Efficient Culpably Sound NIZK Shuffle Argument without Random Oracles

Full Version, November 25, 2015

Prastudy Fauzi and Helger Lipmaa

University of Tartu, Estonia

Abstract. One way to guarantee security against malicious voting servers is to use NIZK shuffle arguments. Up to now, only two NIZK shuffle arguments in the CRS model have been proposed. Both arguments are relatively inefficient compared to known random oracle based arguments. We propose a new, more efficient, shuffle argument in the CRS model. Importantly, its online prover's computational complexity is dominated by only two $(n + 1)$-wide multi-exponentiations, where n is the number of ciphertexts. Compared to the previously fastest argument by Lipmaa and Zhang, it satisfies a stronger notion of soundness.

Keywords: Bilinear pairings, CRS model, mix-net, non-interactive zero knowledge, shuffle argument.

1 Introduction

A mix network, or mix-net, is a network of mix-servers designed to remove the link between ciphertexts and their senders. To achieve this goal, a mix-server of a mix-net initially obtains a list of ciphertexts $(z_i)_{i=1}^n$. It then re-randomizes and permutes this list, and outputs the new list $(z'_i)_{i=1}^n$ together with a non-interactive zero knowledge (NIZK, [BFM88]) shuffle argument [SK95] that proves the re-randomization and permutation was done correctly, without leaking any side information. If enc is a multiplicatively homomorphic public-key cryptosystem like Elgamal [Elg85], a shuffle argument convinces the verifier that there exists a permutation ψ and a vector $\boldsymbol{t}$ of randomizers such that $z'_i = z_{\psi(i)} \cdot \mathsf{enc}_{\mathsf{pk}}(1; t_i)$, without revealing any information about ψ or $\boldsymbol{t}$. Mix-nets improve security against malicious voting servers in e-voting [Cha81,PIK93]. Other applications of mix-nets include anonymous web browsing, payment systems, and secure multiparty computation [KMW12].

It is important to have a *non-interactive* shuffle argument outputting a short bit string that can be verified by anybody (possibly years later) without interacting with the prover. Many NIZK shuffle arguments are known in the random oracle model, see for example [FS01,Nef01,Fur05,TW10,Gro10a]. Since the random oracle model is only a heuristic, it is strongly recommended to construct NIZK arguments in the common reference string (CRS) model [BFM88], without using random oracles. See App. A for motivation for using the CRS model.[1] We note that the most efficient shuffle arguments in the random oracle model like [Gro10a] also require a CRS.

Up to now, only two NIZK shuffle arguments in the CRS model have been proposed, by Groth and Lu [GL07] and Lipmaa and Zhang [LZ12,LZ13], both of which are significantly slower than the fastest arguments in the random oracle model (see Tbl. 1). The Groth-Lu shuffle argument only provides culpable soundness [GL07,GOS12] in the sense that if a malicious prover can create an accepting shuffle argument for an incorrect statement, then this prover *together* with a party that knows the secret key can break the underlying security assumptions. Relaxation of the soundness property is unavoidable, since [AF07] showed that only languages in **P/poly** can have direct black-box adaptive perfect NIZK arguments under a (polynomial) cryptographic hardness assumption. If the underlying cryptosystem is IND-CPA secure, then the shuffle language is *not* in **P/poly**, and thus it is necessary to use knowledge assumptions [Dam91] to

[1] In a practical implementation of a mix-net, one can use the random oracle model also for other purposes, such as to construct a pseudo-number generator or a public-key cryptosystem. In most of such cases, it is known how to avoid the random oracle model, although this almost always incurs some additional cost.

Table 1. A comparison of different NIZK shuffle arguments, compared with the computationally most efficient known shuffle argument in the random oracle model [Gro10a].

	[GL07]	[LZ13]	This work	[Gro10a]
|CRS|	$2n+8$	$7n+6$	$8n+17$	$n+1$
Communication	$15n+120$ $(+3n)$	$6n+11$ $(+6n)$	$7n+2$ $(+2n)$	$480n$ bits
pro's comp.	$51n+246$ $(+3n)$	$22n+11$ $(+6n)$	$16n+3$ $(+2n)$	$6n$ $(+2n)$
ver's comp.	$75n+282$	$28n+18$	$18n+6$	$6n$ exp.
Lifted	No	Yes	No	No
Soundness	Culp. sound	White-box sound	Culp. sound	Sound
Arg. of knowl.	no	yes	yes	yes
PKE (knowl. assm.)	no	yes	yes	no
Random oracle	no			yes

prove its adaptive soundness. Moreover, [GL07] argued that culpable soundness is a sufficient security notion for shuffles, since in any real-life application of the shuffle argument there exists some coalition of parties who knows the secret key.

Lipmaa and Zhang [LZ12] proposed a more efficient NIZK shuffle argument by using knowledge assumptions under which they also bypassed the impossibility result of [AF07] and proved that their shuffle argument is sound. However, their shuffle argument is sound only under the assumption that there is an extractor that has access to the random coins of all encrypters, e.g., all voters, allowing her to extract all plaintexts and randomizers. We say in this case that the argument is *white-box sound*. White-box soundness is clearly a weaker security notion than culpable soundness of [GL07], and it would be good to avoid it.

In addition, the use of knowledge assumptions in [LZ12] forces the underlying BBS [BBS04] cryptosystem to include knowledge components (so ciphertexts are twice as long) and be lifted (meaning that one has to solve discrete logarithm to decrypt, so plaintexts must be small). Thus, one has to use a random oracle-less range argument [RKP09,CLZ12,FLZ13,Lip14] to guarantee that the plaintexts are small and thus to guarantee the soundness of the *shuffle* argument (see [LZ12] for a discussion). While range proofs only have to be verified once (e.g., by only one mix-server), this still means that the shuffle argument of [LZ12] is somewhat slower than what is given in Tbl. 1. Moreover, in the case of e-voting, using only small plaintexts restricts the applicability of a shuffle argument to only certain voting mechanisms like majority. On the other hand, a mechanism such as Single Transferable Vote would likely be unusable due to the length of the ballots.

Tbl. 1 provides a brief comparison between known NIZK shuffle arguments in the CRS model and the most computationally efficient known shuffle argument in the random oracle model [Gro10a]. We emphasize that the values in parentheses show the cost of computing and communicating the shuffled ciphertexts themselves, and must be added to the rest. Moreover, the cost of the shuffle argument from [LZ12] should include the cost of a range argument. Unless written otherwise, the communication and the CRS length are given in group elements, the prover's computational complexity is given in exponentiations, and the verifier's computational complexity is given in bilinear pairings. In each row, highlighted cells denote the best efficiency or best security (e.g., not requiring the PKE assumption) among arguments in the CRS model. Of course, a full efficiency comparison can only be made after implementing the different shuffle arguments.

This brings us to the main question of the current paper:

> *Is it possible to construct an NIZK shuffle argument in the CRS model that is comparable in efficiency with existing random oracle model NIZK shuffle arguments? Moreover, can one do it while minimizing the use of knowledge assumptions (i.e., not requiring the knowledge extractor to have access to the random coins used by all encrypters) and using a standard, non-lifted, cryptosystem?*

Our Contributions. We give a partial answer to the main question. We propose a new pairing-based NIZK shuffle argument in the CRS model. Differently from [LZ12], we prove the culpable soundness of the new argument instead of white-box soundness. Compared to [GL07], which also achieves culpable soundness, the new argument has 3 times faster proving and more than 4 times faster verification. Compared to [GL07,LZ12], it is based on a more standard cryptosystem (Elgamal). While the new shuffle argument is still at least 2 times slower than the most efficient known random oracle based shuffle arguments, it has almost optimal *online* prover's computation. Of course, a full efficiency comparison can only be made after implementing the different shuffle arguments.

Our construction works as as follows. We first commit to the permutation ψ (by committing separately to first $n-1$ rows of the corresponding permutation matrix $\boldsymbol{\Psi}$) and to the vector $\boldsymbol{t}$ of blinding randomizers. Here, we use the *polynomial commitment scheme* (see Sect. 2) with $\mathsf{com}(\mathsf{ck}; \boldsymbol{m}; r) = (g_1, g_2^{\gamma})^{r P_0(\chi) + \sum_{i=1}^{n} m_i P_i(\chi)} \in \mathbb{G}_1 \times \mathbb{G}_2$, in pairing-based setting, where $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear pairing, g_i is a generator of $\mathbb{G}_i$ for $i \in \{1, 2\}$, $(P_i(X))_{i=0}^{n}$ is a tuple of linearly independent polynomials, χ is a trapdoor, γ is a knowledge secret, and $\mathsf{ck} = ((g_1, g_2^{\gamma})^{P_i(\chi)})_{i=0}^{n}$ is the CRS. For different values of $P_i(X)$, variants of this commitment scheme have been proposed before [GJM02,Gro10b,Lip12].

We show that $\boldsymbol{\Psi}$ is a correct permutation matrix by constructing n witness-indistinguishable succinct *unit vector arguments*, each of which guarantees that a row of $\boldsymbol{\Psi}$ is a unit vector, for implicitly constructed $\boldsymbol{\Psi}_n = \boldsymbol{1_n} - \sum_{i=1}^{n-1} \boldsymbol{\Psi}_i$. We use the recent square span programs (SSP, [DFGK14]) approach to choose the polynomials $P_i(X) = y_i(X)$ so that the unit vector argument is efficient. Since unit vectors are used in many contexts, we hope this argument is of independent interest.

After that, we postulate a natural concrete verification equation for shuffles, and construct the shuffle argument from this. If privacy were not an issue (and thus $z'_i = z_{\psi(i)}$ for every i), the verification equation would just be the tautology $\prod_{i=1}^{n} \hat{e}(z'_i, g_2^{y_i(\chi)}) =^? \prod_{i=1}^{n} \hat{e}(z_i, g_2^{y_{\psi^{-1}(i)}(\chi)})$. Clearly, if the prover is honest, this equation holds. However, it does not yet guarantee soundness, since an adversary can use $g_1^{y_j(\chi)}$ (given in the CRS) to create $(z'_i)_{i=1}^{n}$ in a malicious way. To eliminate this possibility, by roughly following an idea from [GL07], we also verify that $\prod_{i=1}^{n} \hat{e}(z'_i, g_2^{\hat{y}_i(\chi)}) =^? \prod_{i=1}^{n} \hat{e}(z_i, g_2^{\hat{y}_{\psi^{-1}(i)}(\chi)})$ for some well-chosen polynomials $\hat{y}_i(X)$. (We note that instead of n univariate polynomials, [GL07] used n random variables χ_i, increasing the size of the secret key to $\Omega(n)$ bits.)

To show that the verifications are instantiated correctly, we also need a *same-message argument* that shows that commitments w.r.t. two tuples of polynomials $(y_i(X))_{i=1}^{n}$ and $(\hat{y}_i(X))_{i=1}^{n}$ commit to the same plaintext vectors. We construct an efficient same-message argument by using an approach that is (again, roughly) motivated by the QAP-based approach of [GGPR13]. This argument is an argument of knowledge, given that the polynomials $\hat{y}_i(X)$ satisfy an additional restriction.

Since we also require privacy, the actual verification equations are more complicated. In particular, $z'_i = z_{\psi(i)} \cdot \mathsf{enc}_{\mathsf{pk}}(1; t_i)$, and (say) $g_2^{y_{\psi^{-1}(i)}(\chi)}$ is replaced by the second element $g_2^{\gamma(r_i y_0(\chi) + y_{\psi^{-1}(i)}(\chi))}$ of a commitment to $\boldsymbol{\Psi}_i$. The resulting complication is minor (it requires one to include into the shuffle argument a single ciphertext $U \in \mathbb{G}_1^2$ that compensates for the added randomness). The full shuffle argument consists of commitments to $\boldsymbol{\Psi}$ and to $\boldsymbol{t}$ (both committed twice, w.r.t. the polynomials $(y_i(X))_{i=0}^{n}$ and $(\hat{y}_i(X))_{i=0}^{n}$), n unit vector arguments (one for each row of $\boldsymbol{\Psi}$), $n-1$ same-message arguments, and finally U.

If $\hat{y}_i(X)$ are well-chosen, then from the two verification equations and the soundness of the unit vector and same-message arguments it follows, under a new computational assumption PSP (*Power Simultaneous Product*, related to an assumption from [GL07]), that $z'_i = z_{\psi(i)}$ for every i.

We prove culpable soundness [GL07,GOS12] of the new argument. Since the security of the new shuffle argument does not depend on the cryptosystem either having knowledge components or being lifted, we can use Elgamal encryption [Elg85] instead of the non-standard knowledge BBS encryption introduced in [LZ12]. Since the cryptosystem does not have to be lifted, one can use more complex voting mechanisms with more complex ballots. The use of knowledge assumptions means that the new argument is an argument of knowledge.

The new shuffle argument can be largely precomputed by the prover and forwarded to the verifier even before the common input (i.e., ciphertexts) arrive. Similarly, the verifier can perform a large part of verification before receiving the ciphertexts. (See [Wik09] for motivation for precomputation.) The prover's computation in the online phase is dominated by just two $(n+1)$-wide multi-exponentiations (the computation of U).

The multi-exponentiations can be parallelized; this is important in practice due to the wide availability of highly parallel graphics processors.

Main Technical Challenges. While the main objective of the current work is efficiency, we emphasize that several steps of the new shuffle argument are technically involved. Throughout the paper, we use and combine very recent techniques from the design of efficient succinct non-interactive arguments of knowledge (SNARKs, [GGPR13,PGHR13,DFGK14], that are constructed with the main goal of achieving efficient verifiable computation) with quite unrelated techniques from the design of efficient shuffle arguments [GL07,LZ12].

The security of the new shuffle argument relies on a new assumption, PSP. We prove that PSP holds in the generic bilinear group model, given that polynomials $\hat{y}_i(X)$ satisfy a very precise criterion. For the security of the SSP-based unit vector argument, we need $y_i(X)$ to satisfy another criterion, and for the security of the same-message argument, we need $y_i(X)$ and $\hat{y}_i(X)$ to satisfy a third criterion. The fact that polynomials $y_i(X)$ and $\hat{y}_i(X)$ that satisfy all three criteria exist is not a priori clear; $y_i(X)$ and $\hat{y}_i(X)$ (see Prop. 3) are also unlike any polynomials from the related literature on non-interactive zero knowledge.

Finally, the PSP assumption was carefully chosen so it will hold in the generic bilinear group model, and so the reduction from culpable soundness of the shuffle argument to the PSP assumption would work. While the PSP assumption is related to the SP assumption from [GL07], the situation in [GL07] was less fragile due to the use of independent random variables X_i and X_i^2 instead of polynomials $y_i(X)$ and $\hat{y}_i(X)$. In particular, the same-message argument is trivial in the case of using independent random variables.

This is the full version of a conference paper [FL16].

2 Preliminaries

Let n be the number of ciphertexts to be shuffled. Let S_d be the symmetric group of d elements. Let $\mathbb{G}^*$ denote the group $\mathbb{G}$ without its identity element. For $a \leq b$, let $[a\,..\,b] := \{c \in \mathbb{Z} : a \leq c \leq b\}$. Denote $(a, b)^c := (a^c, b^c)$. For a set of polynomials $\mathcal{F}$ that have the same domain, denote $g^{\mathcal{F}(\boldsymbol{a})} := (g^{f(\boldsymbol{a})})_{f\in\mathcal{F}}$.

A *permutation matrix* is a Boolean matrix with exactly one 1 in every row and column. If ψ is a permutation then the corresponding permutation matrix $\boldsymbol{\Psi}_\psi$ is such that $(\boldsymbol{\Psi}_\psi)_{ij} = 1$ iff $j = \psi(i)$. Thus $(\boldsymbol{\Psi}_{\psi^{-1}})_{ij} = 1$ iff $i = \psi(j)$. Clearly, $\boldsymbol{\Psi}$ is a permutation matrix iff its every row is a unit vector, and the sum of all its row vectors is equal to the all-ones vector $\mathbf{1}_n$.

Let κ be the security parameter. We denote $f(\kappa) \approx_\kappa g(\kappa)$ if $|f(\kappa) - g(\kappa)|$ is negligible in κ. We abbreviate (non-uniform) probabilistic-polynomial time by (NU)PPT. On input 1^κ, a *bilinear map generator* BP returns $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e})$, where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are multiplicative cyclic groups of prime order p, and $\hat{e}$ is an efficient bilinear map $\hat{e}\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ that satisfies the following two properties, where g_1 (resp., g_2) is an arbitrary generator of $\mathbb{G}_1$ (resp., $\mathbb{G}_2$): (i) $\hat{e}(g_1, g_2) \neq 1$, and (ii) $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$. Thus, $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1^c, g_2^d)$ iff $ab \equiv cd \pmod p$. We give BP another input, n (related to the input length), and allow p to depend on n. Finally, we assume that all algorithms that handle group elements reject if their inputs do not belong to corresponding groups.

We will now give short explanations of the main knowledge assumptions. For formal definitions see App I. Let $1 < d(n) < d^*(n) = \mathrm{poly}(\kappa)$ be two functions. We say that BP is

- $d(n)$-*PDL (Power Discrete Logarithm, [Lip12]) secure* if any NUPPT adversary, given values $((g_1, g_2)^{\chi^i})_{i=0}^{d(n)}$, has negligible probability of producing χ.
- $(d(n), d^*(n))$-*PCDH (Power Computational Diffie-Hellman, [GJM02,Gro10b,GGPR13]) secure* if any NUPPT adversary, given values $((g_1, g_2)^{\chi^i})_{i\in[0\,..\,d^*(n)]\setminus\{d(n)+1\}}$, has negligible probability of producing $g_1^{\chi^{d(n)+1}}$.
- $d(n)$-*TSDH (Target Strong Diffie-Hellman, [BB04,PGHR13]) secure* if any NUPPT adversary, given values $((g_1, g_2)^{\chi^i})_{i=0}^{d(n)}$, has negligible probability of producing a pair of values $\left(r, \hat{e}(g_1, g_2)^{1/(\chi - r)}\right)$ where $r \neq \chi$.

For algorithms A and X_{A}, we write $(y; y') \leftarrow (\mathsf{A}||X_{\mathsf{A}})(\chi)$ if A on input χ outputs y, and X_{A} on the same input (including the random tape of A) outputs y' [AF07]. We will need knowledge assumptions w.r.t. up to

2 knowledge secrets γ_i. Let m be the number of different knowledge secrets in any concrete argument, in the current paper $m \leq 2$. Let $\mathcal{F} = (P_i)_{i=0}^n$ be a tuple of univariate polynomials, and $\mathcal{G}_1$ (resp., $\mathcal{G}_2$) be a tuple of univariate (resp., m-variate) polynomials. For $i \in [1 \mathinner{..} m]$, BP is $(\mathcal{F}, \mathcal{G}_1, \mathcal{G}_2, \gamma_i)$*-PKE (Power Knowledge of Exponent, [Gro10b]) secure* if for any NUPPT adversary A there exists a NUPPT extractor X_A, such that

$$\Pr\left[\begin{array}{l} \mathsf{gk} \leftarrow \mathsf{BP}(1^\kappa, n), (g_1, g_2, \chi) \leftarrow_r \mathbb{G}_1^* \times \mathbb{G}_2^* \times \mathbb{Z}_p, \boldsymbol{\gamma} \leftarrow_r \mathbb{Z}_p^m, \\ \boldsymbol{\gamma_{-i}} = (\gamma_1, \dots, \gamma_{i-1}, \gamma_{i+1}, \dots, \gamma_m), \mathsf{aux} \leftarrow \left(g_1^{\mathcal{G}_1(\chi)}, g_2^{\mathcal{G}_2(\chi, \boldsymbol{\gamma_{-i}})}\right), \\ (h_1, h_2; (a_i)_{i=0}^n) \leftarrow (\mathsf{A} || X_\mathsf{A})(\mathsf{gk}; (g_1, g_2^{\gamma_i})^{\mathcal{F}(\chi)}, \mathsf{aux}) : \\ \hat{e}(h_1, g_2^{\gamma_i}) = \hat{e}(g_1, h_2) \wedge h_1 \neq g_1^{\sum_{i=0}^n a_i P_i(\chi)} \end{array}\right] \approx_\kappa 0 \enspace .$$

Here, aux can be seen as the common auxiliary input to A and X_A that is generated by using benign auxiliary input generation [BCPR14]. The definition implies that aux may depend on $\boldsymbol{\gamma_{-i}}$ but not on γ_i. If $\mathcal{F} = (X^i)_{i=0}^d$ for some $d = d(n)$, then we replace the first argument in $(\mathcal{F}, \dots)$-PKE with d. If $m = 1$, then we omit the last argument γ_i in $(\mathcal{F}, \dots, \gamma_i)$-PKE.

We will use the Elgamal cryptosystem [Elg85] $\mathit{\Pi} = (\mathsf{BP}, \mathsf{genpkc}, \mathsf{enc}, \mathsf{dec})$, defined as follows, in the bilinear setting.

Setup (1^κ): Let $\mathsf{gk} \leftarrow (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}) \leftarrow \mathsf{BP}(1^\kappa)$.

Key Generation $\mathsf{genpkc}(\mathsf{gk})$: Let $g_1 \leftarrow_r \mathbb{G}_1^*$. Set the secret key $\mathsf{sk} \leftarrow_r \mathbb{Z}_p$, and the public key $\mathsf{pk} \leftarrow (g_1, h = g_1^{\mathsf{sk}})$. Output $(\mathsf{pk}, \mathsf{sk})$.

Encryption $\mathsf{enc}_\mathsf{pk}(m; r)$: To encrypt a message $m \in \mathbb{G}_1$ with randomizer $r \in \mathbb{Z}_p$, output the ciphertext $\mathsf{enc}_\mathsf{pk}(m; r) = \mathsf{pk}^r \cdot (1, m) = (g^r, m h^r)$.

Decryption $\mathsf{dec}_\mathsf{sk}(c_1, c_2)$: $m = c_2 / c_1^{\mathsf{sk}} = m h^r / h^r = m$.

Elgamal is clearly multiplicatively homomorphic. In particular, if $t \leftarrow_r \mathbb{Z}_p$, then for *any* m and r, $\mathsf{enc}_\mathsf{pk}(m; r) \cdot \mathsf{enc}_\mathsf{pk}(1; t) = \mathsf{enc}_\mathsf{pk}(m; r + t)$ is a random encryption of m. Elgamal is IND-CPA secure under the XDH assumption.

An extractable trapdoor commitment scheme consists of two efficient algorithms gencom (that outputs a CRS and a trapdoor) and com (that, given a CRS, a message and a randomizer, outputs a commitment), and must satisfy the following four security properties.

Computational binding: without access to the trapdoor, it is intractable to open a commitment to two different messages.

Trapdoor: given access to the original message, the randomizer and the trapdoor, one can open the commitment to any other message.

Perfect hiding: commitments of any two messages have the same distribution.

Extractable: given access to the CRS, the commitment, and the random coins of the committer, one can obtain the value that the committer committed to.

See, e.g., [Gro10b] for formal definitions.

We use the following extractable trapdoor *polynomial commitment scheme* that generalizes various earlier commitment schemes [GJM02,Gro10b,Lip12]. Let $n = \mathrm{poly}(\kappa)$, $n > 0$, be an integer. Let $P_i(X) \in \mathbb{Z}_p[X]$, for $i \in [0 \mathinner{..} n]$, be $n + 1$ linearly independent low-degree polynomials. First, $\mathsf{gencom}(1^\kappa, n)$ generates $\mathsf{gk} \leftarrow \mathsf{BP}(1^\kappa, n)$, picks $g_1 \leftarrow_r \mathbb{G}_1^*$, $g_2 \leftarrow_r \mathbb{G}_2^*$, and then outputs the CRS $\mathsf{ck} \leftarrow ((g_1^{P_i(\chi)}, g_2^{\gamma P_i(\chi)})_{i=0}^n)$ for $\chi \leftarrow_r \mathbb{Z}_p \setminus \{j : P_0(j) = 0\}$ and $\gamma \leftarrow_r \mathbb{Z}_p$. The trapdoor is equal to $\mathsf{td}_\mathsf{com} = \chi$.

The commitment of $\boldsymbol{a} \in \mathbb{Z}_p^n$, given a randomizer $r \leftarrow_r \mathbb{Z}_p$, is $\mathsf{com}(\mathsf{ck}; \boldsymbol{a}; r) := (g_1^{P_0(\chi)}, g_2^{\gamma P_0(\chi)})^r \cdot \prod_{i=1}^n (g_1^{P_i(\chi)}, g_2^{\gamma P_i(\chi)})^{a_i} \in \mathbb{G}_1 \times \mathbb{G}_2$. The validity of a commitment (A_1, A_2^γ) can be checked by verifying that $\hat{e}(A_1, g_2^{\gamma P_0(\chi)}) = \hat{e}(g_1^{P_0(\chi)}, A_2^\gamma)$. To open a commitment, the committer sends $(\boldsymbol{a}, r)$ to the verifier.

The rather standard proof of the following theorem is given in App. B.

Theorem 1. *Denote $\mathcal{F}_\mathsf{com} = (P_i(X))_{i=0}^n$. The polynomial commitment scheme is perfectly hiding and trapdoor. Let $d := \max_{f \in \mathcal{F}_\mathsf{com}} (\deg f)$. If BP is d-PDL secure, then it is computationally binding. If BP is $(\mathcal{F}_\mathsf{com}, \emptyset, \emptyset)$-PKE secure, then it is extractable.*

Alternatively, we can think of com as being a commitment scheme that does not depend on the concrete polynomials at all, and the description of P_i is just given as a part of ck. We instantiate the polynomial commitment scheme with concrete polynomials later in Sect. 3 and Sect. 6.

An NIZK argument for a group-dependent language $\mathcal{L}$ consists of four algorithms, setup, gencrs, pro and ver. The setup algorithm setup takes as input 1^κ and n (the input length), and outputs the group description gk. The CRS generation algorithm gencrs takes as input gk and outputs the prover's CRS crs_p, the verifier's CRS crs_v, and a trapdoor td. (td is only required when the argument is zero-knowledge.) The distinction between crs_p and crs_v is only important for efficiency. The prover pro takes as input gk and crs_p, a statement u, and a witness w, and outputs an argument π. The verifier ver takes as input gk and crs_v, a statement u, and an argument π, and either accepts or rejects.

Some of the properties of an argument are: (i) *perfect completeness* (honest verifier always accepts honest prover's argument), (ii) *perfect witness-indistinguishability* (argument distributions corresponding to all allowable witnesses are equal), (iii) *perfect zero knowledge* (there exists an efficient simulator that can, given u, $(\mathsf{crs}_p, \mathsf{crs}_v)$ and td, output an argument that comes from the same distribution as the argument produced by the prover), (iv) *adaptive computational soundness* (if $u \notin \mathcal{L}$, then an arbitrary non-uniform probabilistic polynomial time prover has negligible success in creating a satisfying argument), and (v) *adaptive computational culpable soundness* [GL07,GOS12] (if $u \notin \mathcal{L}$, then an arbitrary NUPPT prover has negligible success in creating a satisfying argument together with a witness that $u \notin \mathcal{L}$). An argument is an *argument of knowledge*, if from an accepting argument it follows that the prover knows the witness. See App. J for formal definitions.

3 Unit Vector Argument

In a unit vector argument, the prover aims to convince the verifier that he knows how to open a commitment (A_1, A_2^γ) to *some* $(\boldsymbol{e}_I, r)$, where $\boldsymbol{e}_I$ denotes the Ith unit vector for $I \in [1 \mathinner{..} n]$. We construct the unit vector argument by using square span programs (SSP-s, [DFGK14], an especially efficient variant of the quadratic arithmetic programs of [GGPR13]).

Clearly, $\boldsymbol{a} \in \mathbb{Z}_p^n$ is a unit vector iff the following $n+1$ conditions hold:

- $a_i \in \{0, 1\}$ for $i \in [1 \mathinner{..} n]$ (i.e., $\boldsymbol{a}$ is Boolean), and
- $\sum_{i=1}^n a_i = 1$.

We use the methodology of [DFGK14] to obtain an efficient NIZK argument out of these conditions. Let $\{0, 2\}^{n+1}$ denote the set of $(n+1)$-dimensional vectors where every coefficient is from $\{0, 2\}$, let $\circ$ denote the Hadamard (entry-wise) product of two vectors, let $V := \left(\begin{smallmatrix} 2 \cdot I_{n \times n} \\ \mathbf{1}_n^\top \end{smallmatrix}\right) \in \mathbb{Z}_p^{(n+1) \times n}$ and $\boldsymbol{b} := \left(\begin{smallmatrix} \mathbf{0}_n \\ 1 \end{smallmatrix}\right) \in \mathbb{Z}_p^{n+1}$. Clearly, the above $n+1$ conditions hold iff $V\boldsymbol{a} + \boldsymbol{b} \in \{0, 2\}^{n+1}$, i.e.,

$$(V\boldsymbol{a} + \boldsymbol{b} - \mathbf{1}_{n+1}) \circ (V\boldsymbol{a} + \boldsymbol{b} - \mathbf{1}_{n+1}) = \mathbf{1}_{n+1} \quad . \tag{1}$$

Let ω_i, $i \in [1 \mathinner{..} n+1]$ be $n+1$ different values. Let $Z(X) := \prod_{i=1}^{n+1} (X - \omega_i)$ be the unique degree $n+1$ monic polynomial, such that $Z(\omega_i) = 0$ for all $i \in [1 \mathinner{..} n+1]$. Let the ith Lagrange basis polynomial $\ell_i(X) := \prod_{i,j \in [1 \mathinner{..} n+1], j \neq i} ((X - \omega_j)/(\omega_i - \omega_j))$ be the unique degree n polynomial, s.t. $\ell_i(\omega_i) = 1$ and $\ell_i(\omega_j) = 0$ for $j \neq i$. For a vector $\boldsymbol{x} \in \mathbb{Z}_p^{n+1}$, let $L_{\boldsymbol{x}}(X) = \sum_{i=1}^{n+1} x_i \ell_i(X)$ be a degree n polynomial that interpolates $\boldsymbol{x}$, i.e., $L_{\boldsymbol{x}}(\omega_i) = x_i$.

For $i \in [1 \mathinner{..} n]$, let $y_i(X)$ be the polynomial that interpolates the ith column of the matrix V. That is, $y_i(X) = 2\ell_i(X) + \ell_{n+1}(X)$ for $i \in [1 \mathinner{..} n]$. Let $y_0(X) = -1 + \ell_{n+1}(X)$ be the polynomial that interpolates $\boldsymbol{b} - \mathbf{1}_{n+1}$. We will use an instantiation of the polynomial commitment scheme with $\mathcal{F}_{\mathsf{com}} = (Z(X), (y_i(X))_{i=1}^n)$.

As in [DFGK14], we arrive at the polynomial $Q(X) = (\sum_{i=1}^n a_i y_i(X) + y_0(X))^2 - 1 = (y_I(X) + y_0(X))^2 - 1$ (here, we used the fact that $\boldsymbol{a} = \boldsymbol{e}_I$ for some $I \in [1 \mathinner{..} n]$), such that $\boldsymbol{a}$ is a unit vector iff $Z(X) \mid Q(X)$. As in [GGPR13,DFGK14], to obtain privacy, we now add randomness to $Q(X)$, arriving at the degree $2(n+1)$ polynomial $Q_{wi}(X) = (rZ(X) + y_I(X) + y_0(X))^2 - 1$. By [GGPR13,DFGK14], Eq. (1) holds iff

(i) $Q_{wi}(X) = (A(X) + y_0(X))^2 - 1$, where $A(X) = r_a Z(X) + \sum_{i=1}^n a_i y_i(X) \in \mathrm{span}(\mathcal{F}_{\mathsf{com}})$, and

(ii) $Z(X) \mid Q_{wi}(X)$.
An honest prover computes the degree $\leq n+1$ polynomial $\pi_{wi}(X) \leftarrow Q_{wi}(X)/Z(X) \in \mathbb{Z}_p[X]$, and sets the argument to be equal to $\pi^*_{uv} := g_1^{\pi_{wi}(\chi)}$ for a secret χ that instantiates X. If it exists, $\pi_{wi}(X) := Q_{wi}(X)/Z(X)$ is equal to $r^2 Z(X) + r \cdot 2(y_I(X) + y_0(X)) + \mathit{\Pi}_I(X)$, where for $i \in [1 \,..\, n]$, $\mathit{\Pi}_i(X) := ((y_i(X) + y_0(X))^2 - 1)/Z(X)$ is a degree $\leq n-1$ polynomial and $Z(X) \mid ((y_i(X) + y_0(X))^2 - 1)$. Thus, computing π^*_{uv} uses two exponentiations.

We use a knowledge (PKE) assumption in a standard way to guarantee that $A(X)$ is in the span of $\{X^i\}_{i=0}^{n+1}$. As in [GGPR13,DFGK14], we then guarantee condition (i) by using a PCDH assumption and condition (ii) by using a TSDH assumption. Here, we use the same technique as in [GGPR13] and subsequent papers by introducing an additional secret, β, and adding one group element A_1^β to the argument.
System parameters: Let com be the polynomial commitment scheme and let $\mathcal{F}_{\mathsf{com}} = (Z(X), (y_i(X))_{i=1}^n)$.
Setup $\mathsf{setup}_{uv}(1^\kappa, n)$**:** Let $\mathsf{gk} \leftarrow \mathsf{BP}(1^\kappa, n)$.
CRS generation $\mathsf{gencrs}_{uv}(\mathsf{gk})$**:** Let $(g_1, g_2, \chi, \beta, \gamma) \leftarrow_r \mathbb{G}_1^* \times \mathbb{G}_2^* \times \mathbb{Z}_p^3$, s.t. $Z(\chi) \neq 0$. Set $\mathsf{ck} \leftarrow (g_1, g_2^\gamma)^{\mathcal{F}_{\mathsf{com}}(\chi)}$, $\mathsf{crs}_{uv,p} \leftarrow (\mathsf{ck}, (g_1^{2(y_i(\chi)+y_0(\chi))}, g_1^{\mathit{\Pi}_i(\chi)})_{i=1}^n, g_1^{\beta \cdot \mathcal{F}_{\mathsf{com}}(\chi)})$, $\mathsf{crs}_{uv,v} \leftarrow (g_1, g_1^{y_0(\chi)}, g_2^\gamma, g_2^{\gamma y_0(\chi)}, g_2^{\gamma Z(\chi)}, g_2^{\gamma\beta}, \hat{e}(g_1, g_2^\gamma)^{-1})$. Return $\mathsf{crs}_{uv} = (\mathsf{crs}_{uv,p}, \mathsf{crs}_{uv,v})$.
Common input: $(A_1, A_2^\gamma) = ((g_1, g_2^\gamma)^{Z(\chi)})^r (g_1, g_2^\gamma)^{y_I(\chi)}$ where $I \in [1 \,..\, n]$.
Proving $\mathsf{pro}_{uv}(\mathsf{gk}, \mathsf{crs}_{uv,p}; A_1, A_2^\gamma; w_{uv} = (\boldsymbol{a} = \boldsymbol{e_I}, r))$**:** Set $\pi^*_{uv} \leftarrow (g_1^{Z(\chi)})^{r^2} \cdot (g_1^{2(y_I(\chi)+y_0(\chi))})^r \cdot g_1^{\mathit{\Pi}_I(\chi)}$. Set $A_1^\beta \leftarrow (g_1^{\beta Z(\chi)})^r g_1^{\beta y_I(\chi)}$. Output $\pi_{uv} = (\pi^*_{uv}, A_1^\beta) \in \mathbb{G}_1^2$.
Verification $\mathsf{ver}_{uv}(\mathsf{gk}, \mathsf{crs}_{uv,v}; A_1, A_2^\gamma; \pi_{uv})$**:** Parse π_{uv} as $\pi_{uv} = (\pi^*_{uv}, A_1^\beta)$. Verify that (1) $\hat{e}(\pi^*_{uv}, g_2^{\gamma Z(\chi)}) = \hat{e}(A_1 \cdot g_1^{y_0(\chi)}, A_2^\gamma \cdot g_2^{\gamma y_0(\chi)}) \cdot \hat{e}(g_1, g_2^\gamma)^{-1}$,(2) $\hat{e}(g_1, A_2^\gamma) = \hat{e}(A_1, g_2^\gamma)$, and (3) $\hat{e}(A_1, g_2^{\gamma\beta}) = \hat{e}(A_1^\beta, g_2^\gamma)$.
Set $\mathcal{F}_{uv,1} = \{1\} \cup \mathcal{F}_{\mathsf{com}} \cup X_\beta \mathcal{F}_{\mathsf{com}}$ and $\mathcal{F}_{uv,2} = Y\mathcal{F}_{\mathsf{com}} \cup \{Y, YX_\beta\}$. The formal variable X_β (resp., Y) stands for the secret key β (resp., γ). Since other elements of crs_{uv} are only needed for optimization, crs_{uv} can be computed from $\mathsf{crs}^*_{uv} = (g_1^{\mathcal{F}_{uv,1}(\chi,\beta)}, g_2^{\mathcal{F}_{uv,2}(\chi,\beta,\gamma)})$. If $n > 2$ then $1 \notin \operatorname{span}(\{Z(X)\} \cup \{y_i(X)\}_{i=1}^n)$, and thus $\{1, Z(X)\} \cup \{y_i(X)\}_{i=1}^n$ is a basis of all polynomials of degree at most $n+1$. Thus, $\mathcal{F}_{uv,1}$ can be computed iff $\{X^i\}_{i=0}^{n+1} \cup \{X_\beta \mathcal{F}_{\mathsf{com}}\}$ can be computed.

Theorem 2. *The new unit vector argument is perfectly complete and witness-indistinguishable. If* BP *is* $(n+1, 2n+3)$*-PCDH secure,* $(n+1)$*-TSDH secure, and* $(n+1, X_\beta \mathcal{F}_{\mathsf{com}}, \{YX_\beta\})$*-PKE secure, then this argument is an adaptive argument of knowledge.*

The proof of this theorem is given in App. C. The proof of the following proposition is straightforward and thus omitted.

Proposition 1. *The computation of* (π^*_{uv}, A_1^β) *takes one* 2*-wide multi-exponentiation and* 1 *exponentiation in* $\mathbb{G}_1$*. In addition, it takes* 2 *exponentiations (one in* $\mathbb{G}_1$ *and one in* $\mathbb{G}_2$*) in the master argument to compute* (A_1, A_2^γ)*. The verifier computation is dominated by* 6 *pairings.*

4 New Same-Message Argument

In a *same-message argument*, the prover aims to convince the verifier that he knows, given two commitment keys ck and $\widehat{\mathsf{ck}}$ (that correspond to two tuples of polynomials $(P_i(X))_{i=0}^n$ and $(\hat{P}_i(X))_{i=0}^n$, respectively), how to open $(A_1, A_2^\gamma) = \mathsf{com}(\mathsf{ck}; \boldsymbol{m}; r)$ and $(\hat{A}_1, \hat{A}_2^\gamma) = \mathsf{com}(\widehat{\mathsf{ck}}; \boldsymbol{m}; \hat{r})$ as commitments (w.r.t. ck and $\widehat{\mathsf{ck}}$) to the same plaintext vector $\boldsymbol{m}$ (but not necessarily to the same randomizer r).

We propose an efficient same-message argument using $\mathcal{F}_{\mathsf{com}} = (Z(X), (y_i(X))_{i=1}^n)$ as described in Sect. 3. In the shuffle argument, we need $(\hat{P}_i(X))_{i=0}^n$ to satisfy some specific requirements w.r.t. $\mathcal{F}_{\mathsf{com}}$, see Sect. 5. We are free to choose $\hat{P}_i$ otherwise. We concentrate on a choice of $\hat{P}_i$ that satisfies those requirements yet enables us to construct an efficient same-message argument.

Denote $\hat{Z}(X) = \hat{P}_0(X)$. For the same-message argument to be an argument of knowledge *and* efficient, we choose $\hat{P}_i$ such that $(\hat{P}_i(\omega_j))_{j=1}^{n+1} = (y_i(\omega_j))_{j=1}^{n+1} = 2\boldsymbol{e_i} + \boldsymbol{e_{n+1}}$ for $i \in [1 \,..\, n]$. Moreover, $(\hat{Z}(\omega_j))_{j=1}^{n+1} = (Z(\omega_j))_{j=1}^{n+1} = \boldsymbol{0}_{n+1}$.

Following similar methodology as in Sect. 3, define

$$Q_{wi}(X) := (\hat{r}\hat{Z}(X) + \sum_{i=1}^{n} \hat{m}_i \hat{P}_i(X)) - (rZ(X) + \sum_{i=1}^{n} m_i y_i(X)) \enspace .$$

Let $\hat{n}$ be the maximum degree of polynomials in $(y_i(X), \hat{P}_i(X))_{i=0}^{n}$, thus $\deg Q_{wi} \leq \hat{n}$. Since $Q_{wi}(\omega_j) = 2(\hat{m}_j - m_j)$ for $j \in [1 \mathinner{..} n]$, $Q_{wi}(\omega_j) = 0$ iff $m_j = \hat{m}_j$. Moreover, if $\boldsymbol{m} = \hat{\boldsymbol{m}}$ then $Q_{wi}(\omega_{n+1}) = \sum_{i=1}^{n} \hat{m}_i - \sum_{i=1}^{n} m_i = 0$. Hence, $\boldsymbol{m} = \hat{\boldsymbol{m}}$ iff

(i) $Q_{wi}(X) = \hat{A}(X) - A(X)$, where $A(X) \in \mathrm{span}(\{Z(X)\} \cup \{y_i(X)\}_{i=1}^{n})$, and $\hat{A}(X) \in \mathrm{span}(\{\hat{Z}(X)\} \cup \{\hat{P}_i(X)\}_{i=1}^{n})$, and

(ii) there exists a degree $\leq \hat{n} - (n+1)$ polynomial $\pi_{wi}(X) = Q_{wi}(X)/Z(X)$.

If the prover is honest, then $\pi_{wi}(X) = \hat{r}\hat{Z}(X)/Z(X) - r + \sum m_i \cdot ((\hat{P}_i(X) - y_i(X))/Z(X))$. Note that we do not need that $Q_{wi}(X) = 0$ as a polynomial, we just need that $Q_{wi}(\omega_i) = 0$, which is a deviation from the strategy usually used in QAP/QSP-based arguments [GGPR13].

We guarantee the conditions similarly to Sect. 3. The description of the argument follows. (Since it is derived as in Sect. 3, we omit further explanations.)

System parameters: Let $n = \mathrm{poly}(\kappa)$. Let com be the polynomial commitment scheme and let $\mathcal{F}_{\mathsf{com}} = (Z(X), (y_i)_{i=1}^{n})$ and $\hat{\mathcal{F}}_{\mathsf{com}} = (\hat{Z}(X), (\hat{P}_i)_{i=1}^{n})$, where $\hat{P}_i(X)$ is such that $y_i(\omega_j) = \hat{P}_i(\omega_j)$ for $i \in [0 \mathinner{..} n+1]$ and $j \in [1 \mathinner{..} n+1]$.

Setup $\mathsf{setup}_{sm}(1^\kappa, n)$**:** Let $\mathsf{gk} \leftarrow \mathsf{BP}(1^\kappa, n)$.

CRS generation $\mathsf{gencrs}_{sm}(\mathsf{gk})$**:** Let $(g_1, g_2, \chi, \beta, \gamma, \hat{\gamma}) \leftarrow_r \mathbb{G}_1^* \times \mathbb{G}_2^* \times \mathbb{Z}_p^4$ with $Z(\chi) \neq 0$. Set $\mathsf{ck} \leftarrow (g_1, g_2^{\gamma})^{\mathcal{F}_{\mathsf{com}}(\chi)}$ and $\widehat{\mathsf{ck}} \leftarrow (g_1, g_2^{\hat{\gamma}})^{\hat{\mathcal{F}}_{\mathsf{com}}(\chi)}$. Let $\mathsf{crs}_{sm,p} \leftarrow (\mathsf{ck}, \widehat{\mathsf{ck}}, g_1^{\beta \cdot \mathcal{F}_{\mathsf{com}}(\chi)}, g_1^{\hat{Z}(\chi)/Z(\chi)}, g_1, (g_1^{(\hat{P}_i(\chi) - y_i(\chi))/Z(\chi)})_{i=1}^{n})$, and $\mathsf{crs}_{sm,v} \leftarrow (g_1, g_2^{\gamma}, g_2^{\hat{\gamma}}, g_2^{\gamma\beta}, g_2^{\gamma Z(\chi)})$. Return $\mathsf{crs}_{sm} = (\mathsf{crs}_{sm,p}, \mathsf{crs}_{sm,v})$.

Common input: $(A_1, A_2^{\gamma}) = \mathsf{com}(\mathsf{ck}; \boldsymbol{m}; r), (\hat{A}_1, \hat{A}_2^{\hat{\gamma}}) = \mathsf{com}(\widehat{\mathsf{ck}}; \boldsymbol{m}; \hat{r})$.

Argument generation $\mathsf{pro}_{sm}(\mathsf{gk}, \mathsf{crs}_{sm,p}; A_1, A_2^{\gamma}, \hat{A}_1, \hat{A}_2^{\hat{\gamma}}; \boldsymbol{m}, r, \hat{r})$**:** Set $\pi_{sm}^* \leftarrow g_1^{\pi_{wi}(\chi)} = (g_1^{\hat{Z}(\chi)/Z(\chi)})^{\hat{r}} \cdot g_1^{-r} \cdot \prod_{i=1}^{n} (g_1^{(\hat{P}_i(\chi) - y_i(\chi))/Z(\chi)})^{m_i}$. Set $A_1^{\beta} \leftarrow (g_1^{\beta Z(\chi)})^{r} \prod_{i=1}^{n} (g_1^{\beta y_i(\chi)})^{m_i}$. Output $\pi_{sm} = (\pi_{sm}^*, A_1^{\beta}) \in \mathbb{G}_1^2$.

Verification $\mathsf{ver}_{sm}(\mathsf{gk}, \mathsf{crs}_{sm,v}; (A_1, A_2^{\gamma}), (\hat{A}_1, \hat{A}_2^{\hat{\gamma}}); \pi_{sm})$**:**
Parse π_{sm} as $\pi_{sm} = (\pi_{sm}^*, A_1^{\beta})$. Verify that (1) $\hat{e}(g_1, A_2^{\gamma}) = \hat{e}(A_1, g_2^{\gamma})$,(2) $\hat{e}(A_1, g_2^{\gamma\beta}) = \hat{e}(A_1^{\beta}, g_2^{\gamma})$,(3) $\hat{e}(g_1, \hat{A}_2^{\hat{\gamma}}) = \hat{e}(\hat{A}_1, g_2^{\hat{\gamma}})$, and(4) $\hat{e}(\pi_{sm}^*, g_2^{\gamma Z(\chi)}) = \hat{e}(\hat{A}_1/A_1, g_2^{\gamma})$.

Let $\hat{Y}$ be the formal variable corresponding to $\hat{\gamma}$. In the following theorem, it suffices to take $\mathsf{crs}^* = (g_1^{\mathcal{F}_{sm,1}(\chi,\beta)}, g_2^{\mathcal{F}_{sm,2}(\chi,\beta,\gamma,\hat{\gamma})})$, where $\mathcal{F}_{sm,1} = \{1\} \cup \mathcal{F}_{\mathsf{com}} \cup \hat{\mathcal{F}}_{\mathsf{com}} \cup X_\beta \mathcal{F}_{\mathsf{com}} \cup \{\hat{Z}(X)/Z(X)\} \cup \{(\hat{P}_i(X) - y_i(X))/Z(X)\}_{i=1}^{n}$ and $\mathcal{F}_{sm,2} = Y \cdot (\{1, X_\beta\} \cup \mathcal{F}_{\mathsf{com}}) \cup \hat{Y} \cdot (\{1\} \cup \hat{\mathcal{F}}_{\mathsf{com}})$.

Theorem 3. *The same-message argument is perfectly complete and witness-indistinguishable. Let $\hat{n}$ be as above. If* BP *is $(\hat{n}, \hat{n}+n+2)$-PCDH secure, $\hat{n}$-TSDH secure, $(n+1, \mathcal{F}_{sm,1} \setminus (\{1\} \cup \mathcal{F}_{\mathsf{com}}), \mathcal{F}_{sm,2} \setminus Y \cdot (\{1\} \cup \mathcal{F}_{\mathsf{com}}), \gamma)$-PKE secure, and $(\hat{\mathcal{F}}_{\mathsf{com}}, \mathcal{F}_{sm,1} \setminus \hat{\mathcal{F}}_{\mathsf{com}}, \mathcal{F}_{sm,2} \setminus \hat{Y}\hat{\mathcal{F}}_{\mathsf{com}}, \hat{\gamma})$-PKE secure, then this argument is an adaptive argument of knowledge.*

The proof of this theorem is similar to the proof of Thm. 2, see App. D.

The proof of the following proposition is straightforward and thus omitted.

Proposition 2. *The prover's computation is dominated by one $(W+2)$-wide and one $(W+1)$-wide multi-exponentiation in $\mathbb{G}_1$, where $0 \leq W \leq n$ is the number of elements in the vector $\boldsymbol{m}$ that are not in $\{0,1\}$. The verifier's computation is dominated by 8 pairings.*

In the shuffle argument below, the prover uses $r = \hat{r}$, so prover's computation is $2W + 2$ exponentiations. For a unit vector $\boldsymbol{m}$, we additionally have $W = 0$ and computing A_1^{β} and the first two verification steps are already done in the unit vector argument anyway, so the argument only adds 1 exponentiation for the prover, and 4 pairings for the verifier.

5 New Assumption: PSP

We will next describe a new computational assumption (PSP) that is needed in the shuffle argument. The PSP assumption is related to but not equal to the SP assumption (see App. H, that also provides short comparison to PSP) from [GL07]. Interestingly, the generic group proof of the PSP assumption relies on the Schwartz-Zippel lemma, while in most of the known interactive shuffle arguments (like [Nef01]), the Schwartz-Zippel lemma is used in the reduction from the shuffle security to some underlying assumption.

Let let $d(n) > n$ be a function. Let $\hat{\mathcal{F}} = (\hat{P}_i(X))_{i=0}^n$ be a tuple of polynomials. We say $(d(n), \hat{\mathcal{F}})$ is *PSP-friendly*, if the following set is linearly independent: $\hat{\mathcal{F}}_{d(n)} := \{X^i\}_{i=0}^{2d(n)} \cup \{X^i \cdot \hat{P}_j(X)\}_{0 \le i \le d(n), 0 \le j \le n} \cup \{\hat{P}_0(X)\hat{P}_j(X)\}_{j=0}^n$.

Let $(d(n), \hat{\mathcal{F}})$ be PSP-friendly. Let $\mathcal{F} = (P_i(X))_{i=0}^n$ be a tuple of polynomials of degree $\le d(n)$. The $(\mathcal{F}, \hat{\mathcal{F}})$*-Power Simultaneous Product (PSP) assumption* states that for any $n = \mathrm{poly}(\kappa)$ and any NUPPT adversary A,

$$\Pr\left[\begin{array}{l}\mathsf{gk} \leftarrow \mathsf{BP}(1^\kappa, n), (g_1, g_2, \chi) \leftarrow_r \mathbb{G}_1^* \times \mathbb{G}_2^* \times \mathbb{Z}_p, \\ \mathbb{G}_1^{n+2} \ni (t, \hat{t}, (s_i)_{i=1}^n) \leftarrow \mathsf{A}(\mathsf{gk}; ((g_1, g_2)^{\chi^i})_{i=0}^{d(n)}, (g_1, g_2)^{\hat{\mathcal{F}}(\chi)}) : \\ t^{P_0(\chi)} \cdot \prod_{i=1}^n s_i^{P_i(\chi)} = \hat{t}^{\hat{P}_0(\chi)} \cdot \prod_{i=1}^n s_i^{\hat{P}_i(\chi)} = 1 \wedge (\exists i \in [1 \,..\, n] : s_i \neq 1)\end{array}\right] \approx_\kappa 0 \enspace .$$

In this section, we prove that the PSP assumption holds in the generic bilinear group model. PSP-friendliness and the PSP assumption are defined so that both the generic model proof and the reduction from the shuffle soundness to the PSP in Thm. 5 would go through. As in the case of SP, it is essential that two simultaneous products have to hold true; the simpler version of the PSP assumption with only one product (i.e., $t^{P_0(\chi)} \cdot \prod_{i=1}^n s_i^{P_i(\chi)} = 1$) does not hold in the generic bilinear group model. Differently from SP, the PSP assumption incorporates possibly distinct t and $\hat{t}$ since the same-message argument does not guarantee that the randomizers of two commitments are equal.

Generic Security of the PSP Assumption. We will briefly discuss the security of the PSP assumption in the generic bilinear group model. Similarly to [GL07], we start by picking a random asymmetric bilinear group $\mathsf{gk} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}) \leftarrow \mathsf{BP}(1^\kappa)$. We now give a generic bilinear group model proof for the PSP assumption.

Theorem 4. *Let $\mathcal{F} = (P_i(X))_{i=0}^n$ be linearly independent with $1 \notin \mathrm{span}(\mathcal{F})$. Let $d = \max\{\deg P_i(X)\}$ and let $\hat{\mathcal{F}} = (\hat{P}_i(X))_{i=0}^n$ be such that $(d, \hat{\mathcal{F}})$ is PSP-friendly. The $(\mathcal{F}, \hat{\mathcal{F}})$-PSP assumption holds in the generic bilinear group model.*

Proof. Assume there exists a successful adversary A. In the generic bilinear group model, A acts obliviously to the actual representation of the group elements and only performs generic bilinear group operations such as multiplying elements in $\mathbb{G}_i$ for $i \in \{1, 2, T\}$, pairing elements in $\mathbb{G}_1$ and $\mathbb{G}_2$, and comparing elements to see if they are identical. hence it can only produce new elements in $\mathbb{G}_1$ by multiplying existing group elements together.

Recall that the A's input is gk and $\mathsf{crs} = (((g_1, g_2)^{\chi^i})_{i=0}^d, (g_1, g_2)^{\hat{\mathcal{F}}(\chi)})$. Hence, keeping track of the group elements we get that A outputs $t, \hat{t}, s_i \in \mathbb{G}_1$, where $\log_{g_1} t = \sum_{j=0}^d t_j \chi^j + \sum_{j=0}^n t'_j \hat{P}_j(\chi)$, $\log_{g_1} \hat{t} = \sum_{j=0}^d \hat{t}_j \chi^j + \sum_{j=0}^n \hat{t}'_j \hat{P}_j(\chi)$, and $\log_{g_1} s_i = \sum_{j=0}^d s_{ij} \chi^j + \sum_{j=0}^n s'_{ij} \hat{P}_j(\chi)$, for *known* constants t_j, t'_j, $\hat{t}_j$, $\hat{t}'_j$, s_{ij}, s'_{ij}. Taking discrete logarithms of the PSP condition $t^{P_0(\chi)} \cdot \prod_{i=1}^n s_i^{P_i(\chi)} = \hat{t}^{\hat{P}_0(\chi)} \cdot \prod_{i=1}^n s_i^{\hat{P}_i(\chi)} = 1$, we get that the two polynomials (for *known* coefficients)

$$d_1(X) := \left(\sum_{j=0}^{d} t_j X^j + \sum_{j=0}^{n} t'_j \hat{P}_j(X)\right) \cdot P_0(X) + \sum_{i=1}^{n} \left(\sum_{j=0}^{d} s_{ij} X^j + \sum_{j=0}^{n} s'_{ij} \hat{P}_j(X)\right) P_i(X) \ ,$$

$$d_2(X) := \left(\sum_{j=0}^{d} \hat{t}_j X^j + \sum_{j=0}^{n} \hat{t}'_j \hat{P}_j(X)\right) \cdot \hat{P}_0(X) + \sum_{i=1}^{n} \left(\sum_{j=0}^{d} s_{ij} X^j + \sum_{j=0}^{n} s'_{ij} \hat{P}_j(X)\right) \hat{P}_i(X)$$

satisfy $d_1(\chi) = d_2(\chi) = 0$. Since the adversary is oblivious to the actual representation of the group elements it will do the same group operations no matter the actual value of $X (= \chi)$; so the values $t_j, \ldots, s'_{ij}$ are generated (almost[2]) independently of χ. By the Schwartz-Zippel lemma there is a negligible probability that $d_i(\chi) = 0$, for non-zero $d_i(X)$, when we choose χ randomly. Thus, with all but a negligible probability $d_1(X)$ and $d_2(X)$ are zero polynomials.

Since $\mathcal{F}$ and $\{X^i\}_{i=0}^{2d} \cup \{X^i \cdot \hat{P}_j(X)\}_{i \in [0\,..\,d], j \in [0\,..\,n]}$ are both linearly independent, $\{X^i\}_{i=0}^{2d} \cup \{P_i(X)\hat{P}_j(X)\}_{i,j \in [0\,..\,n]}$ is also linearly independent. We get from $d_1(X) = 0$ that $\sum_{j=0}^{n} t'_j P_0(X)\hat{P}_j(X) + \sum_{i=1}^{n}\sum_{j=0}^{n} s'_{ij} P_i(X)\hat{P}_j(X) = 0$, which implies $s'_{ij} = 0$ for $i \in [1\,..\,n], j \in [0\,..\,n]$. Substituting these values into $d_2(X) = 0$, we get that $\left(\sum_{j=0}^{d} \hat{t}_j X^j + \sum_{j=0}^{n} \hat{t}'_j \hat{P}_j(X)\right) \hat{P}_0(X) + \sum_{i=1}^{n}\sum_{j=0}^{d} s_{ij} X^j \hat{P}_i(X) = 0$. Since $\hat{\mathcal{F}}_d$ is linearly independent, we get that all coefficients in the above equation are zero, and in particular $s_{ij} = 0$ for $i \in [1\,..\,n], j \in [0\,..\,n]$. Thus $s_i = 1$ for $i \in [1\,..\,n]$. Contradiction to the fact that the adversary is successful. □

6 New Shuffle Argument

Let Elgamal operate in $\mathbb{G}_1$ defined by gk. In a shuffle argument, the prover aims to convince the verifier that, given the description of a group, a public key, and two vectors of ciphertexts, the second vector of the ciphertexts is a permutation of rerandomized versions of the ciphertexts from the first vector. However, to achieve better efficiency, we construct a shuffle argument that is only culpably sound with respect to the next relation (i.e., $\mathcal{R}_{sh}^{\mathsf{guilt}}$-sound, see App. J):

$$\mathcal{R}_{sh,n}^{\mathsf{guilt}} = \left\{ \begin{array}{l} (\mathsf{gk}, (\mathsf{pk}, (z_i)_{i=1}^{n}, (z'_i)_{i=1}^{n}), \mathsf{sk}) : \mathsf{gk} \in \mathsf{BP}(1^\kappa, n) \wedge \\ (\mathsf{pk}, \mathsf{sk}) \in \mathsf{genpkc}(\mathsf{gk}) \wedge \left(\forall \psi \in S_n : \exists i : \mathsf{dec}_{\mathsf{sk}}(z'_i) \neq \mathsf{dec}_{\mathsf{sk}}(z_{\psi(i)})\right) \end{array} \right\} \ .$$

The argument of [GL07] is proven to be $\mathcal{R}_{sh}^{\mathsf{guilt}}$-sound with respect to the same relation. See [GL07] or the introduction for an explanation why $\mathcal{R}_{sh}^{\mathsf{guilt}}$ is sufficient.

As noted in the introduction, we need to use same-message arguments and rely on the PSP assumption. Thus, we need polynomials $\hat{P}_j$ that satisfy two different requirements at once. First, to be able to use the same-message argument, we need that $y_j(\omega_k) = \hat{P}_j(\omega_k)$ for $k \in [1\,..\,n+1]$. Second, to be able to use the PSP assumption, we need $(d, \hat{\mathcal{F}})$ to be PSP-friendly, and for this we need $\hat{P}_j(X)$ to have a sufficiently large degree. Recall that y_j are fixed by the unit vector argument. We now show that such a choice for $\hat{P}_j$ exists. (See App. E for a proof.)

Proposition 3. *Let $\hat{y}_j(X) := (XZ(X)+1)^{j-1}(X^2 Z(X)+1) y_j(X)$ for $j \in [1\,..\,n]$, and $\hat{Z}(X) = \hat{y}_0(X) := (XZ(X)+1)^{n+1} Z(X)$. Let $\hat{\mathcal{F}}_{\mathsf{com}} = (\hat{y}_j(X))_{j=0}^{n}$. Then $\hat{y}_j(\omega_k) = y_j(\omega_k)$ for all j, k, and $(n+1, \hat{\mathcal{F}}_{\mathsf{com}})$ is PSP-friendly.*

Next, we will provide the full description of the new shuffle argument. Note that $(c_i)_{i=1}^{n}$ are commitments to the rows of the permutation matrix $\boldsymbol{\Psi}$, proven by the n unit vector arguments $(\pi_{uv,i})_{i=1}^{n}$ and by the implicit computation of c_n. We denote $\hat{E}((a,b),c) := (\hat{e}(a,c), \hat{e}(b,c))$.

[2] A generic bilinear group adversary may learn a negligible amount of information about χ by comparing group elements; we skip this part in the proof.

System parameters: Let $(\mathsf{genpkc}, \mathsf{enc}, \mathsf{dec})$ be the Elgamal cryptosystem. Let com be the polynomial commitment scheme. Consider polynomials $\mathcal{F}_{\mathsf{com}} = \{Z(X)\} \cup (y_i(X))_{i=1}^n$ from Sect. 3. Let $\hat{\mathcal{F}}_{\mathsf{com}} = (\hat{y}_i(X))_{i=0}^n$ be as in Prop. 3.

Setup $\mathsf{setup}_{sh}(1^\kappa, n)$**:** Let $\mathsf{gk} \leftarrow \mathsf{BP}(1^\kappa, n)$.

CRS generation $\mathsf{gencrs}_{sh}(\mathsf{gk})$**:** Let $(g_1, g_2, \chi, \beta, \gamma) \leftarrow_r \mathbb{G}_1^* \times \mathbb{G}_2^* \times \mathbb{Z}_p^3$ with $Z(\chi) \neq 0$. Let $(\mathsf{crs}_{uv,p}, \mathsf{crs}_{uv,v}) \leftarrow_r \mathsf{gencrs}_{uv}(\mathsf{gk}, n)$, $(\mathsf{crs}_{sm,p}, \mathsf{crs}_{sm,v}) \leftarrow_r \mathsf{gencrs}_{sm}(\mathsf{gk}, n)$, but by using the same $(g_1, g_2, \chi, \beta, \gamma)$ in both cases. Let $\mathsf{ck} \leftarrow (g_1, g_2^\gamma)^{\mathcal{F}_{\mathsf{com}}(\chi)}$ and $\widehat{\mathsf{ck}} \leftarrow (g_1, g_2^{\hat{\gamma}})^{\hat{\mathcal{F}}_{\mathsf{com}}(\chi)}$. Set $(D_1, D_2^\gamma) \leftarrow \mathsf{com}(\mathsf{ck}; \mathbf{1}_{\boldsymbol{n}}; 0)$, $(\hat{D}_1, \hat{D}_2^{\hat{\gamma}}) \leftarrow \mathsf{com}(\widehat{\mathsf{ck}}; \mathbf{1}_{\boldsymbol{n}}; 0)$. Set $\mathsf{crs}_{sh,p} \leftarrow (\mathsf{crs}_{uv,p}, \widehat{\mathsf{ck}}, g_1^{\hat{Z}(\chi)/Z(\chi)}, g_1, (g_1^{(\hat{y}_i(\chi) - y_i(\chi))/Z(\chi)})_{i=1}^n, D_1, D_2^\gamma, \hat{D}_1, \hat{D}_2^{\hat{\gamma}})$, $\mathsf{crs}_{sh,v} \leftarrow (\mathsf{crs}_{uv,v}, g_2^{\hat{\gamma}}, \{g_2^{\gamma y_i(\chi)}, g_2^{\hat{\gamma}\hat{y}_i(\chi)}\}_{i=0}^n, D_1, D_2^\gamma, \hat{D}_1, \hat{D}_2^{\hat{\gamma}})$, and $\mathsf{td}_{sh} \leftarrow \chi$. Return $((\mathsf{crs}_{sh,p}, \mathsf{crs}_{sh,v}), \mathsf{td}_{sh})$.

Common input: $(\mathsf{pk}, (z_i, z'_i)_{i=1}^n)$, where $\mathsf{pk} = (g_1, h) \in \mathbb{G}_1^2$, $z_i \in \mathbb{G}_1^2$ and $z'_i = z_{\psi(i)} \cdot \mathsf{enc}_{\mathsf{pk}}(1; t_i) \in \mathbb{G}_1^2$.

Argument $\mathsf{pro}_{sh}(\mathsf{gk}, \mathsf{crs}_{sh,p}; \mathsf{pk}, (z_i, z'_i)_{i=1}^n; \psi, (t_i)_{i=1}^n)$**:**

```
(1) Let $\boldsymbol{\Psi} = \boldsymbol{\Psi}_{\psi^{-1}}$ be the $n \times n$ permutation matrix corresponding to $\psi^{-1}$.
(2) For $i \in [1 \,..\, n-1]$:
    – Set $r_i \leftarrow \mathbb{Z}_p$, $(c_{i1}, c_{i2}^\gamma) \leftarrow \mathsf{com}(\mathsf{ck}; \boldsymbol{\Psi}_i; r_i)$, $(\hat{c}_{i1}, \hat{c}_{i2}^{\hat{\gamma}}) \leftarrow \mathsf{com}(\widehat{\mathsf{ck}}; \boldsymbol{\Psi}_i; r_i)$.
(3) Set $r_n \leftarrow -\sum_{i=1}^{n-1} r_i$, $(c_{n1}, c_{n2}^\gamma) \leftarrow (D_1, D_2^\gamma) / \prod_{i=1}^{n-1} (c_{i1}, c_{i2}^\gamma)$.
(4) Set $(\hat{c}_{n1}, \hat{c}_{n2}^{\hat{\gamma}}) \leftarrow (\hat{D}_1, \hat{D}_2^{\hat{\gamma}}) / \prod_{i=1}^{n-1} (\hat{c}_{i1}, \hat{c}_{i2}^{\hat{\gamma}})$.
(5) For $i \in [1 \,..\, n]$: set $\pi_{uv,i} = (\pi^*_{uv,i}, c_{i1}^\beta) \leftarrow \mathsf{pro}_{uv}(\mathsf{gk}, \mathsf{crs}_{uv,p}; c_{i1}, c_{i2}^\gamma; \boldsymbol{\Psi}_i, r_i)$.
(6) Set $r_t \leftarrow_r \mathbb{Z}_p$, $(d_1, d_2^\gamma) \leftarrow \mathsf{com}(\mathsf{ck}; \boldsymbol{t}; r_t)$, and $(\hat{d}_1, \hat{d}_2^{\hat{\gamma}}) \leftarrow \mathsf{com}(\widehat{\mathsf{ck}}; \boldsymbol{t}; r_t)$.
(7) For $i \in [1 \,..\, n-1]$:
    – Set $(\pi^*_{sm,i}, c_{i1}^\beta) \leftarrow \mathsf{pro}_{sm}(\mathsf{gk}, \mathsf{crs}_{sm,p}; c_{i1}, c_{i2}^\gamma, \hat{c}_{i1}, \hat{c}_{i2}^{\hat{\gamma}}; \boldsymbol{\Psi}_i, r_i, r_i)$.
(8) Set $\pi_{sm,d} \leftarrow \mathsf{pro}_{sm}(\mathsf{gk}, \mathsf{crs}_{sm,p}; d_1, d_2^\gamma, \hat{d}_1, \hat{d}_2^{\hat{\gamma}}; \boldsymbol{t}, r_t, r_t)$.
(9) Compute $U = (U_1, U_2) \leftarrow \mathsf{pk}^{r_t} \cdot \prod_{i=1}^n z_i^{r_i} \in \mathbb{G}_1^2$. // The only online step
(10) Output $\pi_{sh} \leftarrow ((c_{i1}, c_{i2}^\gamma, \hat{c}_{i1}, \hat{c}_{i2}^{\hat{\gamma}})_{i=1}^{n-1}, d_1, d_2^\gamma, \hat{d}_1, \hat{d}_2^{\hat{\gamma}}, (\pi_{uv,i})_{i=1}^n, (\pi^*_{sm,i})_{i=1}^{n-1}, \pi_{sm,d}, U)$
```

Verification $\mathsf{ver}_{sh}(\mathsf{gk}, \mathsf{crs}_{sh,v}; \mathsf{pk}, (z_i, z'_i)_{i=1}^n, \pi_{sh})$**:**

```
(1) Let $(c_{n1}, c_{n2}^\gamma) \leftarrow (D_1, D_2^\gamma) / \prod_{i=1}^{n-1} (c_{i1}, c_{i2}^\gamma)$.
(2) Let $(\hat{c}_{n1}, \hat{c}_{n2}^{\hat{\gamma}}) \leftarrow (\hat{D}_1, \hat{D}_2^{\hat{\gamma}}) / \prod_{i=1}^{n-1} (\hat{c}_{i1}, \hat{c}_{i2}^{\hat{\gamma}})$.
(3) For $i \in [1 \,..\, n]$: reject if $\mathsf{ver}_{uv}(\mathsf{gk}, \mathsf{crs}_{uv,v}; c_{i1}, c_{i2}^\gamma; \pi_{uv,i})$ rejects.
(4) For $i \in [1 \,..\, n-1]$: reject if $\mathsf{ver}_{sm}(\mathsf{gk}; \mathsf{crs}_{sm,v}; c_{i1}, c_{i2}^\gamma, \hat{c}_{i1}, \hat{c}_{i2}^{\hat{\gamma}}; \pi_{sm,i})$ rejects.
(5) Reject if $\mathsf{ver}_{sm}(\mathsf{gk}, \mathsf{crs}_{sm,v}; d_1, d_2^\gamma, \hat{d}_1, \hat{d}_2^{\hat{\gamma}}; \pi_{sm,d})$ rejects.
(6) Check the PSP-related verification equations: // The only online step
    (a) $\prod_{i=1}^n \hat{E}(z'_i, g_2^{\gamma y_i(\chi)}) / \prod_{i=1}^n \hat{E}(z_i, c_{i2}^\gamma) = \hat{E}((g_1, h), d_2^\gamma) / \hat{E}(U, g_2^{\gamma Z(\chi)})$,
    (b) $\prod_{i=1}^n \hat{E}(z'_i, g_2^{\hat{\gamma}\hat{y}_i(\chi)}) / \prod_{i=1}^n \hat{E}(z_i, \hat{c}_{i2}^{\hat{\gamma}}) = \hat{E}((g_1, h), \hat{d}_2^{\hat{\gamma}}) / \hat{E}(U, g_2^{\hat{\gamma}\hat{Z}(\chi)})$.
```

Since $\mathsf{ck}, \widehat{\mathsf{ck}} \subset \mathsf{crs}_{sh,p}$, $(D_1, D_2^\gamma) = \mathsf{com}(\mathsf{ck}; \mathbf{1}_n; 0)$ and $(\hat{D}_1, \hat{D}_2^{\hat{\gamma}}) = \mathsf{com}(\widehat{\mathsf{ck}}; \mathbf{1}_n; 0)$ can be computed from the rest of the CRS. (These four elements are only needed to optimize the computation of (c_{n1}, c_{n2}^γ) and $(\hat{c}_{n1}, \hat{c}_{n2}^{\hat{\gamma}})$.) For security, it suffices to take $\mathsf{crs}^*_{sh} = (g_1^{\mathcal{F}_{sh,1}(\chi,\beta)}, g_2^{\mathcal{F}_{sh,2}(\chi,\beta,\gamma,\hat{\gamma})})$, where $\mathcal{F}_{sh,1} = \mathcal{F}_{uv,1} \cup \hat{\mathcal{F}}_{\mathsf{com}} \cup \{\hat{Z}(X)/Z(X)\} \cup \{(\hat{y}_i(X) - y_i(X))/Z(X)\}_{i=1}^n$ and $\mathcal{F}_{sh,2} = \mathcal{F}_{uv,2} \cup \hat{Y} \cdot (\{1\} \cup \hat{\mathcal{F}}_{\mathsf{com}})$.

Theorem 5. *The new shuffle argument is a non-interactive perfectly complete and perfectly zero-knowledge shuffle argument for Elgamal ciphertexts. If the $(n+1)$-TSDH, $(\hat{n}, \hat{n}+n+2)$-PCDH, $(\mathcal{F}_{\mathsf{com}}, \hat{\mathcal{F}}_{\mathsf{com}})$-PSP, $(n+1, \mathcal{F}_{sh,1} \setminus (\{1\} \cup \mathcal{F}_{\mathsf{com}}), \mathcal{F}_{sh,2} \setminus Y \cdot (\{1\} \cup \mathcal{F}_{\mathsf{com}}), \gamma)$-PKE, $(\hat{\mathcal{F}}_{\mathsf{com}}, \mathcal{F}_{sh,1} \setminus \hat{\mathcal{F}}_{\mathsf{com}}, \mathcal{F}_{sh,2} \setminus \hat{Y}\hat{\mathcal{F}}_{\mathsf{com}}, \hat{\gamma})$-PKE assumptions hold, then the shuffle argument is adaptively computationally culpably sound w.r.t. the language $\mathcal{R}^{\mathsf{guilt}}_{sh,n}$ and an argument of knowledge.*

A full proof of this theorem is given in App. F. When using a Barreto-Naehrig curve [BN05], exponentiations in $\mathbb{G}_1$ are three times cheaper than in $\mathbb{G}_2$. Moreover, a single $(N+1)$-wide multi-exponentiations is considerably cheaper than $N+1$ exponentiations. Hence, we compute separately the number of exponentiations and multi-exponentiations in both $\mathbb{G}_1$ and $\mathbb{G}_2$ [Str64,Pip80]. For the sake of the simplicity, Prop. 4 only summarizes those numbers. See App. G for a proof.

Proposition 4. *The prover's CRS consists of $6n + 7$ elements of $\mathbb{G}_1$ and $2n + 4$ elements of $\mathbb{G}_2$. The verifier's CRS consists of 4 elements of $\mathbb{G}_1$, $2n + 8$ elements of $\mathbb{G}_2$, and 1 element of $\mathbb{G}_T$. The total CRS is $6n + 8$ elements of $\mathbb{G}_1$, $2n + 8$ elements of $\mathbb{G}_2$, and 1 element of $\mathbb{G}_T$, in total $8n + 17$ group elements. The communication complexity is $5n + 2$ elements of $\mathbb{G}_1$ and $2n$ elements of $\mathbb{G}_2$, in total $7n + 2$ group elements. The prover's and the verifier's computational complexity are as in Tbl. 1.*

Importantly, both the proving and verification algorithm of the new shuffle argument can be divided into offline (independent of the common input $(\mathsf{pk}, (z_i, z'_i)_{i=1}^n)$) and online (dependent on the common input) parts. The prover can precompute all elements of π_{sh} except U (i.e., execute all steps of the proving algorithm, except step (9)), and send them to the verifier before the inputs are fixed. The verifier can verify $\pi_{sh} \setminus \{U\}$ (i.e., execute all steps of the verification algorithm, except step (6)) in the precomputation step. Thus, the online computational complexity is dominated by two $(n+1)$-wide multi-exponentiations for the prover, and $8n + 4$ pairings for the verifier (note that $\hat{E}((g_1, h), d_2^{\gamma})$ and $\hat{E}((g_1, h), \hat{d}_2^{\hat{\gamma}})$ can also be precomputed by the verifier).

Low online complexity is highly important in e-voting, where the online time (i.e., the time interval after the ballots are gathered and before the election results are announced) can be limited for legal reasons. In this case, the mix servers can execute all but step (9) of the proving algorithm and step (6) of the verification algorithm before the votes are even cast, assuming one is able to set a priori a reasonable upper bound on n, the number of votes. See [Wik09] for additional motivation.

Acknowledgments. The authors were supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 653497 (project PANORAMIX), and the Estonian Research Council.

References

AF07. Masayuki Abe and Serge Fehr. Perfect NIZK with Adaptive Soundness. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 118–136, Amsterdam, The Netherlands, February 21–24, 2007. Springer, Heidelberg.

BB04. Dan Boneh and Xavier Boyen. Secure Identity Based Encryption Without Random Oracles. In Matthew K. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 443–459, Santa Barbara, USA, August 15–19, 2004. Springer, Heidelberg.

BBS04. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short Group Signatures. In Matthew K. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55, Santa Barbara, USA, August 15–19, 2004. Springer, Heidelberg.

BCPR14. Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. On the Existence of Extractable One-Way Functions. In David Shmoys, editor, *STOC 2014*, pages 505–514, New York, NY, USA, May 31 – Jun 3, 2014. ACM Press.

BFM88. Manuel Blum, Paul Feldman, and Silvio Micali. Non-Interactive Zero-Knowledge and Its Applications. In *STOC 1988*, pages 103–112, Chicago, Illinois, USA, May 2–4, 1988. ACM Press.

BN05. Paulo S. L. M. Barreto and Michael Naehrig. Pairing-Friendly Elliptic Curves of Prime Order. In Bart Preneel and Stafford E. Tavares, editors, *SAC 2005*, volume 3897 of *LNCS*, pages 319–331, Kingston, ON, Canada, August 11–12, 2005. Springer, Heidelberg.

CGH98. Ran Canetti, Oded Goldreich, and Shai Halevi. The Random Oracle Methodology, Revisited. In Jeffrey Scott Vitter, editor, *STOC 1998*, pages 209–218, Dallas, Texas, USA, May 23–26, 1998.

Cha81. David Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

CLZ12. Rafik Chaabouni, Helger Lipmaa, and Bingsheng Zhang. A Non-Interactive Range Proof with Constant Communication. In Angelos Keromytis, editor, *FC 2012*, volume 7397 of *LNCS*, pages 179–199, Bonaire, The Netherlands, Feb 27–Mar 2, 2012. Springer, Heidelberg.

Dam91. Ivan Damgård. Towards Practical Public Key Systems Secure against Chosen Ciphertext Attacks. In Joan Feigenbaum, editor, *CRYPTO 1991*, volume 576 of *LNCS*, pages 445–456, Santa Barbara, California, USA, August 11–15, 1991. Springer, Heidelberg, 1992.

DDO$^+$01. Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust Non-interactive Zero Knowledge. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 566–598, Santa Barbara, USA, August 19–23, 2001. Springer, Heidelberg.

DFGK14. George Danezis, Cédric Fournet, Jens Groth, and Markulf Kohlweiss. Square Span Programs with Applications to Succinct NIZK Arguments. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014 (1)*, volume 8873 of *LNCS*, pages 532–550, Kaohsiung, Taiwan, R.O.C., December 7–11, 2014. Springer, Heidelberg.

Elg85. Taher Elgamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Trans. on Inf. Theory*, 31(4):469–472, 1985.

FL16. Prastudy Fauzi and Helger Lipmaa. Efficient Culpably Sound NIZK Shuffle Argument without Random Oracles. In Kazue Sako, editor, *CT-RSA 2016*, volume ? of *LNCS*, pages ?–?, San Franscisco, CA, USA, February 29–March 4, 2016. Springer, Heildeberg.

FLZ13. Prastudy Fauzi, Helger Lipmaa, and Bingsheng Zhang. Efficient Modular NIZK Arguments from Shift and Product. In Michel Abdalla, Cristina Nita-Rotaru, and Ricardo Dahab, editors, *CANS 2013*, volume 8257 of *LNCS*, pages 92–121, Paraty, Brazil, November 20–22, 2013. Springer, Heidelberg.

FS86. Amos Fiat and Adi Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In Andrew M. Odlyzko, editor, *CRYPTO 1986*, volume 263 of *LNCS*, pages 186–194, Santa Barbara, California, USA, 11–15 August 1986. Springer, Heidelberg, 1987.

FS01. Jun Furukawa and Kazue Sako. An Efficient Scheme for Proving a Shuffle. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 368–387, Santa Barbara, USA, August 19–23, 2001. Springer, Heidelberg.

Fur05. Jun Furukawa. Efficient and Verifiable Shuffling and Shuffle-Decryption. *IEICE Transactions*, 88-A(1):172–188, 2005.

GGPR13. Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic Span Programs and NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645, Athens, Greece, April 26–30, 2013. Springer, Heidelberg.

GJM02. Philippe Golle, Stanislaw Jarecki, and Ilya Mironov. Cryptographic Primitives Enforcing Communication and Storage Complexity. In Matt Blaze, editor, *FC 2002*, volume 2357 of *LNCS*, pages 120–135, Southhampton Beach, Bermuda, March 11–14, 2002. Springer, Heidelberg.

GK03. Shafi Goldwasser and Yael Tauman Kalai. On the (In)security of the Fiat-Shamir Paradigm. In *FOCS 2003*, pages 102–113, Cambridge, MA, USA, October 11–14, 2003. IEEE, IEEE Computer Society Press.

GL07. Jens Groth and Steve Lu. A Non-interactive Shuffle with Pairing Based Verifiability. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 51–67, Kuching, Malaysia, December 2–6, 2007. Springer, Heidelberg.

GOS12. Jens Groth, Rafail Ostrovsky, and Amit Sahai. New Techniques for Noninteractive Zero-Knowledge. *Journal of the ACM*, 59(3), 2012.

Gro10a. Jens Groth. A Verifiable Secret Shuffle of Homomorphic Encryptions. *J. Cryptology*, 23(4):546–579, 2010.

Gro10b. Jens Groth. Short Pairing-Based Non-interactive Zero-Knowledge Arguments. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340, Singapore, December 5–9, 2010. Springer, Heidelberg.

KMW12. Shahram Khazaei, Tal Moran, and Douglas Wikström. A Mix-Net from Any CCA2 Secure Cryptosystem. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 607–625, Beijing, China, December 2–6, 2012. Springer, Heidelberg.

Lip12. Helger Lipmaa. Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 169–189, Taormina, Italy, March 18–21, 2012. Springer, Heidelberg.

Lip14. Helger Lipmaa. Prover-Efficient Commit-And-Prove Zero-Knowledge SNARKs. Technical Report 2014/396, IACR, May 30, 2014. Available at `http://eprint.iacr.org/2014/396`, updated on Nov 16, 2015.

LZ12. Helger Lipmaa and Bingsheng Zhang. A More Efficient Computationally Sound Non-Interactive Zero-Knowledge Shuffle Argument. In Ivan Visconti and Roberto De Prisco, editors, *SCN 2012*, volume 7485 of *LNCS*, pages 477–502, Amalfi, Italy, September 5–7, 2012. Springer, Heidelberg.

LZ13. Helger Lipmaa and Bingsheng Zhang. A More Efficient Computationally Sound Non-Interactive Zero-Knowledge Shuffle Argument. *Journal of Computer Security*, 21(5):685–719, 2013.

Nef01. C. Andrew Neff. A Verifiable Secret Shuffle and Its Application to E-Voting. In *ACM CCS 2001*, pages 116–125, Philadelphia, Pennsylvania, USA, November 6–8 2001. ACM Press.

PGHR13. Bryan Parno, Craig Gentry, Jon Howell, and Mariana Raykova. Pinocchio: Nearly Practical Verifiable Computation. In *IEEE SP 2013*, pages 238–252, Berkeley, CA, USA, May 19-22, 2013. IEEE Computer Society.

PIK93. Choonsik Park, Kazutomo Itoh, and Kaoru Kurosawa. Efficient Anonymous Channel and All/Nothing Election Scheme. In Tor Helleseth, editor, *EUROCRYPT 1993*, volume 765 of *LNCS*, pages 248–259, Lofthus, Norway, May 23–27, 1993. Springer, Heidelberg, 1994.

Pip80. Nicholas Pippenger. On the Evaluation of Powers and Monomials. *SIAM J. Comput.*, 9(2):230–250, 1980.

RKP09. Alfredo Rial, Markulf Kohlweiss, and Bart Preneel. Universally Composable Adaptive Priced Oblivious Transfer. In Hovav Shacham and Brent Waters, editors, *Pairing 2009*, volume 5671 of *LNCS*, pages 231–247, Palo Alto, CA, USA, August 12–14, 2009. Springer, Heidelberg.

SK95. Kazue Sako and Joe Kilian. Receipt-Free Mix-Type Voting Scheme - A Practical Solution to the Implementation of a Voting Booth. In Louis C. Guillou and Jean-Jacques Quisquater, editors, *EUROCRYPT 1995*, volume 921 of *LNCS*, pages 393–403, Saint-Malo, France, 21–25 May 1995. Springer, Heidelberg.

Str64. Ernst G. Straus. Addition Chains of Vectors. *Amer. Math. Monthly*, 70:806–808, 1964.

TW10. Björn Terelius and Douglas Wikström. Proofs of Restricted Shuffles. In Daniel J. Bernstein and Tanja Lange, editors, *AFRICACRYPT 2010*, volume 6055 of *LNCS*, pages 100–113, Stellenbosch, South Africa, May 3–6, 2010. Springer, Heidelberg.

Wik09. Douglas Wikström. A Commitment-Consistent Proof of a Shuffle. In Colin Boyd and Juan Manuel González Nieto, editors, *ACISP 2009*, volume 5594 of *LNCS*, pages 4007–421, Brisbane, Australia, July 1–3, 2009. Springer, Heidelberg.

A On Random Oracle Versus CRS Model

NIZK arguments for non-trivial languages require the use of either the random oracle (RO, [FS86]) model or the common reference string (CRS, [BFM88]) model. In the random oracle model, it is assumed that all parties have an oracle access to a uniformly random function. In practice, this is instantiated with say a secure hash function [FS86]. Many shuffle arguments have been proposed in the random oracle model, see, e.g., [Nef01,FS01,Fur05,TW10,Gro10a]. However, it is well known that the random oracle model is not always instantiable [CGH98,GK03]. Thus, one should aim to provide a security proof in the in the (random oracle-less) CRS model. In the CRS model, all parties have access to an honestly generated CRS. In addition, the simulator will have access to a trapdoor. The CRS model is much more realistic than the RO model, especially since the common CRS can be generated by using either multi-party computation or secure hardware.

B Proof of Thm. 1 (Security of Commitment Scheme)

Proof. Perfect Hiding: since $P_0(X)$ is a non-zero polynomial (this follows from linear independence), then due to the choice of χ, $rP_0(\chi)$ (and thus also $\log_{g_1} A_1$) is uniformly random in $\mathbb{Z}_p$. Thus, (A_1, A_2^γ) is a uniformly random element of the multiplicative subgroup of $\mathbb{G}_1^* \times \mathbb{G}_2^*$ generated by (g_1, g_2^γ), independently of the committed value.

Extractability: clear from the statement.

Computational Binding: assume that the adversary outputs $(\boldsymbol{a}, r_a)$ and $(\boldsymbol{b}, r_b)$ with $(\boldsymbol{a}, r_a) \neq (\boldsymbol{b}, r_b)$, such that $d(X) := (r_a P_0(X) + \sum_{i=1}^n a_i P_i(X)) - (r_b P_0(X) + \sum_{i=1}^n b_i P_i(X))$ has a root at χ. If the adversary is successful, then $d(X) \in \mathbb{Z}_p[X]$ is a non-trivial polynomial. Since the coefficients of $d(X)$ are known, we can use an efficient polynomial factorization algorithm to compute all roots r_i of $d(X)$. Since one of these roots has to be equal to χ, the adversary can just output one of the r_i randomly. (We note that in previous papers like [Gro10b], one instead compared g_1^χ (given in the CRS) to all values $g_1^{r_i}$. In our case, it is not guaranteed that both g_1 and g_1^χ belong to the CRS, and thus for simplicity, we have this randomized step.)

Trapdoor: given χ, $\boldsymbol{a}$, r, $\boldsymbol{a^*}$, and $c = \mathsf{com}(\mathsf{ck}; \boldsymbol{a}; r)$, we compute r^* such that $(r^* - r)P_0(\chi) + \sum_{i=1}^n (a_i^* - a_i)P_i(\chi) = 0$. This is possible since $P_0(\chi) \neq 0$. Clearly, $c = \mathsf{com}(\mathsf{ck}; \boldsymbol{a^*}; r^*)$. □

C Proof of Thm. 2 (Security of Unit Vector)

Proof. Perfect completeness follows from the above derivation. Perfect witness indistinguishability is due to the fact that there is a unique value of π_{uv} that satisfies the verification equations.

Argument of knowledge: Assume that the PKE assumption holds. According to the preceding discussion, in this case, the argument of knowledge property means that any NUPPT adversary has negligible chance of outputting an input $u_{uv} \leftarrow (A_1, A_2^\gamma)$, an accepting argument $\pi_{uv} = (\pi^*_{uv}, A_1^\beta)$, and a witness $w_{uv} = \boldsymbol{a'} \in \mathbb{Z}_p^{n+2}$, such that

1. $(A_1, A_2^\gamma) = (g_1, g_2^\gamma)^{A(\chi)}$ with $A(X) = \sum_{i=0}^{n+1} a'_i X^i$,
2. One of the following holds:
   (a) $A(X) \not\in \operatorname{span}(\mathcal{F}_{\mathsf{com}})$, where $\mathcal{F}_{\mathsf{com}} = (Z(X), (y_i(X))_{i=1}^n)$, or
   (b) $Z(X) \nmid Q_{wi}(X)$ where $Q_{wi}(X) = (A(X) + y_0(X))^2 - 1$.

Suppose there exists an adversary A_{aok} that breaks the argument of knowledge property. Then given a correctly generated CRS, he can output $(u_{uv}, \pi_{uv}, w_{uv})$ such that both conditions 1 and 2 hold. By the PKE assumption, condition 1 holds (it will also be guaranteed in the master argument by a similar knowledge assumption). We will handle separately the cases when the conditions 2a and 2b hold. In each case we derive a contradiction to one security assumption.

Condition 2a holds: assume that there exists an adversary A_{aok} that, given gk and $(\mathsf{crs}_{uv,p}, \mathsf{crs}_{uv,v})$ output by $\mathsf{gencrs}_{uv}(\mathsf{gk})$, breaks the argument of knowledge property of the unit vector argument (with some non-negligible probability ε) by outputting $A(X)$ such that $A(X) \in \operatorname{span}(\{X^i\}_{i=0}^{n+1}) \setminus \operatorname{span}(\mathcal{F}_{\mathsf{com}})$. We construct the following $(n+1, 2n+3)$-PCDH adversary A_{pcdh}. A_{pcdh} receives an $(n+1, 2n+3)$-PCDH challenge $ch = (\mathsf{gk}, ((g_1, g_2)^{\chi^i})_{i \in [0\,..\,2n+3] \setminus \{n+2\}})$. Let $\mathcal{D}$ be the set of polynomials $q(X)$ from $\mathbb{Z}_p[X]$, such that $\deg q \le n+1$ and $q(X) f(X)$ has a zero coefficient for X^{n+1} for all $f \in \mathcal{F}_{\mathsf{com}}$.

A_{pcdh} picks $q(X)$ randomly from $\mathcal{D}$. Note that $\deg(q(X)f(X)) \le (n+1) + (n+1) = 2(n+1)$ for any $f(X) \in \mathcal{F}_{\mathsf{com}}$. There are $(n+2) - |\mathcal{F}_{\mathsf{com}}| = 1 > 0$ degrees of freedom for choosing $q(X)$. Thus, for a polynomial $\pi(X)$ outside of the span of $\mathcal{F}_{\mathsf{com}}$, the coefficient of X^{n+1} in $q(X)\pi(X)$ will be random.

A_{pcdh} then picks $b \leftarrow_r \mathbb{Z}_p$, sets $\beta(X) \leftarrow X q(X) + b$ and $\beta \leftarrow \beta(\chi)$. Since $\mathcal{D}$ consists of polynomials of degree at most $n+1$, then for $f \in \mathcal{F}_{\mathsf{com}}$, $\beta(X) f(X)$ is of degree at most $(n+2) + (n+1) = 2n+3$, with a zero coefficient for X^{n+2}. This means that A_{pcdh} can compute $g_1^{\beta f(\chi)}$ from ch by using generic bilinear group operations.

With the given values, A_{pcdh} generates $\gamma \leftarrow_r \mathbb{Z}_p$, and computes a correct CRS which is sent to A_{aok}. Suppose that A_{aok} replies with $(u_{uv}, \pi_{uv}, w_{uv})$ such that $A(X) = \sum_{i=0}^{n+1} a'_i X^i$ is not in the span of $\mathcal{F}_{\mathsf{com}}$, verification succeeds, and $A_1 = g_1^{A(\chi)}$. Since $A(X)$ is not in the span of $\mathcal{F}_{\mathsf{com}}$, the coefficient of X^{n+1} in $q(X)A(X)$ is random. This means that with probability $1 - 1/p$, the coefficient c of X^{n+2} in the known polynomial $\beta(X)A(X)$ is non-zero. Since $\hat{e}(A_1, g_2^{\gamma\beta}) = \hat{e}(A_1^\beta, g_2^\gamma)$, A_{pcdh} can compute $A_1^\beta = g_1^{\beta A(\chi)}$. However, A_{pcdh} knows all the coefficients of $\beta(X)A(X)$, and hence from ch she can compute $g_1^{\chi^{n+2}} = (g_1^{c\chi^{n+2}})^{c^{-1}}$. Thus, A_{pcdh} solves the $(n+1, 2n+3)$-PCDH problem with non-negligible probability $(1 - 1/p) \cdot \varepsilon$.

Condition 2b holds: assume that there exists an adversary A_{aok} that, given gk and $(\mathsf{crs}_{uv,p}, \mathsf{crs}_{uv,v})$ output by $\mathsf{gencrs}_{uv}(\mathsf{gk})$, breaks the argument of knowledge property of the unit vector argument by outputting $A(X)$ such that $Z(X) \nmid Q_{wi}(X)$ where $Q_{wi}(X)$ is defined as in condition 2b. In this case we construct a TSDH adversary A_{tsdh}. Assume that A_{tsdh} gets as an input a TSDH challenge $ch = ((g_1, g_2)^{\chi^i})_{i=0}^{n+1}$. Then, A_{tsdh} generates random γ, β, and uses them together with ch to generate a correct CRS for A_{aok}. Assume that (with some probability ε) A_{aok} then outputs u_{uv}, an accepting argument π_{uv}, and a witness w_{uv}.

Since $Z(X) = \prod_{i=1}^{n+1}(X - \omega_i)$ and all $X - \omega_i$ are pairwise relatively prime, $Z(X) \nmid Q_{wi}(X)$ means there exists $i \in [1\,..\,n+1]$ such that $(X - \omega_i)$ does not divide $Q_{wi}(X)$. Thus, there exists a non-zero constant $t \in \mathbb{Z}_p$ and a degree $2n+1$ polynomial $q(X)$ such that $Q_{wi}(X) = (X - \omega_i) q(X) + t$.

Since $Q_{wi}(X) = (A(X) + y_0(X))^2 - 1$ and the first verification equation accepts,

$$\hat{e}(\pi^*_{uv}, g_2^{\gamma Z(\chi)}) = \hat{e}(A_1 \cdot g_1^{y_0(\chi)}, A_2^\gamma \cdot g_2^{\gamma y_0(\chi)}) / \hat{e}(g_1, g_2^\gamma) = \hat{e}(g_1, g_2^\gamma)^{Q_{wi}(\chi)} = \hat{e}(g_1, g_2^\gamma)^{(\chi - \omega_i) q(\chi) + t} \quad .$$

Then $\hat{e}(g_1, g_2^\gamma)^{q(\chi)+t/(\chi-\omega_i)} = \hat{e}(\pi^*_{uv}, g_2^{\gamma Z(\chi)/(\chi-\omega_i)})$, that is,

$$\hat{e}(g_1, g_2^\gamma)^{1/(\chi-\omega_i)} = (\hat{e}(\pi^*_{uv}, g_2^{\gamma Z(\chi)/(\chi-\omega_i)})/\hat{e}(g_1, g_2^\gamma)^{q(\chi)})^{t^{-1}} .$$

Since $\deg Z(X) = n + 1$, there exist polynomials $q_1(X), q_2(X)$ of degree at most n such that $q(X) = q_1(X)Z(X) + q_2(X)$. Thus we can evaluate $\hat{e}(g_1, g_2^\gamma)^{q(\chi)}$ as $\hat{e}(g_1, g_2^\gamma)^{q_1(\chi)Z(\chi)+q_2(\chi)} = \hat{e}(g_1^{q_1(\chi)}, g_2^{\gamma Z(\chi)}) \cdot \hat{e}(g_1^{q_2(\chi)}, g_2^\gamma)$ by using elements $(g_1^{\chi^i})_{i=0}^n$, g_2^γ, and $g_2^{\gamma Z(\chi)}$. We can also evaluate $g_2^{\gamma Z(\chi)/(\chi-\omega_i)}$ from $(g_2^{\gamma y_i(\chi)})_{i=1}^n$ and $g_2^{\gamma \ell_{n+1}(\chi)}$, since $Z(X)/(X - \omega_i) = \ell_i(X) \prod_{j \neq i}(\omega_i - \omega_j)$, and for $i \neq n+1$, $\ell_i(X) = (y_i(X) - \ell_{n+1}(X))/2$. So since g_2^γ, $g_2^{\gamma Z(\chi)}$ and $g_2^{\gamma \ell_{n+1}(\chi)}$ can all be computed from $(g_2^{\gamma \chi^i})_{i=0}^{n+1}$, $\hat{e}(g_1, g_2^\gamma)^{1/(\chi-\omega_i)}$ can be computed from ch.

Thus, the adversary A_{tsdh}, given as input the TSDH challenge, u_{uv}, π_{uv}, and w_{uv}, can efficiently compute $(r = \omega_j, \hat{e}(g_1, g_2^\gamma)^{1/(\chi-r)})$ (and thus also, with probability $1 - 1/p$ over the choice of γ, $(r, \hat{e}(g_1, g_2)^{1/(\chi-r)})$) for some $j \in [1 \mathinner{..} n+1]$, hence solving the $(n+1)$-TSDH problem with non-negligible probability $(1-1/p) \cdot \varepsilon$. Thus, this argument is an argument of knowledge. □

D Proof of Thm. 3 (Security of Same-Message)

Proof. PERFECT WITNESS-INDISTINGUISHABILITY: All witnesses result in the same argument π_{sm}, hence this argument is witness-indistinguishable. PERFECT COMPLETENESS: Follows from the argumentation preceding the same-message argument description.

ARGUMENT OF KNOWLEDGE: Suppose there exists an adversary A_{aok} that breaks the argument of knowledge property. That is, A_{aok} can output a common input $u_{sm} = (A_1, A_2^\gamma, \hat{A}_1, \hat{A}_2^{\hat{\gamma}})$ and an accepting argument π_{sm}, but (A_1, A_2^γ) and $(\hat{A}_1, \hat{A}_2^{\hat{\gamma}})$ are commitments to different message vectors.

We first use the PKE assumptions to argue that from A_{aok} and certain extractors we can construct another adversary A^*_{aok} who outputs (u_{sm}, π_{sm}) together with a witness $w_{sm} = (\boldsymbol{a}, \hat{\boldsymbol{m}}, \hat{r})$:

- By the $(\dots, \gamma)$-PKE assumption and since the verification equation (1) holds, there exists an extractor that can obtain $\boldsymbol{a} \in \mathbb{Z}_p^{n+2}$ such that $(A_1, A_2^\gamma) = (g_1, g_2^\gamma)^{A(\chi)}$, where $A(X) = \sum_{i=0}^{n+1} a_i X^i$.
- By the $(\dots, \hat{\gamma})$-PKE assumption and since the verification equation (3) holds, there exists an extractor that can obtain $(\hat{\boldsymbol{m}}, \hat{r})$ such that $(\hat{A}_1, \hat{A}_2^{\hat{\gamma}}) = \mathsf{com}(\widehat{\mathsf{ck}}; \hat{\boldsymbol{m}}; \hat{r}) = (g_1, g_2^{\hat{\gamma}})^{\hat{A}(\chi)}$ for some $\hat{A}(X) \in \operatorname{span}(\hat{\mathcal{F}}_{\mathsf{com}})$.

(If any of the extractors fails, then we can abort. By assumption, this happens with a negligible probability.) Thus, assume A^*_{aok} outputs $(u_{sm}, \pi_{sm}, w_{sm} = (\boldsymbol{a}, \hat{\boldsymbol{m}}, \hat{r}))$, such that the verification holds.

As in the case of Thm. 2, if A^*_{aok} is successful, then either

(a) $A(X) \notin \operatorname{span}(\mathcal{F}_{\mathsf{com}})$, or

(b) $A(X) \in \operatorname{span}(\mathcal{F}_{\mathsf{com}})$ but $Z(X) \nmid Q_{wi}(X)$.

Case (a). Assume $A(X) \notin \operatorname{span}(\mathcal{F}_{\mathsf{com}})$. We construct the following $(\hat{n}, \hat{n} + n + 2)$-PCDH adversary A_{pcdh}. A_{pcdh} receives a $(\hat{n}, \hat{n} + n + 2)$-PCDH challenge $ch = (\mathsf{gk}, ((g_1, g_2)^{\chi^i})_{i \in [0 \mathinner{..} \hat{n}+n+2] \setminus \{\hat{n}+1\}})$. Let

$$\mathcal{D} := \left\{ \begin{array}{l} q(X) \in \mathbb{Z}_p[X] : \deg(q) \leq \hat{n} \wedge \\ q(X)f(X) \text{ has a zero coefficient for } X^{\hat{n}} \text{ for all } f \in \mathcal{F}_{\mathsf{com}} \end{array} \right\} .$$

A_{pcdh} picks $q(X)$ randomly from $\mathcal{D}$. Note that $\deg(q(X)f(X)) \leq \hat{n} + (n+1) = (\hat{n} + n + 2) - 1$ for any $f(X) \in \mathcal{F}_{\mathsf{com}}$. There are $(\hat{n} + 1) - |\mathcal{F}_{\mathsf{com}}| > 0$ degrees of freedom for choosing $q(X)$. Thus, for a polynomial $\pi(X)$ outside of the span of $\mathcal{F}_{\mathsf{com}}$, the coefficient of $X^{\hat{n}}$ in $q(X)\pi(X)$ will be random.

A_{pcdh} then picks $b \leftarrow_r \mathbb{Z}_p$, sets $\beta(X) \leftarrow Xq(X) + b$ and $\beta \leftarrow \beta(\chi)$. Since $\mathcal{D}$ consists of polynomials of degree at most $\hat{n}$, then for $f \in \mathcal{F}_{\mathsf{com}}$, $\beta(X)f(X)$ is of degree at most $(\hat{n}+1)+(n+1) = \hat{n}+n+2$, with a zero coefficient for $X^{\hat{n}+1}$. This means that A_{pcdh} can compute $g_1^{\beta f(\chi)}$ from ch by using generic group operations.

With the given values, A_{pcdh} can now generate $(\gamma, \hat{\gamma}) \leftarrow_r \mathbb{Z}_p^2$, and compute a correct CRS which is sent to A^*_{aok}. (To be able to compute the CRS, we need $\hat{n}$ to be like chosen in the theorem.) Suppose that A^*_{aok} replies with $(u_{sm}, \pi_{sm}, w_{sm})$ such that $A(X) = \sum_{i=0}^{n+1} a_i X^i$ is not in the span of $\mathcal{F}_{\mathsf{com}}$, verification succeeds, and $A_1 = g_1^{A(\chi)}$. Since $A(X)$ is not in the span of $\mathcal{F}_{\mathsf{com}}$, the coefficient of $X^{\hat{n}}$ in $q(X)A(X)$ is random. This means that with probability $1 - 1/p$, the coefficient c of $X^{\hat{n}+1}$ in the known polynomial $\beta(X)A(X)$ is non-zero. Since $\hat{e}(A_1, g_2^{\gamma\beta}) = \hat{e}(A_1^{\beta}, g_2^{\gamma})$, A_{pcdh} can compute $A_1^{\beta} = g_1^{\beta A(\chi)}$. However, A_{pcdh} knows all the coefficients of $\beta(X)A(X)$, and hence from ch she can compute $g_1^{\chi^{\hat{n}+1}} = (g_1^{c\chi^{\hat{n}+1}})^{c^{-1}}$. Thus, A_{pcdh} solves the PCDH problem with non-negligible probability $(1 - 1/p) \cdot \varepsilon$.

So $A(X) \in \operatorname{span}((y_i(X))_{i=0}^{n})$ and thus A^*_{aok} knows a witness $(\boldsymbol{m} = (m_1, \dots, m_n), r)$ such that $(A_1, A_2^{\gamma}) = \mathsf{com}(\mathsf{ck}; \boldsymbol{m}; r)$.

Case (b). We construct a $\hat{n}$-TSDH adversary A_{tsdh}. Assume that A_{tsdh} gets as an input a TSDH challenge $ch = ((g_1, g_2)^{\chi^i})_{i=0}^{\hat{n}}$. Then, A_{tsdh} generates random γ, $\hat{\gamma}$, β, and uses them together with ch to generate a correct CRS for A^*_{aok}. (Again, to create correct CRS we need the chosen value of $\hat{n}$.) Assume that (with some probability ε), A^*_{aok} then outputs u_{sm}, accepting argument $\pi_{sm} = (\pi^*_{sm}, A_1^{\beta})$, and a witness w_{sm}.

Since $Z(X) = \prod_{i=1}^{n+1}(X - \omega_i)$ and all the $X - \omega_i$ are pairwise relatively prime, $Z(X) \nmid Q_{wi}(X)$ means there exists $i \in [1 \,..\, n+1]$ such that $(X - \omega_i) \nmid Q_{wi}(X)$. Thus, there exists a non-zero constant $t \in \mathbb{Z}_p$ and a degree $\leq \hat{n} - 1$ polynomial $q(X)$ such that $Q_{wi}(X) = (X - \omega_i)q(X) + t$.

Since $Q_{wi}(X) = \hat{A}(X) - A(X)$ and the verification equation (4) holds, we have that $\hat{e}(\pi^*_{sm}, g_2^{\gamma Z(\chi)}) = \hat{e}(\hat{A}_1/A_1, g_2^{\gamma}) = \hat{e}(g_1, g_2^{\gamma})^{Q_{wi}(\chi)} = \hat{e}(g_1, g_2^{\gamma})^{(\chi - \omega_i)q(\chi) + t}$. But then $\hat{e}(g_1, g_2^{\gamma})^{q(\chi) + t/(\chi - \omega_i)} = \hat{e}(\pi^*_{sm}, g_2^{\gamma Z(\chi)/(\chi - \omega_i)})$, which is equivalent to

$$\hat{e}(g_1, g_2^{\gamma})^{1/(\chi - \omega_i)} = (\hat{e}(\pi^*_{sm}, g_2^{\gamma Z(\chi)/(\chi - \omega_i)})/\hat{e}(g_1, g_2^{\gamma})^{q(\chi)})^{t^{-1}} .$$

Since $\deg Z(X) = n+1$, there exist polynomials $q_1(X), q_2(X)$ of degree at most $\hat{n} - (n+2)$ and n respectively, such that $q(X) = q_1(X)Z(X) + q_2(X)$. Thus, by the same argumentation used in Thm. 2, we can evaluate $\hat{e}(g_1, g_2^{\gamma})^{q(\chi)}$, $\hat{e}(g_1, g_2^{\gamma})^{1/(\chi - \omega_i)}$, and finally, with probability $1 - 1/p$, $(r = \omega_i, \hat{e}(g_1, g_2)^{1/(\chi - r)})$ from the given values ch, breaking the $\hat{n}$-TSDH assumption. Thus, this argument is an argument of knowledge. □

E Proof of Prop. 3 (PSP-friendliness of $\hat{\mathcal{F}}_{\mathsf{com}}$)

Proof. First, since $Z(\omega_k) = 0$, $\hat{y}_j(\omega_k) = y_j(\omega_k)$ and $\hat{Z}(\omega_k) = Z(\omega_k)$ for $j \in [1 \,..\, n], k \in [1 \,..\, n+1]$. To prove $(n+1, \hat{\mathcal{F}}_{\mathsf{com}})$ is PSP-friendly, it suffices to show that the degrees of all polynomials in $(\hat{\mathcal{F}}_{\mathsf{com}})_{n+1}$ are distinct.

- For $j > 0$, $\hat{y}_j(X)$ has degree $(j-1)(n+2) + (n+3) + n = (2n+3) + (j-1)(n+2)$, while $\deg \hat{Z}(X) = (n+1)(n+2) + (n+1) = (2n+3) + n(n+2)$.
- The degrees of all polynomials $X^i \hat{y}_j(X)$, for $i \in [0 \,..\, n+1]$, $j \in [0 \,..\, n]$, are unique (namely, $(2n+3) + (j-1)(n+2) + i$ for $j \in [1 \,..\, n]$ and $(2n+3) + n(n+2) + i$ for $j = 0$) and also larger than $\max\{\deg X^i\}_{i \in [0 \,..\, 2(n+1)]}$.
- The degree of $\hat{Z}(X)\hat{y}_j(X)$ is at least $\deg \hat{Z}(X) + (2n+3) > \deg \hat{Z}(X) + (n+1) = \max\{\deg X^i \hat{y}_j(X)\}_{i \in [0 \,..\, n+1], j \in [0 \,..\, n]}$.

Thus, $(n+1, \hat{\mathcal{F}}_{\mathsf{com}})$ is PSP-friendly, and $\hat{n} = \deg \hat{Z}(X) = (n+1)(n+3)$. □

F Full Proof of Thm. 5 (Security of Shuffle Argument)

Proof. Perfect completeness: Assume that the prover is honest. To verify the proof, the verifier first checks the consistency of the commitment (d_1, d_2^{γ}) and n unit vector arguments; here we use the fact that the unit vector argument is perfectly complete. Moreover, since $\boldsymbol{\Psi}_n = \boldsymbol{1_n} - \sum_{i=1}^{n-1} \boldsymbol{\Psi}_i$, $(c_{n1}, c_{n2}^{\gamma})$ (and similarly, $(\hat{c}_{n1}, \hat{c}_{n2}^{\hat{\gamma}})$) is a commitment to $\boldsymbol{\Psi}_n$ with randomizer r_n. Also, since the same-message argument is perfectly complete, the verification equations on steps (4) and (5) all hold. The verification equations on step (6) hold since for the verification equation (6a) (the verification equation (6b) is true similarly),

$$\begin{aligned}
&\prod_{i=1}^{n} \hat{E}(z'_i, g_2^{\gamma y_i(\chi)}) / \prod_{i=1}^{n} \hat{E}(z_i, c_{i2}^{\gamma}) \\
&\quad = \prod_{i=1}^{n} \hat{E}(z_{\psi(i)} \cdot (g_1, h)^{t_i}, g_2^{\gamma y_i(\chi)}) / \prod_{i=1}^{n} \hat{E}(z_i, g_2^{\gamma(r_i Z(\chi) + y_{\psi^{-1}(i)}(\chi))}) \\
&\quad = \prod_{i=1}^{n} \hat{E}((g_1, h)^{t_i}, g_2^{\gamma y_i(\chi)}) / \prod_{i=1}^{n} \hat{E}(z_i, g_2^{\gamma r_i Z(\chi)}) \\
&\quad = \hat{E}((g_1, h), g_2^{\gamma \sum_{i=1}^{n} t_i y_i(\chi)}) / \hat{E}(\prod_{i=1}^{n} z_i^{r_i}, g_2^{\gamma Z(\chi)}) \\
&\quad = \hat{E}((g_1, h), g_2^{\gamma(r_t Z(\chi) + \sum_{i=1}^{n} t_i y_i(\chi))}) / \hat{E}(\prod_{i=1}^{n} z_i^{r_i} \cdot (g_1, h)^{r_t}, g_2^{\gamma Z(\chi)}) \\
&\quad = \hat{E}((g_1, h), d_2^{\gamma}) / \hat{E}(U, g_2^{\gamma Z(\chi)}) \enspace .
\end{aligned}$$

Culpable soundness: Let $\mathsf{A}_{\mathsf{guilt}}$ be an NUPPT adversary that, given gk and a correctly generated crs, outputs a statement $(\mathsf{pk} = (g_1, h), (z_i, z'_i)_{i=1}^{n})$, a secret key sk, and an accepting shuffle argument π_{sh}, such that the plaintext vector $(z'_i)_{i=1}^{n}$ is not a permutation of the plaintext vector $(z_i)_{i=1}^{n}$. (I.e., $\mathsf{A}_{\mathsf{guilt}}$ is an adversary against $\mathcal{R}_{sh,n}^{\mathsf{guilt}}$.) Assume that the unit vector argument and the same-message argument are arguments of knowledge, and that both PKE assumptions hold (these claims are guaranteed by the assumptions of Thm. 5). We construct the following adversary A_{psp} that breaks the $(\mathcal{F}_{\mathsf{com}}, \hat{\mathcal{F}}_{\mathsf{com}})$-PSP assumption.

A_{psp} obtains input $(\mathsf{gk}, \mathsf{crs}_{psp})$ where $\mathsf{crs}_{psp} = (((g_1, g_2)^{\chi^i})_{i=0}^{n+1}, (g_1, g_2)^{\hat{\mathcal{F}}_{\mathsf{com}}(\chi)})$. A_{psp} then generates random γ, $\hat{\gamma}$, and β. A_{psp} constructs crs_{sh}^{*} as follows:

(a) from $((g_1, g_2)^{\chi^i})_{i=0}^{n+1}$ and $\mathcal{F}_{\mathsf{com}}$, she can create $\mathsf{ck} = (g_1, g_2)^{\mathcal{F}_{\mathsf{com}}(\chi)}$.
(b) from g_1, $g_1^{\mathcal{F}_{\mathsf{com}}(\chi)}$, and β, she can create $g_1^{\beta \mathcal{F}_{\mathsf{com}}(\chi)}$, $g_1^{\mathcal{F}_{uv,1}(\chi, \beta)}$ and $g_1^{\mathcal{F}_{sm,1}(\chi, \beta)}$, and thus $g_1^{\mathcal{F}_{sh,1}(\chi, \beta)}$;
(c) from g_2, γ, and β, she can create g_2^{γ}, $g_2^{\gamma\beta}$, and $g_2^{\gamma \mathcal{F}_{\mathsf{com}}(\chi)}$, and thus $g_2^{\mathcal{F}_{uv,2}(\chi, \beta, \gamma)}$;
(d) finally, from g_2 and $\hat{\gamma}$, she can create $g_2^{\hat{\gamma}}$, and thus $g_2^{\mathcal{F}_{sh,2}(\chi, \beta, \gamma, \hat{\gamma})}$.

So A_{psp} can create a valid CRS $\mathsf{crs}_{sh}^{*} = (\mathsf{crs}_{sh,p}, \mathsf{crs}_{sh,v})$ of the shuffle argument. A_{psp} then sends $\mathsf{crs}_{sh,p}$ to $\mathsf{A}_{\mathsf{guilt}}$, who returns $((\mathsf{pk}, (z_i, z'_i)_{i=0}^{n}), \mathsf{sk}, \pi_{sh})$, such that the verification algorithm ver_{sh} accepts π_{sh}.

Recall that $\pi_{uv,i} = (\pi_{uv,i}^{*} = g_1^{\pi_{wi,i}(\chi)}, c_{i1}^{\beta})$. By applying the relevant knowledge assumption, we can postulate the existence of the following NUPPT knowledge extractors that, with all but a negligible probability, return some witness:

1. By the $(\dots, \gamma)$-PKE assumption and the PCDH assumption, for every $i \in [1 \mathinner{..} n-1]$ there exists a knowledge extractor that, given $(c_{i1}, c_{i2}^{\gamma})$ and access to $\mathsf{A}_{\mathsf{guilt}}$'s random coins, returns $((\boldsymbol{\Psi}_{ij})_{j \in [1 \mathinner{..} n]}, r_i)$ such that $(c_{i1}, c_{i2}^{\gamma}) = \mathsf{com}(\mathsf{ck}; \boldsymbol{\Psi}_i; r_i)$. Let $\boldsymbol{\Psi}_n \leftarrow \boldsymbol{1_n} - \sum_{i=1}^{n-1} \boldsymbol{\Psi}_i$ and $r_n \leftarrow - \sum_{i=1}^{n-1} r_i$.
2. By the the $(\dots, \gamma)$-PKE assumption and the PCDH assumption, there exists a knowledge extractor that, given (d_1, d_2^{γ}) and access to $\mathsf{A}_{\mathsf{guilt}}$'s random coins, returns $(\boldsymbol{t^*}, r_t^*)$, such that $(d_1, d_2^{\gamma}) = \mathsf{com}(\mathsf{ck}; \boldsymbol{t^*}; r_t^*)$.
3. By the $(\dots, \hat{\gamma})$-PKE assumption, for every $i \in [1 \mathinner{..} n-1]$ there exists a knowledge extractor that, given $(\hat{c}_{i1}, \hat{c}_{i2}^{\hat{\gamma}})$ and access to $\mathsf{A}_{\mathsf{guilt}}$'s random coins, returns $((\hat{\boldsymbol{\Psi}}_{ij})_{j \in [1 \mathinner{..} n]}, \hat{r}_i)$, such that $(\hat{c}_{i1}, \hat{c}_{i2}^{\hat{\gamma}}) = \mathsf{com}(\widehat{\mathsf{ck}}; \hat{\boldsymbol{\Psi}}_{\boldsymbol{i}}; \hat{r}_i)$. Let $\hat{\boldsymbol{\Psi}}_{\boldsymbol{n}} \leftarrow \boldsymbol{1_n} - \sum_{i=1}^{n-1} \hat{\boldsymbol{\Psi}}_{\boldsymbol{i}}$ and $\hat{r}_n \leftarrow - \sum_{i=1}^{n-1} \hat{r}_i$.
4. By the $(\dots, \hat{\gamma})$-PKE assumption, there exists a knowledge extractor that, given $(\hat{d}_1, \hat{d}_2^{\hat{\gamma}})$ and access to $\mathsf{A}_{\mathsf{guilt}}$'s random coins, returns $(\hat{\boldsymbol{t}}\boldsymbol{^*}, \hat{r}_t^*)$, such that $(\hat{d}_1, \hat{d}_2^{\hat{\gamma}}) = \mathsf{com}(\widehat{ck}; \hat{\boldsymbol{t}}\boldsymbol{^*}; \hat{r}_t^*)$.

Since we assume the PKE and PCDH assumptions are true, the probability that any of these extractors fails is negligible, in this case we can abort. In the following, we will assume that all extractors succeeded.

Let $\mathfrak{a}$ be $\mathsf{A}_{\mathsf{guilt}}$'s output. Based on $\mathsf{A}_{\mathsf{guilt}}$ and the output of extractors, we can build an adversary $\mathsf{A}_{\mathsf{guilt}}^{*}$ that returns $\mathfrak{a}$ to A_{psp} together with the witness $((\boldsymbol{\Psi}_i, r_i, \hat{\boldsymbol{\Psi}}_{\boldsymbol{i}}, \hat{r}_i)_{i \in [1 \mathinner{..} n-1]}, \boldsymbol{t}^*, r_t^*, \hat{\boldsymbol{t}}^*, \hat{r}_t^*)$.

Since the unit vector argument is an argument of knowledge, $\boldsymbol{\Psi}_n = \boldsymbol{1_n} - \sum_{i=1}^{n-1} \boldsymbol{\Psi}_i$, and $\pi_{uv,i}$ verifies for each $i \in [1 \mathinner{..} n]$, we have that $(c_{i1}, c_{i2}^{\gamma})_{i=1}^{n}$ commits to a permutation matrix. (Otherwise, we can break the unit vector arguments and thus either the PCDH or the TSDH assumption.) Hence, $\boldsymbol{\Psi}$ corresponds to a permutation $\psi \in S_n$, such that for $i \in [1 \mathinner{..} n]$, $c_{i1} = g_2^{r_i + y_{\psi^{-1}(i)}(\chi)}$.

Since the same-message argument is an argument of knowledge, it must be the case that $\boldsymbol{\Psi} = \hat{\boldsymbol{\Psi}}$. In particular, $(\hat{c}_{i1}, \hat{c}_{i2}^{\hat{\gamma}}) = \mathsf{com}(\widehat{\mathsf{ck}}; \boldsymbol{\Psi}_i; \hat{r}_i)$.

Finally, the verification equation (6a) accepts. Then (since we are proving culpable soundness and thus A_{psp} knows sk) setting $\mu^* \leftarrow U_2/U_1^{\mathsf{sk}}$, $\mu_i \leftarrow z_{i2}/z_{i1}^{\mathsf{sk}}$, and $\mu'_i \leftarrow z'_{i2}/(z'_{i1})^{\mathsf{sk}}$, the verification equation (6a) is equivalent to verifying that $\prod_{i=1}^{n} \hat{E}((z'_{i1}, \mu'_i), g_2^{\gamma y_i(\chi)}) / \prod_{i=1}^{n} \hat{E}((z_{i1}, \mu_i), c_{i2}^{\gamma}) = \hat{E}((g_1, 1), d_2^{\gamma}) / \hat{E}((U_1, \mu^*), g_2^{\gamma Z(\chi)})$. Here, A_{psp} knows all mentioned variables. After eliminating γ, we get (with probability $1 - 1/p$ over the choice of γ) that

$$\begin{aligned}\prod_{i=1}^{n} \hat{E}((z'_{i1}, \mu'_i), g_2^{y_i(\chi)}) / \prod_{i=1}^{n} \hat{E}((z_{i1}, \mu_i), g_2^{r_i Z(\chi) + y_{\psi^{-1}(i)}(\chi)}) \\ = \hat{E}((g_1, 1), g_2^{r_t^* Z(\chi) + \sum_{i=1}^{n} t_i^* y_i(\chi)}) / \hat{E}((U_1, \mu^*), g_2^{Z(\chi)}) \enspace .\end{aligned}$$

Set $R_0 = U_1 / (\prod_{i=1}^{n} z_{i1}^{r_i} \cdot g_1^{r_t^*})$, $M_0 = \mu^* / \prod_{i=1}^{n} \mu_i^{r_i}$, $R_i = z'_{i1} / (g_1^{t_i^* y_i(\chi)} z_{\psi(i),1})$, and $M_i = \mu'_i / \mu_{\psi(i)}$ for $i \in [1 \mathinner{..} n]$. Since the pairing is non-degenerate, by reordering the terms we get that $R_0^{Z(\chi)} \cdot \prod_{i=1}^{n} R_i^{y_i(\chi)} = 1$ and $M_0^{Z(\chi)} \cdot \prod_{i=1}^{n} M_i^{y_i(\chi)} = 1$.

From the verification equation (6b), we get that $\hat{R}_0^{\hat{Z}(\chi)} \cdot \prod_{i=1}^{n} R_i^{\hat{y}_i(\chi)} = 1$ and $\hat{M}_0^{\hat{Z}(\chi)} \cdot \prod_{i=1}^{n} M_i^{\hat{y}_i(\chi)} = 1$ for (due to the soundness of the same-message argument) the same values of M_i and R_i for $i \in [1 \mathinner{..} n]$ as before.

Now, due to the construction of the CRS we can apply the PSP assumption to $R_0^{Z(\chi)} \cdot \prod_{i=1}^{n} R_i^{y_i(\chi)} = 1$ and $\hat{R}_0^{\hat{Z}(\chi)} \cdot \prod_{i=1}^{n} R_i^{\hat{y}_i(\chi)} = 1$ (getting $R_i = 1$ for all $i \in [1 \mathinner{..} n]$), and to $M_0^{Z(\chi)} \cdot \prod_{i=1}^{n} M_i^{y_i(\chi)} = 1$ and $\hat{M}_0^{\hat{Z}(\chi)} \cdot \prod_{i=1}^{n} M_i^{\hat{y}_i(\chi)} = 1$ (getting $M_i = 1$ for all $i \in [1 \mathinner{..} n]$). Thus, $(z'_{i1}, z'_{i2}) = (z_{\psi(i),1}, z'_{\psi(i),2})$ for $i \in [1 \mathinner{..} n]$, and the shuffle argument is culpably sound.

Argument of knowledge: follows from the proof of culpable soundness, where ψ and t_i were recovered by using the PKE assumption.

Perfect zero-knowledge: We construct the following simulator S. It inputs $\mathsf{gk} \leftarrow \mathsf{setup}(1^{\kappa}, n)$, $((\mathsf{crs}_{sh,p}, \mathsf{crs}_{sh,v}), \mathsf{td}_{sh}) \leftarrow \mathsf{gencrs}(\mathsf{gk})$, and the common input $(\mathsf{pk}, (z_i, z'_i)_{i=1}^{n})$, and does the following:

(1) Pick $r_t \leftarrow_r \mathbb{Z}_p$. For $i \in [1 \mathinner{..} n-1]$: pick $r_i \leftarrow_r \mathbb{Z}_p$.
(2) Simulate commitments $(c_{i1}, c_{i2}^{\gamma})$:
  (a) For $i \in [1 \mathinner{..} n-1]$: set $(c_{i1}, c_{i2}^{\gamma}) \leftarrow \mathsf{com}(\mathsf{ck}; \boldsymbol{e_i}; r_i)$.
  (b) Set $(c_{n1}, c_{n2}^{\gamma}) \leftarrow \mathsf{com}(\mathsf{ck}; \boldsymbol{e_n}; -\sum_{i=1}^{n-1} r_i)$.
(3) Simulate commitments $(\hat{c}_{i1}, \hat{c}_{i2}^{\hat{\gamma}})$:
  (a) For $i \in [1 \mathinner{..} n-1]$: set $(\hat{c}_{i1}, \hat{c}_{i2}^{\hat{\gamma}}) \leftarrow \mathsf{com}(\widehat{\mathsf{ck}}; \boldsymbol{e_i}; r_i)$.
  (b) Set $(\hat{c}_{n1}, \hat{c}_{n2}^{\gamma}) \leftarrow \mathsf{com}(\widehat{\mathsf{ck}}; \boldsymbol{e_n}; -\sum_{i=1}^{n-1} r_i)$.
(4) For $i \in [1 \mathinner{..} n]$: set $\pi_{uv,i} \leftarrow \mathsf{pro}_{uv}(\mathsf{gk}, \mathsf{crs}_{uv,p}; c_{i1}, c_{i2}^{\gamma}; \boldsymbol{e_i}, r_i)$.
(5) For $i \in [1 \mathinner{..} n-1]$: set $\pi_{sm,i} \leftarrow \mathsf{pro}_{sm}(\mathsf{gk}, \mathsf{crs}_{sm,p}; c_{i1}, c_{i2}^{\gamma}, \hat{c}_{i1}, \hat{c}_{i2}^{\hat{\gamma}}; \boldsymbol{e_i}, r_i, r_i)$.
(6) Set $(d_1, d_2^{\gamma}) \leftarrow \mathsf{com}(\mathsf{ck}; \boldsymbol{0}_n; r_t)$ and $(\hat{d}_1, \hat{d}_2^{\hat{\gamma}}) \leftarrow \mathsf{com}(\widehat{\mathsf{ck}}; \boldsymbol{0}_n; r_t)$.
(7) Set $\pi_{sm,d} \leftarrow \mathsf{pro}_{sm}(\mathsf{gk}, \mathsf{crs}_{sm,p}; d_1, d_2^{\gamma}, \hat{d}_1, \hat{d}_2^{\hat{\gamma}}; \boldsymbol{0_n}, r_t, r_t)$.
(8) Set $U \leftarrow (g_1, h)^{r_t} \cdot \prod_{i=1}^{n} (z_i^{r_i + y_i(\chi)/Z(\chi)} / (z'_i)^{y_i(\chi)/Z(\chi)})$.
(9) Output $\pi_{sh} \leftarrow ((c_{i1}, c_{i2}^{\gamma}, \hat{c}_{i1}, \hat{c}_{i2}^{\hat{\gamma}})_{i=1}^{n-1}, d_1, d_2^{\gamma}, \hat{d}_1, \hat{d}_2^{\hat{\gamma}}, (\pi_{uv,i})_{i=1}^{n}, (\pi_{sm,i}^{*})_{i=1}^{n-1}, \pi_{sm,d}, U)$

Note that we need to ensure $Z(\chi) \neq 0$.

Next, we give an analysis of the simulated proof. Clearly, (d_1, d_2^{γ}), $(\hat{d}_1, \hat{d}_2^{\hat{\gamma}})$, $(c_{i1}, c_{i2}^{\gamma})$, and $(\hat{c}_{i1}, \hat{c}_{i2}^{\hat{\gamma}})$ for $i \in [1 \mathinner{..} n-1]$ are independent and random variables in $\mathbb{G}_1 \times \mathbb{G}_2$, exactly as in the real run of the protocol. Moreover, the values $(c_{n1}, c_{n2}^{\gamma})$ and $(\hat{c}_{n1}, \hat{c}_{n2}^{\hat{\gamma}})$ are generated in the same way as in the real run of the protocol, and the unit vector and same-message arguments are generated exactly as in the honest case. Thus, verification steps (4) to (5) hold.

After that, S defines U so that it satisfies the verification equations. Since U is uniquely defined by the verification equation, it also has the same distribution as in the real protocol. Thus, we are now only left to show that the verification equations in step (6) of the new shuffle argument hold. But

$$\begin{aligned}
&\prod_{i=1}^{n} \hat{E}(z_i', g_2^{\gamma y_i(\chi)}) / \prod_{i=1}^{n} \hat{E}(z_i, c_{12}^{\gamma}) \\
&\quad = \prod_{i=1}^{n} \hat{E}(z_i', g_2^{\gamma y_i(\chi)}) / \prod_{i=1}^{n} \hat{E}(z_i, g_2^{\gamma(r_i Z(\chi) + y_i(\chi))}) \\
&\quad = \hat{E}(\prod_{i=1}^{n} (z_i')^{y_i(\chi)}, g_2^{\gamma}) / \hat{E}(\prod_{i=1}^{n} z_i^{r_i Z(\chi) + y_i(\chi)}, g_2^{\gamma}) \\
&\quad = \hat{E}(\prod_{i=1}^{n} ((z_i')^{y_i(\chi)/Z(\chi)} / z_i^{r_i + y_i(\chi)/Z(\chi)})), g_2^{\gamma Z(\chi)}) \\
&\quad = \hat{E}((g_1, h), g_2^{\gamma r_t Z(\chi)}) / \hat{E}((g_1, h)^{r_t} \cdot \prod_{i=1}^{n} (z_i^{r_i + y_i(\chi)/Z(\chi)} / (z_i')^{y_i(\chi)/Z(\chi)}), g_2^{\gamma Z(\chi)}) \\
&\quad = \hat{E}((g_1, h), d_2^{\gamma}) / \hat{E}(U, g_2^{\gamma Z(\chi)}) \enspace .
\end{aligned}$$

Thus, the verification equation (6a) holds. Analogously, the verification equation (6b) holds, and thus the simulator has succeeded in generating an accepting argument that has the same distribution as the real argument. □

G Proof of Prop. 4 (Efficiency of Shuffle Argument)

Proof. CRS. The prover's CRS of the unit vector argument consists of $4n + 2$ (resp., $n + 1$) elements of $\mathbb{G}_1$ (resp., $\mathbb{G}_2$), while the verifier's CRS consists of 2 / 4 / 1 elements of $\mathbb{G}_1$ / $\mathbb{G}_2$ / $\mathbb{G}_T$, respectively. The CRS of the shuffle argument needs $2(n+1)+3 = 2n+5$ additional elements of $\mathbb{G}_1$ and $(n+1)+2 = n+3$ additional elements of $\mathbb{G}_2$ for the prover, and 2 (resp., $(2n + 4)$) additional elements of $\mathbb{G}_1$ (resp., $\mathbb{G}_2$) for the verifier. Thus, the claim about the length of the CRS follows.

Communication. The communication complexity of the unit vector arguments $(\pi_{uv,i})_{i=1}^{n}$ is $2n - 1$ elements of $\mathbb{G}_1$ (by the construction of c_{n1}, we do not need to send c_{n1}^{β}), part of the same-message argument $\pi_{sm,i}^{*}$ adds $n - 1$ elements of $\mathbb{G}_1$, and the same-message argument $\pi_{sm,d}$ adds 2 elements of $\mathbb{G}_1$. The shuffle argument proper adds $(2n + 2)$ (resp., $(2n)$) elements of $\mathbb{G}_1$ (resp., $\mathbb{G}_2$), and hence the result.

Prover's computational complexity. We write down the computation step-by-step:

(1) $(c_{i1}, c_{i2}^{\gamma})_{i=1}^{n-1}$: commitments are of the form $(c_{i1}, c_{i2}^{\gamma}) = (g_1^{P_0(\chi)}, g_2^{\gamma P_0(\chi)})^{r_i} \cdot \prod_{i=1}^{n} (g_1^{P_i(\chi)}, g_2^{\gamma P_i(\chi)})^{a_i}$. For an honest prover $a_i \in \{0, 1\}$, so $n - 1$ exponentiations (by r_i) are needed in both $\mathbb{G}_1$ and $\mathbb{G}_2$.

(2) $(\hat{c}_{i1}, \hat{c}_{i2})^{\gamma})_{i=1}^{n-1}$: by a similar argument, requires $n - 1$ exponentiations in both $\mathbb{G}_1$ and $\mathbb{G}_2$.

(3) $(\pi_{uv,i})_{i=1}^{n}$: n two-wide exponentiations and n exponentiations in $\mathbb{G}_1$, which can be reduced by one exponentiation, since from $r_n = -\sum_{i=1}^{n-1} r_i$ we can compute c_{n1}^{β} using only multiplications.

(4) (d_1, d_2^{γ}): one $(n + 1)$-wide multi-exponentiation in both $\mathbb{G}_1$ and $\mathbb{G}_2$,

(5) $(\hat{d}_1, \hat{d}_2^{\gamma})$: one $(n + 1)$-wide multi-exponentiation in both $\mathbb{G}_1$ and $\mathbb{G}_2$,

(6) $(\pi_{sm,i}^{*}, c_{i1}^{\beta})_{i=1}^{n-1}$: for an honest prover, $\pi_{uv,i} = (\pi_{uv,i}^{*}, c_{i1}^{\beta})$ uses the same c_{i1}^{β}, so we only need to compute $\pi_{sm,i}^{*} = (g_1^{\hat{Z}(\chi)/Z(\chi)})^{\hat{r}} \cdot g_1^{-r} \cdot \prod_{i=1}^{n} (g_1^{(\hat{P}_i(\chi) - y_i(\chi))/Z(\chi)})^{a_i}$ for $i \in [1 \,..\, n-1]$. Similar to $\pi_{uv,i}^{*}$, this only needs $n - 1$ exponentiations (using $\hat{r} = r$, and $(g_1^{\hat{Z}(\chi)/Z(\chi)})^{r} \cdot g_1^{-r} = (g_1^{\hat{Z}(\chi)/Z(\chi)}/g_1)^{r}$).

(7) $\pi_{sm,d}$: one $(n + 2)$-wide multi-exponentiation and one $(n + 1)$-wide multi-exponentiation in $\mathbb{G}_1$,

(8) U (online): two $(n + 1)$-wide multi-exponentiations in $\mathbb{G}_1$.

For the sake of simplicity, assume that an $(n+2)$-wide multi-exponentiations are computed by executing one $(n+1)$-wide multi-exponentiation and one exponentiation, and that a three-wide / two-wide multi-exponentiation is as expensive as three / two exponentiations.

Thus, in total the prover has to execute

(i) $(n-1)+(n-1)+(3n-1)+0+0+(n-1)+1+0 = 6n-3$ exponentiations in $\mathbb{G}_1$,

(ii) $0+0+0+1+1+0+2+2 = 6$ $(n+1)$-wide multi-exponentiations in $\mathbb{G}_1$,

(iii) $(n-1)+(n-1)+0+0+0+0+0+0 = 2n-2$ exponentiations in $\mathbb{G}_2$,

(iv) $0+0+0+1+1+0+0+0 = 2$ $(n+1)$-multi-exponentiations in $\mathbb{G}_2$.

Thus we get $16n+3$ exponentiations in total, only $8n-5$ of which are not part of a multi-exponentiation.

Verifier's computational complexity. Each unit vector argument can be verified using 6 pairings. However, for the n unit vector arguments we do not need to check the consistency of $(c_{n1}, c_{n2}^{\gamma})$ or c_{n1}^{β}, saving 4 pairings.

The shuffle argument proper requires additional bilinear pairings :

(i) two to verify the consistency of $(\hat{c}_{i1}, \hat{c}_{i2}^{\hat{\gamma}})$ for $i \in [1 \mathinner{..} n-1]$,

(ii) four to verify the consistency of (d_1, d_2^{γ}) and $(\hat{d}_1, \hat{d}_2^{\hat{\gamma}})$,

(iii) two to verify $\pi_{sm,i}^*$ for $i \in [1 \mathinner{..} n-1]$,

(iv) four to verify the consistency of $\pi_{sm,d}$,

(v) $8(n+1)$ to verify the step 6.

However, $\hat{e}(g_1, d_2^{\gamma})$ and $\hat{e}(g_1, \hat{d}_2^{\hat{\gamma}})$ are used in steps 5 and 6 of the verification, but only need to be computed once. Thus, the shuffle argument adds $12n+10$ pairings to the verifier's $6n-4$ pairings for the n unit vector arguments, which means $18n+6$ pairings in total. □

H SP Assumption

Assume that we have a symmetric pairing, i.e., $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$. The Simultaneous Pairing [GL07] assumption states that for any $n = \text{poly}(\kappa)$ and NUPPT adversary A,

$$\Pr\left[\begin{array}{l}\mathsf{gk} := (p, \mathbb{G}, \mathbb{G}_T, \hat{e}) \leftarrow \mathsf{BP}(1^\kappa, n), g_1 \leftarrow_r \mathbb{G}^*, \boldsymbol{\chi} \leftarrow_r \mathbb{Z}_p^n,\\ (s_i)_{i=1}^n \leftarrow \mathsf{A}(\mathsf{gk}, \{(g_1^{\chi_i}, g_1^{\chi_i^2})\}_{i=1}^n) : (s_i)_{i=1}^n \in \mathbb{G}^n \wedge \\ \prod_{i=1}^n s_i^{\chi_i} = \prod_{i=1}^n s_i^{\chi_i^2} = 1 \wedge (\exists i \in [1 \mathinner{..} n] : s_i \neq 1)\end{array}\right] \approx_\kappa 0 \enspace .$$

The PSP assumption can be seen as a q-type variant of the SP assumption, where instead of n independent random variables χ_i and χ_i^2, one uses the values $y_i(\chi)$ and $\hat{y}_i(\chi)$ for single random variable χ and public polynomials $y_i(X)$, $\hat{y}_i(X)$. The use of independent random variables makes the SP assumption conceptually simpler; however, it means that the prover of the shuffle argument has a considerably longer secret key. Finally, since the new efficient same-message argument does not guarantee that the randomizers used in two commitments are equal, the adversary of the PSP assumption is allowed to output two non-one values t and $\hat{t}$. Due to the use of independent random variables χ_i and the use of specific "polynomials" χ_i^2 in second "commitment", in the case of [GL07] there exist an almost trivial same-message argument.

I Preliminaries: Knowledge Assumptions

We give the formal definitions of the knowledge assumptions introduced in Section 2.

Let $1 < d(n) < d^*(n) = \text{poly}(\kappa)$ be two functions. We say that BP is

- *$d(n)$-PDL (Power Discrete Logarithm, [Lip12]) secure* if for any $n = \text{poly}(\kappa)$ and NUPPT adversary A,

$$\Pr\left[\begin{array}{l}\mathsf{gk} \leftarrow \mathsf{BP}(1^\kappa, n), (g_1, g_2, \chi) \leftarrow_r \mathbb{G}_1^* \times \mathbb{G}_2^* \times \mathbb{Z}_p : \\ \mathsf{A}(\mathsf{gk}, ((g_1, g_2)^{\chi^i})_{i=0}^{d(n)}) = \chi\end{array}\right] \approx_\kappa 0 \enspace .$$

- $(d(n), d^*(n))$*-PCDH (Power Computational Diffie-Hellman, [GJM02,Gro10b,GGPR13]) secure* if for any $n = \mathrm{poly}(\kappa)$ and NUPPT adversary A,

$$\Pr\left[\begin{array}{l}\mathsf{gk} \leftarrow \mathsf{BP}(1^\kappa, n), (g_1, g_2, \chi) \leftarrow_r \mathbb{G}_1^* \times \mathbb{G}_2^* \times \mathbb{Z}_p : \\ \mathsf{A}(\mathsf{gk}, ((g_1, g_2)^{\chi^i})_{i \in [0\,..\,d^*(n)] \setminus \{d(n)+1\}}) = g_1^{\chi^{d(n)+1}}\end{array}\right] \approx_\kappa 0 \enspace .$$

- $d(n)$*-TSDH (Target Strong Diffie-Hellman, [BB04,PGHR13]) secure* if for any $n = \mathrm{poly}(\kappa)$ and NUPPT adversary A,

$$\Pr\left[\begin{array}{l}\mathsf{gk} \leftarrow \mathsf{BP}(1^\kappa, n), (g_1, g_2, \chi) \leftarrow_r \mathbb{G}_1^* \times \mathbb{G}_2^* \times \mathbb{Z}_p : \\ \mathsf{A}\left(\mathsf{gk}, ((g_1, g_2)^{\chi^i})_{i=0}^{d(n)}\right) = \left(r, \hat{e}(g_1, g_2)^{1/(\chi - r)}\right) \wedge r \neq \chi\end{array}\right] \approx_\kappa 0 \enspace .$$

J Preliminaries: Zero Knowledge

Let $\mathcal{R} = \{(u, w)\}$ be an efficiently computable binary relation with $|w| = \mathrm{poly}(|u|)$. Here, u is a statement, and w is a witness. Let $\mathcal{L} = \{u : \exists w, (u, w) \in \mathcal{R}\}$ be an **NP**-language. Let $n = |u|$ be the input length. For fixed n, we have a relation $\mathcal{R}_n$ and a language $\mathcal{L}_n$. Here, as in [GL07], since we argue about group elements, both $\mathcal{L}_n$ and $\mathcal{R}_n$ are group-dependent and thus we add gk as an input to $\mathcal{L}_n$ and $\mathcal{R}_n$. Let $\mathcal{R}_n(\mathsf{gk}) := \{(u, w) : (\mathsf{gk}, u, w) \in \mathcal{R}_n\}$.

A *non-interactive argument* for a group-dependent relation family $\mathcal{R}$ consists of four PPT algorithms: a setup algorithm setup, a common reference string (CRS) generator gencrs, a prover pro, and a verifier ver. For $\mathsf{gk} \leftarrow \mathsf{setup}(1^\kappa, n)$ (where n is the input length) and $(\mathsf{crs} = (\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td}) \leftarrow \mathsf{gencrs}(\mathsf{gk})$ (where td is not accessible to anybody but the simulator), $\mathsf{pro}(\mathsf{crs}_p; u, w)$ produces an argument π, and $\mathsf{ver}(\mathsf{crs}_v; u, \pi)$ outputs either 1 (accept) or 0 (reject). Here, crs_p (resp., crs_v) is the part of the CRS given to the prover (resp., the verifier). Distinction between crs_p and crs_v is not important from the security point of view, but in many cases crs_v is significantly shorter.

Π is *perfectly complete*, if for all $n = \mathrm{poly}(\kappa)$,

$$\Pr\left[\begin{array}{l}\mathsf{gk} \leftarrow \mathsf{setup}(1^\kappa, n), ((\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td}) \leftarrow \mathsf{gencrs}(\mathsf{gk}), (u, w) \leftarrow \mathcal{R}_n(\mathsf{gk}) : \\ \mathsf{ver}(\mathsf{gk}, \mathsf{crs}_v; u, \mathsf{pro}(\mathsf{gk}, \mathsf{crs}_p; u, w)) = 1\end{array}\right] = 1 \enspace .$$

Π is adaptively *computationally sound* for $\mathcal{L}$, if for all $n = \mathrm{poly}(\kappa)$ and non-uniform probabilistic polynomial-time A,

$$\Pr\left[\begin{array}{l}\mathsf{gk} \leftarrow \mathsf{setup}(1^\kappa, n), ((\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td}) \leftarrow \mathsf{gencrs}(\mathsf{gk}), \\ (u, \pi) \leftarrow \mathsf{A}(\mathsf{gk}, \mathsf{crs}_p, \mathsf{crs}_v) : (\mathsf{gk}, u) \not\in \mathcal{L}_n \wedge \mathsf{ver}(\mathsf{gk}, \mathsf{crs}_v; u, \pi) = 1\end{array}\right] \approx_\kappa 0 \enspace .$$

We recall that in situations where the inputs have been committed by using a computationally binding trapdoor commitment scheme, the notion of computational soundness does not make sense (since the commitments could be to any input messages). Instead, one should either proof culpable soundness or the argument of knowledge property.

Π is adaptively *computationally culpably sound* [GL07,GOS12] for $\mathcal{L}$, if for all $n = \mathrm{poly}(\kappa)$, for all polynomial-time decidable binary relations $\mathcal{R}^{\mathsf{guilt}} = \{\mathcal{R}_n^{\mathsf{guilt}}\}$ consisting of elements from $\bar{\mathcal{L}}$ and witnesses w^{guilt}, and for all non-uniform probabilistic polynomial-time A,

$$\Pr\left[\begin{array}{l}\mathsf{gk} \leftarrow \mathsf{setup}(1^\kappa, n), ((\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td}) \leftarrow \mathsf{gencrs}(\mathsf{gk}), \\ (u, \pi, w^{\mathsf{guilt}}) \leftarrow \mathsf{A}(\mathsf{gk}, \mathsf{crs}_p, \mathsf{crs}_v) : \\ (\mathsf{gk}, u, w^{\mathsf{guilt}}) \in \mathcal{R}_n^{\mathsf{guilt}} \wedge \mathsf{ver}(\mathsf{gk}, \mathsf{crs}_v; u, \pi) = 1\end{array}\right] \approx_\kappa 0 \enspace .$$

Π is *an argument of knowledge*, if for all $n = \mathrm{poly}(\kappa)$ and every non-uniform probabilistic polynomial-time A, there exists a non-uniform probabilistic polynomial-time extractor X, such that for every auxiliary input $\mathsf{aux} \in \{0,1\}^{\mathrm{poly}(\kappa)}$,

$$\Pr\left[\begin{array}{l}\mathsf{gk} \leftarrow \mathsf{setup}(1^\kappa, n), ((\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td}) \leftarrow \mathsf{gencrs}(\mathsf{gk}),\\ ((u, \pi); w) \leftarrow (\mathsf{A}||X_\mathsf{A})(\mathsf{crs}_p, \mathsf{crs}_v; \mathsf{aux}) :\\ (u, w) \notin \mathcal{R} \wedge \mathsf{ver}(\mathsf{crs}_v; u, \pi) = 1\end{array}\right] \approx_\kappa 0 \enspace .$$

Here, the notation $\mathsf{A}||X_\mathsf{A}$ is defined in Sect. 2. As in the definition of PKE (see Sect. 2), we can restrict the definition of an argument of knowledge to benign auxiliary information generators, where aux is known to come from. For the sake of simplicity, we omit further discussion.

$\mathit{\Pi}$ is *perfectly witness-indistinguishable*, if for all $n = \mathrm{poly}(\kappa)$, if $\mathsf{gk} \in \mathsf{setup}(1^\kappa, n)$, $((\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td}) \in \mathsf{gencrs}(\mathsf{gk})$, and $((\mathsf{gk}, u, w_0), (\mathsf{gk}, u, w_1)) \in \mathcal{R}_n^2$, then the distributions $\mathsf{pro}(\mathsf{crs}_p; u, w_0)$ and $\mathsf{pro}(\mathsf{crs}_p; u, w_1)$ are equal. $\mathit{\Pi}$ is *perfectly zero-knowledge*, if there exists a probabilistic polynomial-time simulator $\mathcal{S}$, such that for all stateful non-uniform probabilistic polynomial-time adversaries A and $n = \mathrm{poly}(\kappa)$,

$$\Pr\left[\begin{array}{l}\mathsf{gk} \leftarrow \mathsf{setup}(1^\kappa, n),\\ ((\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td}) \leftarrow \mathsf{gencrs}(\mathsf{gk}),\\ (u, w) \leftarrow \mathsf{A}(\mathsf{gk}, \mathsf{crs}_p, \mathsf{crs}_v),\\ \pi \leftarrow \mathsf{pro}(\mathsf{gk}, \mathsf{crs}_p; u, w) :\\ (\mathsf{gk}, u, w) \in \mathcal{R}_n \wedge \mathsf{A}(\mathsf{gk}, \pi) = 1\end{array}\right] = \Pr\left[\begin{array}{l}\mathsf{gk} \leftarrow \mathsf{setup}(1^\kappa, n),\\ ((\mathsf{crs}_p, \mathsf{crs}_v); \mathsf{td}) \leftarrow \mathsf{gencrs}(\mathsf{gk}),\\ (u, w) \leftarrow \mathsf{A}(\mathsf{gk}, \mathsf{crs}_p, \mathsf{crs}_v),\\ \pi \leftarrow \mathcal{S}(\mathsf{gk}, \mathsf{crs}_p, \mathsf{crs}_v; u, \mathsf{td}) :\\ (\mathsf{gk}, u, w) \in \mathcal{R}_n \wedge \mathsf{A}(\mathsf{gk}, \pi) = 1\end{array}\right] \enspace .$$

Here, the prover and the simulator use the same CRS. That is, we have *same-string zero knowledge* [DDO$^+$01]. We recall that same-string statistical zero knowledge allows to use the same CRS an unbounded number of times.