k \in \mathbb{Z}_q$, $|(k + \alpha P_{p,f}) \cap P_{p,f}| \leq 1$.*

Proof. Suppose by contradiction that there exist at least two $x_1, x_2 \in (k + \alpha P_{p,f}) \cap P_{p,f}$. Thus there must exist $a_1, a_2, b_1, b_2 \in P_{p,f}$, such that $x_1 = k + \alpha a_1 = b_1$ and $x_2 = k + \alpha a_2 = b_2$. Thus it must be that $x_1 - x_2 = \alpha(a_1 - a_2) = b_1 - b_2$. Denote $a_1 - a_2 =: c, b_1 - b_2 =: d$. We have that $c, d \in P_{p,f} - P_{p,f}$ and $\alpha c = d$. By the first claim of the previous lemma it must hold that $c = d = 0$ and thus $a_1 = a_2$ and $b_1 = b_2$, hence $x_1 = x_2$. Hence the claim is proven.

Lemma 8. *Let D be a $n \times n$ square matrix with elements in \mathbb{Z}_q with rows $D^{(1)}, \dots, D^{(n)}$. Let $E = P_{p,f}$ be a set. Consider the probability*

$$p_{D,E} := \Pr[\langle D^{(i)}, \mathbf{b} \rangle \in E, \forall i \in [1, n] | b_1, b_2, \dots, b_n \stackrel{\$}{\leftarrow} E, \mathbf{b} = (b_1, \dots, b_n)].$$

Suppose that in D there is an element with value α such that $(|\alpha| + 1) \cdot (\sum_{i=0}^f p^i) < q$ and $\alpha \in [2, p-2] \pmod{p}$. Then $p_{D,E} \leq \frac{1}{|E|}$.

Proof. Without loss of generality, let the element α be the last element in the first row, i.e. $d_{1,n}$.

Let us consider the probability that $\langle D^{(i)}, \mathbf{b} \rangle \in P_{p,f}$. This probability is equal to

$$\sum_{j=0}^{q-1} \Pr[\sum_{i=1}^{n-1} d_{1,i} b_i = j] \cdot \Pr[\sum_{i=1}^n d_{1,i} b_i \in P_{p,f} | \sum_{i=1}^{n-1} d_{1,i} b_i = j]. \quad (8)$$

Let us try to bound $\Pr[\sum_{i=1}^n d_{1,i} b_i \in P_{p,f} | \sum_{i=1}^{n-1} d_{1,i} b_i = j]$.

We note that if the b_1, \dots, b_{n-1} have been fixed, then we can consider the function $f(b_n) = \sum_{i=1}^n d_{1,i} b_i = k + d_{1,n} b_n$ where $k = \sum_{i=1}^{n-1} d_{1,i} b_i$. We first note that this function must be injective because if for some different b_n, b'_n it held that $f(b_n) = f(b'_n)$ then $0 = f(b_n) - f(b'_n) = k + d_{1,n} b_n - k - d_{1,n} b'_n = \alpha(b_n - b'_n)$. Because $\alpha, b_n - b'_n \neq 0$, and $|\alpha| |b_n - b'_n| < q$, this is not possible.

This means that

$$\Pr_{b_n \in P_{p,f}} [j + d_{1,n} b_n \in P_{p,f}] = \frac{(j + \alpha P_{p,f}) \cap P_{p,f}}{|P_{p,f}|} \leq \frac{1}{|P_{p,f}|}.$$

The last inequality comes from Lemma 7.

Hence the claim is proven.

Lemma 9. *Let $a_1, a_2 \in [-m, m]$ and let $x_1, x_2 \in P_{p,f}$. Let $a_1, a_2 \neq 0$. Let $2m + 1 \leq p$. Let $p \cdot (\sum_{i=0}^f p^i) \leq q$. Let $a_1 x_1 = a_2 x_2$. Let $x_1, x_2 \neq 0$. Then $a_1 = \pm a_2$.*

Proof. We have that by the second clause of Lemma 6, if $(|a_1| + |a_2|) \cdot (\sum_{i=0}^f p^i) < q$ and $a_1 \not\equiv a_2, p - a_2 \pmod{p}$, and $a_1, a_2 \not\equiv 0 \pmod{p}$, then it follows that $x_1 = x_2 = 0$. Thus it must be that some of the conditions are not satisfied. We first note that

$$(|a_1| + |a_2|) \cdot \left(\sum_{i=0}^f p^i \right) \leq 2m \left(\sum_{i=0}^f p^i \right) < p \cdot \left(\sum_{i=0}^f p^i \right) \leq q,$$

thus this condition is satisfied. Thus we must have that $a_1 \equiv \pm a_2 \pmod{p}$.

We first note that if $a_1 \equiv a_2 \pmod{p}$ or $a_1 \equiv p - a_2 \pmod{p}$, then, because $p > 2m + 1$, we have that these cases mean that $a_1 = a_2$ or $a_1 = -a_2$, i.e. $a_1 \equiv \pm a_2$.

Lemma 10. [19]

$$\begin{aligned} - \sum_{i=0}^{\lfloor \frac{n}{4} \rfloor} \binom{n}{4i} &= \frac{1}{2} (2^{n-1} + 2^{\frac{n}{2}} \cos \frac{n\pi}{4}) \\ - \sum_{i=0}^{\lfloor \frac{n-1}{4} \rfloor} \binom{n}{4i+1} &= \frac{1}{2} (2^{n-1} + 2^{\frac{n}{2}} \sin \frac{n\pi}{4}) \\ - \sum_{i=0}^{\lfloor \frac{n-2}{4} \rfloor} \binom{n}{4i+2} &= \frac{1}{2} (2^{n-1} - 2^{\frac{n}{2}} \cos \frac{n\pi}{4}) \\ - \sum_{i=0}^{\lfloor \frac{n-3}{4} \rfloor} \binom{n}{4i+3} &= \frac{1}{2} (2^{n-1} - 2^{\frac{n}{2}} \sin \frac{n\pi}{4}) \end{aligned}$$

Lemma 11. Let $p, k < n$ with $p \geq 4$ and $k \in [0, p-1]$. Denote sum $v_{n,p,k} := \binom{n}{k} + \binom{n}{k+p} + \binom{n}{k+2p} + \dots + \binom{n}{k + \lfloor \frac{n-k}{p} \rfloor p}$. Then $v_{n,p,k} \leq \frac{1}{2} (2^{n-1} + 2^{\frac{n}{2}})$.

Proof. We note that by Lemma 10, we have that $v_{n,4,k} \leq \frac{1}{2} (2^{n-1} + 2^{\frac{n}{2}})$ for any n and k where $k < p < n$.

We will now show that for any n and k and $p > 4$, there exists a $k' \in [0, p-1]$ such that $v_{n,4,k'} \geq v_{n,p,k}$.

Let $p > 4$. Let the largest among the $\binom{n}{k+ip}$ be $\binom{n}{k+jp}$. Choose k' to be smallest such nonnegative integer that there exists a nonnegative integer j' such that $k + jp = k' + j'4$.

Now $v_{n,p,k} = \sum_{i=-J}^K \binom{n}{k+(j+i)p}$ for some nonnegative J and K where $k + (j - J)p \geq 0$ and $k + (j + K)p \leq n$. Because $k + jp = k' + j'4$ and $p > 4$ and $J, K \leq 0$, we have that

$$k + jp - pJ \leq k' + j'4 - 4J$$

and

$$k + jp + pK \geq k' + j'4 + 4K$$

We can rewrite these as $k' + (j' - J)4 > k + (j - J)p$ and $k' + (j' + K)4 < k + (j + K)p$.

We note that $\{k' + (j' + i)4\}_{i=-J}^K$ is a subset of $\{k' + 4i\}_{i=0}^{\lfloor \frac{n-k'}{4} \rfloor}$ because $k' + (j' - J)4 \geq k + jp - pJ \geq 0$ and $k' + (j' + K)4 \leq k + jp + pK \leq n$ and $\{k' + 4i\}_{i=0}^{\lfloor \frac{n-k'}{4} \rfloor}$ is the set of all integers $a \in [0, n]$ such that $a \equiv k' \pmod{4}$.

Thus

$$v_{n,4,k'} = \sum_{i=0}^{\lfloor \frac{n-k'}{4} \rfloor} \binom{n}{k'+4i} \geq \sum_{i=-J}^K \binom{n}{k'+(j'+i)4}.$$

Now, for any $i \in [-J, K]$, we will show that $\binom{n}{k'+(j'+i)4} \geq \binom{n}{k+(j+i)p}$.

For this, note that if $k+jp \geq \frac{n}{2}$, and if $i > 0$, then $k+(j+i)p = k+jp+ip = k'+j'4+ip > k'+j'4+4p = k'+(j'+i)4$. Because the binomial function $f(x) = \binom{n}{x}$ is decreasing when $x \geq \frac{n}{2}$, we have that in that case $\binom{n}{k'+(j'+i)4} \geq \binom{n}{k+(j+i)p}$.

Analogously, if $k+jp \leq \frac{n}{2}$, and if $i < 0$, we likewise get that $k+(j+i)p < k'+(j'+i)4$ and because $f(x) = \binom{n}{x}$ is increasing when $x \leq \frac{n}{2}$, we have that in that case $\binom{n}{k'+(j'+i)4} \geq \binom{n}{k+(j+i)p}$.

Now, if $k+jp < \frac{n}{2}$, and if $i > 0$, then we must have that $k+(j+i)p > \frac{n}{2}$. If we had that $k+(j+i)p \leq \frac{n}{2}$, then, because the binomial function is increasing for $x \leq \frac{n}{2}$, we would have that $\binom{n}{k+(j+i)p} > \binom{n}{k+jp}$ which would contradict the maximality of $\binom{n}{k+jp}$.

Now, for $k'+(j'+i)4$, there are two options. Either $k'+(j'+i)4 \leq \frac{n}{2}$ or $k'+(j'+i)4 > \frac{n}{2}$. If $k'+(j'+i)4 \leq \frac{n}{2}$, then $\binom{n}{k'+(j'+i)4} \geq \binom{n}{k+jp} \geq \binom{n}{k+(j+i)p}$. The first of these equalities comes from the increasingness of the binomial function for all values not greater than $\frac{n}{2}$ and the second inequality comes from the maximality of $\binom{n}{k+jp}$.

If $k'+(j'+i)4 > \frac{n}{2}$, then, because $k+(j+i)p < k'+(j'+i)4$, both of the values are greater than $\frac{n}{2}$ and the binomial function is decreasing for values greater than $\frac{n}{2}$, we have that $\binom{n}{k'+(j'+i)4} \geq \binom{n}{k+(j+i)p}$.

The proof for the case when $k+jp > \frac{n}{2}$, and if $i < 0$ is analogous. Thus for all j , $\binom{n}{k'+(j'+i)4} \geq \binom{n}{k+(j+i)p}$.

Thus we get that

$$v_{n,p,k} = \sum_{i=-J}^K \binom{n}{k+(j+i)p} \leq \sum_{i=-J}^K \binom{n}{k'+(j'+i)4} \leq v_{n,4,k'} \leq \frac{1}{2}(2^{n-1} + 2^{\frac{n}{2}}).$$

Hence the result follows.

Lemma 12. *Let a vector \mathbf{d} of length N with elements from \mathbb{Z}_q contain $k+1$ elements 1, k elements -1 and let the rest of the elements be 0. Consider the probability*

$$J_{k,t} := \Pr[\langle \mathbf{d}, \mathbf{b} \rangle = t | b_1, b_2, \dots, b_N \stackrel{\$}{\leftarrow} \{0, 1\}, \mathbf{b} = (b_1, \dots, b_n)].$$

Then for any $t \in [-k, k]$, we have that $J_{k,t} = \frac{\binom{2k+1}{t+k}}{2^{2k+1}}$.

Proof. We do this inductively over k . It is clear that the elements of \mathbf{d} that are zero can be ignored. W.l.o.g, let $\mathbf{d} = (1, -1, 1, -1, \dots, 1, 0, \dots, 0)$. Let us thus only consider the vector \mathbf{d}_k formed out of the first $2k+1$ elements of \mathbf{d} . Let $I_{k,t}$ be the number of binary vectors \mathbf{b} of length $2k+1$ such that $\langle \mathbf{b}, \mathbf{d}_k \rangle = t$. By definition, clearly $J_{k,t} = \frac{I_{k,t}}{2^{2k+1}}$.

Now let us do the proof by induction. For the base case, if $k = 0$, then $\mathbf{d}_0 = (1)$. There is one vector $\mathbf{b} = (0)$ for which $\langle \mathbf{b}, \mathbf{d}_k \rangle = 0$ and one vector $\mathbf{b} = (1)$ for which $\langle \mathbf{b}, \mathbf{d}_k \rangle = 1$. Thus $I_{0,0} = I_{0,1} = 1$ and $J_{0,0} = J_{0,1} = \frac{1}{2}$. Now let us do the step of the induction. Suppose that the statement holds for k , let us show that it holds for $k + 1$. We notice that moving from k to $k + 1$ is equivalent to concatenating $(-1, 1)$ to the vector \mathbf{d}_k . We thus will characterize the values $I_{k+1,t}$ in terms of $I_{k,t}$. Consider now how we can obtain $I_{k+1,t}$ when the first $2k - 1$ values of \mathbf{b} have been fixed. Let us denote those values with \mathbf{b}_k . Let the last two elements of \mathbf{b} be b_{2k} and b_{2k+1} . There are four cases how it can happen that the sum is t : either the scalar product of $\langle \mathbf{b}_k, \mathbf{d}_k \rangle = t - 1$ and $b_{2k} = 0$ and $b_{2k+1} = 1$; the scalar product of $\langle \mathbf{b}_k, \mathbf{d}_k \rangle = t$ and $b_{2k} = 0$ and $b_{2k+1} = 0$; the scalar product of $\langle \mathbf{b}_k, \mathbf{d}_k \rangle = t$ and $b_{2k} = 1$ and $b_{2k+1} = 1$; or the scalar product of $\langle \mathbf{b}_k, \mathbf{d}_k \rangle = t + 1$ and $b_{2k} = 1$ and $b_{2k+1} = 0$. Thus we get that

$$I_{k+1,t} = I_{k+1,t-1} + 2I_{k+1,t} + I_{k+1,t+1}.$$

By the premise of the inductive step, we have that

$$I_{k+1,t} = \binom{2k+1}{t+k-1} + \binom{2k+1}{t+k} + \binom{2k+1}{t+k} + \binom{2k+1}{t+k+1} = \binom{2k+2}{t+k} + \binom{2k+2}{t+k+1} = \binom{2k+3}{t+k+1}.$$

Thus we have by induction that $I_{k,t} = \binom{2k+1}{t+k}$. From here it follows that $J_{k,t} = \frac{I_{k,t}}{2^{2k+1}} = \frac{\binom{2k+1}{t+k}}{2^{2k+1}}$.

Lemma 13. *Let a vector \mathbf{d} of length N with elements from \mathbb{Z}_q contain $k + 1$ elements 1, k elements -1 and let the rest of the elements be 0. Consider the probability*

$$p_{k,\{0,1\}} := \Pr[\langle \mathbf{d}, \mathbf{b} \rangle \in \{0, 1\} | b_1, b_2, \dots, b_N \stackrel{\$}{\leftarrow} \{0, 1\}, \mathbf{b} = (b_1, \dots, b_n)].$$

Then, for any integer $k \geq 1$, $p_{k,\{0,1\}} \leq \frac{3}{4}$.

Proof. First notice that by the previous lemma 12, we have that $p_{k,\{0,1\}} = \frac{\binom{2k+1}{k} + \binom{2k+1}{k+1}}{2^{2k+1}} = \frac{\binom{2k+2}{k+1}}{2^{2k+1}}$. For $k = 1$, we have that $p_{k,\{0,1\}} = \frac{\binom{4}{2}}{2^3} = \frac{6}{8} = \frac{3}{4}$. Now we will show that for any $k \geq 0$, we have that $p_{k,\{0,1\}} > p_{k+1,\{0,1\}}$. For that, first note that $\frac{\binom{2k+2}{k+1}}{\binom{2k+4}{k+2}} = \frac{(k+2)(k+2)}{(2k+3)(2k+4)}$ by the definition of the binomial coefficient.

Now,

$$\frac{p_{k,\{0,1\}}}{p_{k+1,\{0,1\}}} = \frac{\frac{\binom{2k+2}{k+1}}{2^{2k+1}}}{\frac{\binom{2k+4}{k+2}}{2^{2k+3}}} = \frac{4(k+2)(k+2)}{(2k+3)(2k+4)} = \frac{4k^2 + 16k + 16}{4k^2 + 10k + 12} > 1.$$

Thus $p_{k,\{0,1\}} > p_{k+1,\{0,1\}}$ and thus for all $k \geq 1$, we have $p_{k,\{0,1\}} \leq \frac{3}{4}$.

Lemma 14. Let \mathbf{d} be a vector of length n over \mathbb{Z}_q . Let all $d_i \in \mathbf{d}$ be in $\{-1, 0, 1\}$. Let the number of ones in \mathbf{d} be $k+1$ and the number of minus ones in \mathbf{d} be k . Let $(2k+1) \sum_{i=0}^f p^i < \frac{q}{2}$. Define

$$p_{\mathbf{d},E} := \Pr[\langle \mathbf{d}, \mathbf{b} \rangle \in P_{p,f} | b_1, b_2, \dots, b_n \stackrel{\$}{\leftarrow} P_{p,f}, \mathbf{b} = (b_1, \dots, b_n)].$$

Then

$$p_{\mathbf{d},E} \leq \left(\frac{1}{2} + \frac{1}{2^{k+1}}\right)^{f+1}$$

Proof. Fix the \mathbf{d} and denote $\langle \mathbf{d}, \mathbf{b} \rangle$ by g_b .

Because

$$|\langle \mathbf{d}, \mathbf{b} \rangle| \leq \sum_{i=0}^{2k+1} b_i \leq (2k+1) \sum_{i=0}^f p^i < \frac{q}{2}$$

we can consider g_b as an integer in $[\frac{-q}{2}, \frac{q}{2})$.

We can write $g_b = \sum_{i=0}^F g_i p^i$ in p -ary where $g_0, \dots, g_F \in [\lfloor \frac{-p}{2} \rfloor, \lfloor \frac{p}{2} \rfloor]$ and $F \geq f$. Note that this representation is unique, as we can first consider g_b modulo p to obtain g_0 , then consider $g_b - g_0$ modulo p^2 to obtain pg_1 and so on.

We notice that $g_b \in P_{p,f}$ iff $g_0, \dots, g_f \in \{0, 1\}$ and $g_{f+1}, \dots, g_F = 0$. Unfortunately we cannot consider these cases independently as there might be an overflow term from some lower terms might influence higher terms. However, we shall see that this will not make the result particularly worse.

Namely, if $b_j = \sum_{i=0}^f b_{i,j} p^i$ where $b_{i,j} \in \{0, 1\}$, then we obtain that

$$\sum_{i=0}^F g_i p^i = \langle \mathbf{d}, \mathbf{b} \rangle = \sum_{j=1}^n d_j b_j = \sum_{j=1}^n d_j \sum_{i=0}^f b_{i,j} p^i = \sum_{i=0}^f p^i \sum_{j=1}^n d_j b_{i,j}.$$

This does not necessarily mean that $g_b \in P_{p,f}$ iff $\sum_{j=1}^n d_j b_{i,j} \in \{0, 1\}$ for all $i \in [0, f]$. If the number of d_j that are nonzero is greater than p , then it might happen that for some i , $\sum_{j=1}^n d_j b_{i,j} \geq p$ and thus it will both give an overflow term to the coefficients of p^{i+1} and that for the coefficient of p^i it suffices when the sum of the $d_j b_{i,j}$ (plus a possible overflow or underflow term from the lower values) is equal to 0 or 1 (mod p).

Thus, consider the following. For some fixed values of the vector \mathbf{b} , define iteratively:

$$\sum_{j=1}^N b_{0,j} d_j =: pc_{b,0} + c'_{b,0}$$

where $c'_{b,0} \in [\lfloor \frac{-p}{2} \rfloor, \lfloor \frac{p}{2} \rfloor]$, and for $i \in [1, F]$:

$$c_{b,i-1} + \sum_{j=1}^N b_{i,j} d_j =: pc_{b,i} + c'_{b,i}$$

where $c'_{b,i} \in [\lfloor \frac{-p}{2} \rfloor, \lfloor \frac{p}{2} \rfloor]$.

Also define $c_{b,-1} = 0$.

We note that thus defined $c_{b,i}$ and $c'_{b,i}$ are a function of b , as \mathbf{d} has been assumed to be fixed.

Thus we can say that $\langle \mathbf{d}, \mathbf{b} \rangle \in P_{p,f}$ if and only $c'_{b,i} \in \{0, 1\}$ for all $i \in [0, f]$ and $c'_{b,i} = 0$ for all $i > f$. However, we will now focus only on the first of these requirements — $c'_{b,i} \in \{0, 1\}$ for all $i \in [0, f]$.

We note that this is equivalent to $c_{b,i-1} + \sum_{j=1}^n b_{i,j}d_j \in \{0, 1\} \pmod{p}$ for all $i \in [0, f]$.

Note that, assuming that the elements $b_{i,j}$ are chosen uniformly at random from $\{0, 1\}$, the probability $\Pr[c'_{b,i} \in \{0, 1\} \pmod{p} | c_{b,i-1} = w]$ is well-defined for every $w \in \mathbb{Z}_q$ where $\Pr[c_{b,i-1} = w] > 0$.

For the probabilities given from here on, assume that the probability is given for picking all the elements $b_{i,j}$ uniformly at random from $\{0, 1\}$.

We can write for all $w \in \mathbb{Z}_q$ that

$$\Pr[c'_{b,i} \in \{0, 1\} \pmod{p} | c_{b,i-1} = w] = \Pr[w + \sum_{j=1}^n b_{i,j}d_j \in \{0, 1\} \pmod{p}]$$

Let v_i be defined as such that the value $\Pr[v_i + \sum_{j=1}^n b_{i,j}d_j \in \{0, 1\} \pmod{p}]$ is maximal for $v_i \in [0, p-1]$.

Now by that definition

$$\Pr[c_{b,i-1} + \sum_{j=1}^n b_{i,j}d_j \in \{0, 1\} \pmod{p}] \leq \Pr[v_i + \sum_{j=1}^n b_{i,j}d_j \in \{0, 1\} \pmod{p}].$$

We note that by Lemma 12, the probability that $\sum_{j=1}^n b_{i,j}d_j = t$ if $k+1$ of the d_j are equal to 1, k are equal to -1 and the $b_{i,j}$ are random bits, is equal to $\frac{\binom{2k+1}{k+t}}{2^{2k+1}}$. From here we have directly that the probability that $v_i + \sum_{j=1}^n b_{i,j}d_j = t$ if $k+1$ of the d_j are equal to 1, k are equal to -1 and the $b_{i,j}$ are random bits, is equal to $\frac{\binom{2k+1}{k+t-v_i}}{2^{2k+1}}$. We note that the only values for t for which this value is nonzero are $[v_i - k, v_i + k + 1]$, thus we only have to look at them.

The respective values t that we would be interested in are the values where $t \in \{0, 1\} \pmod{p}$. In $[v_i - k - 1, 2k + 1]$ let the smallest value that is equal to $0 \pmod{p}$ be v'_i and let the largest value in $[v_i - k - 1, 2k + 1]$ that is equal to $0 \pmod{p}$ be v''_i . Thus we obtain that

$$\Pr[v + \sum_{j=1}^n b_{i,j}d_j \in \{0, 1\} \pmod{p}] = \frac{\sum_{j=0}^{\frac{v''_i - v'_i}{p}} \binom{2k+1}{k-v_i+v'_i+jp} + \binom{2k+1}{k-v_i+v'_i+jp+1}}{2^{2k+1}}.$$

(For edge cases when one of the $k - v_i + v'_i + jp$ and $k - v_i + v'_i + jp + 1$ is in $[0, 2k + 1]$ and the other is not, recall that $\binom{2k+1}{-1} = \binom{2k+1}{2k+2} = 0$ and thus it does not matter whether we include them or not, we choose to include them to make the equation nicer.)

Additionally, $\binom{2k+1}{k-v_i+v'_i+jp} + \binom{2k+1}{k-v_i+v'_i+jp+1} = \binom{2k+2}{k-v_i+v'_i+jp+1}$.
Hence

$$\Pr[v_i + \sum_{j=1}^n b_{i,j}d_j \in \{0, 1\} \pmod{p}] \leq \frac{\sum_{j=0}^{\frac{v'_i-v_i}{p}} \binom{2k+2}{k-v_i+v'_i+jp+1}}{2^{2k+1}}.$$

We can obtain an upper-bound for the right-hand side from Lemma 11 and obtain that it is upper-bounded by $\frac{\frac{1}{2}(2^{2k+1}+2^{k+1})}{2^{2k+1}} = \frac{1}{2} + \frac{1}{2^{k+1}}$.

Knowing this, we can bound the entire probability. Thus

$$\begin{aligned} & \Pr\left[\bigwedge_{i=0}^f c'_{b,i} \in \{0, 1\} \pmod{p}\right] = \\ & \sum_{\substack{w_{-1}, w_0, \dots, w_{f-1} \in \mathbb{Z}_q \\ \Pr\left[\bigwedge_{i=-1}^{f-1} c_{b,i} = w_i\right] > 0}}^n \Pr\left[\bigwedge_{i=0}^f c'_{b,i} \in \{0, 1\} \pmod{p} \mid \bigwedge_{i=-1}^{f-1} c_{b,i} = w_i\right] \Pr\left[\bigwedge_{i=-1}^{f-1} c_{b,i} = w_i\right] = \\ & \sum_{\substack{w_{-1}, w_0, \dots, w_{f-1} \in \mathbb{Z}_q \\ \Pr\left[\bigwedge_{i=-1}^{f-1} c_{b,i} = w_i\right] > 0}}^n \Pr\left[\bigwedge_{i=0}^f w_{i-1} + \sum_{j=1}^n b_{i,j}d_j \in \{0, 1\} \pmod{p}\right] \Pr\left[\bigwedge_{i=-1}^{f-1} c_{b,i} = w_i\right] \leq \\ & \sum_{\substack{w_{-1}, w_0, \dots, w_{f-1} \in \mathbb{Z}_q \\ \Pr\left[\bigwedge_{i=-1}^{f-1} c_{b,i} = w_i\right] > 0}}^n \Pr\left[\bigwedge_{i=0}^f v_i + \sum_{j=1}^n b_{i,j}d_j \in \{0, 1\} \pmod{p}\right] \Pr\left[\bigwedge_{i=-1}^{f-1} c_{b,i} = w_i\right] \leq \\ & \sum_{\substack{w_{-1}, w_0, \dots, w_{f-1} \in \mathbb{Z}_q \\ \Pr\left[\bigwedge_{i=-1}^{f-1} c_{b,i} = w_i\right] > 0}}^n \left(\prod_{i=0}^f \Pr[v_i + \sum_{j=1}^n b_{i,j}d_j \in \{0, 1\} \pmod{p}]\right) \Pr\left[\bigwedge_{i=-1}^{f-1} c_{b,i} = w_i\right] \leq \\ & \sum_{\substack{w_{-1}, w_0, \dots, w_{f-1} \in \mathbb{Z}_q \\ \Pr\left[\bigwedge_{i=-1}^{f-1} c_{b,i} = w_i\right] > 0}}^n \left(\prod_{i=0}^f \left(\frac{1}{2} + \frac{1}{2^{k+1}}\right)\right) \Pr\left[\bigwedge_{i=-1}^{f-1} c_{b,i} = w_i\right] = \\ & \left(\frac{1}{2} + \frac{1}{2^{k+1}}\right)^{f+1} \sum_{\substack{w_{-1}, w_0, \dots, w_{f-1} \in \mathbb{Z}_q \\ \Pr\left[\bigwedge_{i=-1}^{f-1} c_{b,i} = w_i\right] > 0}}^n \Pr\left[\bigwedge_{i=-1}^{f-1} c_{b,i} = w_i\right] = \left(\frac{1}{2} + \frac{1}{2^{k+1}}\right)^{f+1}. \end{aligned}$$

Hence we have the desired result.

Lemma 15. *Let D be a $n \times n$ matrix over \mathbb{Z}_q where the only elements are $-1, 0$ and 1 and all rows and columns sum to 1 . Let $p > 4$. Let $2p^{f+1} \leq q$. Additionally, let there be no row with more than 2 elements 1 . Denote*

$$p_{D,E} := \Pr[\langle D^{(i)}, \mathbf{b} \rangle \in P_{p,f}, \forall i \in [1, n] | b_1, b_2, \dots, b_n \stackrel{\$}{\leftarrow} P_{p,f}, \mathbf{b} = (b_1, \dots, b_n)]$$

and

$$p_{D,\{0,1\}} := \Pr[\langle D^{(i)}, \mathbf{b} \rangle \in [0, 1], \forall i \in [1, n] | b_1, b_2, \dots, b_n \stackrel{\$}{\leftarrow} [0, 1], \mathbf{b} = (b_1, \dots, b_n)].$$

Then

$$p_{D,E} = p_{D,\{0,1\}}^{f+1}.$$

Proof. In the first experiment where we randomly choose $b_1, b_2, \dots, b_n \stackrel{\$}{\leftarrow} P_{p,f}$, denote $b_j = \sum_{k=0}^f b_{j,k} p^k$ where all $b_{j,k}$ are bits. Thus the first experiment is the same when we choose the bits separately, that is

$$\begin{aligned} & \Pr[\langle D^{(i)}, \mathbf{b} \rangle \in P_{p,f}, \forall i \in [1, n] | b_1, b_2, \dots, b_n \stackrel{\$}{\leftarrow} P_{p,f}, \mathbf{b} = (b_1, \dots, b_n)] = \\ & \Pr[\langle D^{(i)}, \mathbf{b} \rangle \in P_{p,f}, \forall i \in [1, n] | \{b_{j,k}\}_{i,j=1,k=0}^{N,f} \stackrel{\$}{\leftarrow} \{0, 1\}, b_j \leftarrow \sum_{k=0}^f b_{j,k} p^k, \mathbf{b} = (b_1, \dots, b_n)] \end{aligned}$$

Let the matrix D be $\{d_{i,j}\}_{i,j=1}^N$. We note that $\langle D^{(i)}, \mathbf{b} \rangle \in P_{p,f}, \forall i \in [1, n]$ means that for all i , $\sum_{j=1}^N d_{i,j} b_j = \sum_{k=0}^f c_{k,i} p^k$ and all $c_{k,i}$ are bits.

We have that

$$\sum_{j=1}^N d_{i,j} b_j = \sum_{j=1}^N d_{i,j} \sum_{k=0}^f b_{j,k} p^k = \sum_{k=0}^f p^k \sum_{j=1}^N d_{i,j} b_{j,k}.$$

Because we have that any row is either a unit vector or the nonzero elements are two elements 1 and one element -1 , we have that for any i, k , $-1 \leq \sum_{j=1}^N d_{i,j} b_{j,k} \leq 2$. This means that there can be no "overflow" to the next power of p , i.e

$$\sum_{j=1}^N d_{i,j} b_j = \sum_{k=0}^f c_{k,i} p^k, \forall k, i : c_{k,i} \in \{0, 1\}$$

is equivalent to

$$\sum_{j=1}^N d_{i,j} b_{j,k} \in \{0, 1\}, \forall k, i.$$

We note, however, that if we fix a value k , then the probability that for all i , we have that $\sum_{j=1}^N d_{i,j} b_{j,k} \in \{0, 1\}$, can be written as

$$\Pr[\langle D^{(i)}, \mathbf{b} \rangle \in [0, 1], \forall i \in [1, n] | b_1, b_2, \dots, b_n \stackrel{\$}{\leftarrow} [0, 1], \mathbf{b} = (b_1, \dots, b_n)] = p_{D,\{0,1\}}.$$

Now, because there are $f + 1$ values of k and for any particular value of k the outcome does not depend on any other value of k , these experiments are independent. Thus the result follows. \square

Theorem 8. Let D be a $n \times n$ square matrix with elements in \mathbb{Z}_q with columns $D^{(1)}, \dots, D^{(n)}$ where every row and column of D sums to 1. Let $p > 4$. Let $2p^{f+1} \leq q$. Let $n \sum_{i=0}^f p^i < \frac{q}{2}$. Let all values in D have absolute values no greater than $p - 2$. Let D contain a column that is not a unit vector. Then

$$p_{D,E} := \Pr[(D^{(i)}, \mathbf{b}) \in P_{p,f}, \forall i \in [1, n] | b_1, b_2, \dots, b_n \stackrel{\$}{\leftarrow} P_{p,f}, \mathbf{b} = (b_1, \dots, b_n)] \leq \left(\frac{5}{8}\right)^{f+1}.$$

Proof. We will split the proof down to a number of cases.

1. The matrix contains an element with an absolute value larger than 1.
2. All the elements of the matrix are either $-1, 0$ or 1 . There is a row with at least 3 elements 1 .
3. All the elements of the matrix are either $-1, 0$ or 1 . In every row there is at most 2 elements 1 .

1. Let us first consider the case when there is an element (w.l.o.g $d_{1,N}$) with an absolute value larger than 1 but no larger than $p - 2$. By Lemma 8, we have that

$$p_{D,E} \leq \frac{1}{|P_{p,f}|} = \frac{1}{2^{f+1}} < \left(\frac{5}{8}\right)^{f+1}.$$

2. By Lemma 14, we have that $k \geq 3$. (The requirements for the lemma are satisfied because $n \geq 2k + 1$ and thus $(2k + 1) \sum_{i=0}^f p^i < \frac{q}{2}$.) In that row we get the bound $\left(\frac{1}{2} + \frac{1}{2^{k+1}}\right)^f \leq \left(\frac{5}{8}\right)^{f+1}$.
3. There are not very many distinct cases here. First, consider the case where there are two rows with 3 nonzero elements where the sets of the indices of the nonzero elements are disjoint. By a simple argument, we can apply the Lemma 14 to both of these rows independently and thus obtain a bound of $\left(\frac{3}{4}\right)^f \left(\frac{3}{4}\right)^f = \left(\frac{9}{16}\right)^f < \left(\frac{5}{8}\right)^f$ which suffices for our claim. If there are no two rows with 3 nonzero elements where the sets of the indices of the nonzero elements are disjoint, there are, modulo permutations of rows, a small amount of possible matrices that satisfy this property. We will describe these matrices and show that the rate is at most $\frac{5}{8}$ for the test set $\{0, 1\}$. Using the lemma 15, this will give us the result.

We will count the cases by considering what is the largest overlap of indices of nonzero coefficients of rows of three nonzero elements, classifying them, and showing that we will always have that $p_{D,E} \leq \frac{5}{8}$. First, consider the case when the largest overlap of indices of nonzero coefficients of rows of three nonzero elements is one. Then consider two rows that overlap. The submatrix where the nonzero elements of those two rows will be must be (modulo permutations) one of the following:

$$\begin{aligned} & - \begin{pmatrix} 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & -1 \end{pmatrix} \\ & - \begin{pmatrix} 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 \end{pmatrix} \\ & - \begin{pmatrix} 1 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -1 \end{pmatrix} \end{aligned}$$

$$- \begin{pmatrix} 1 & 1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 \end{pmatrix}$$

For all those cases, we can consider all the 32 possible cases for challenge bits and see that for each matrix, at least for 12 cases, the scalar product with at least one of the rows will not be a bit. Thus we have that for a matrix that contains one of these matrices as a submatrix, we must have that $p_{D,\{0,1\}} \leq \frac{5}{8}$.

Second, consider the case when the largest overlap of indices of nonzero coefficients of rows of three nonzero elements is two. Then consider two rows that overlap. The submatrix where the nonzero elements of those two rows will be must be (modulo permutations) one of the following:

$$- \begin{pmatrix} 1 & 1 & -1 & 0 \\ 1 & 1 & 0 & -1 \end{pmatrix}$$

$$- \begin{pmatrix} 1 & 1 & -1 & 0 \\ 1 & 0 & 1 & -1 \end{pmatrix}$$

$$- \begin{pmatrix} 1 & 1 & -1 & 0 \\ 1 & 0 & -1 & 1 \end{pmatrix}$$

$$- \begin{pmatrix} 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix}$$

$$- \begin{pmatrix} 1 & 1 & -1 & 0 \\ -1 & 1 & 0 & 1 \end{pmatrix}$$

$$- \begin{pmatrix} 1 & 1 & -1 & 0 \\ -1 & 0 & 1 & 1 \end{pmatrix}$$

For all those cases, we can consider all the 16 possible cases for challenge bits and see that for each matrix, at least for 6 cases, the scalar product with at least one of the rows will not be a bit. Thus we have that for a matrix that contains one of these matrices as a submatrix, we must have that $p_{D,\{0,1\}} \leq \frac{5}{8}$.

Third, consider the case when the largest overlap of indices of nonzero coefficients of rows of three nonzero elements is three. Then consider two rows that overlap. The submatrix where the nonzero elements of those two rows will be must be (modulo permutations) one of the following.

$$- \begin{pmatrix} 1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix}$$

$$- \begin{pmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \end{pmatrix}$$

For the first of those two cases, we can see that $p_{D,\{0,1\}}$ must be no greater than $\frac{1}{2}$. However, for the second case we get the bound $\frac{3}{4}$. This, however, turns out not to be an issue. Because we have that the sum of the columns must also be 1, we cannot have that all the rows with three elements are the same. Somewhere there must be another row with three elements that is not $(1 \ 1 \ -1)$ and thus must be an example of the first case, which means that for the whole matrix, we get the bound $\frac{1}{2}$.

Thus we have enumerated all the possible cases and seen that for all of them, we have that $p_{D,\{0,1\}} \leq \frac{5}{8}$, and thus, by Lemma15, this will give us the result.