IACR Communications in Cryptology
ISSN 3006-5496, Vol. 2, No. 1, 20 pages.

https://doi.org/10.62056/a6qj893y6

Bulletproofs for R1CS: Bridging the Completeness-Soundness Gap and a ZK Extension

Gil Segev

School of Computer Science and Engineering, Hebrew University of Jerusalem, Israel
Coinbase, USA

Abstract. Bulletproofs, introduced by Bünz, Bootle, Boneh, Poelstra, Wuille and Maxwell (IEEE S&P, 2018), is a highly efficient non-interactive argument system that does not require a trusted setup. Recently, Bünz (PhD Thesis, 2023) extended Bulletproofs to support arguments for rank-1 constraint satisfaction (R1CS) systems, a widely-used representation for arithmetic satisfiability problems. Although the argument system constructed by Bünz preserves the attractive properties of Bulletproofs, it presents a gap between its completeness and soundness guarantees: The system is complete for a restricted set of instances, but sound for only a significantly broader set. Although argument systems for such gap relations nevertheless provide clear and concrete guarantees, the gaps they introduce may lead to various inconsistencies or undesirable gaps within proofs of security, especially when used as building blocks within larger systems.

In this work we show that the argument system presented by Bünz can be extended to bridge the gap between its completeness and soundness, and to additionally provide honest-verifier zero-knowledge. For the extended argument system, we introduce a refined R1CS relation that captures the precise set of instances for which both completeness and soundness hold without resorting to a gap formulation. The extended argument system preserves the performance guarantees of the argument system presented by Bünz, and yields a non-interactive argument system using the Fiat-Shamir transform.

Keywords: Zero Knowledge · Bulletproofs · R1CS

1 Introduction

Bulletproofs is a practical argument system constructed by Bünz, Bootle, Boneh, Poelstra, Wuille and Maxwell [BBB+18], building on the techniques of Bootle, Cerulli, Chaidos, Groth and Petit [BCC+16]. The Bulletproofs argument system does not require a trusted setup, and provides logarithmic-length inner-product, range and arithmetic circuit satisfiability arguments relative to a committed witness. The practical applicability of Bulletproofs was further demonstrated by Bünz [Bün23], who constructed a Bulletproofs argument system for rank-1 constraint satisfaction (R1CS) systems relative to a committed witness. Such systems generalize arithmetic circuits, and have become the interface to a variety of state-of-the-art argument systems (see, for example, [GGPR13, Gro16, BCR+19, CHM+20, OB22] and the references therein). Specifically, an R1CS instance is parameterized by integers $m, n \geq 1$ and a prime q, and consists of three matrices $A, B, C \in \mathbb{Z}_q^{m \times n}$. In turn, an R1CS witness is vector $\boldsymbol{z} \in \mathbb{Z}_q^n$ that satisfies $A\boldsymbol{z} \circ B\boldsymbol{z} = C\boldsymbol{z}$, where $\circ$ denotes the Hadamard

This work was carried out while at Coinbase.
E-mail: segev@cs.huji.ac.il (Gil Segev)

Received: 2025-01-08 Accepted: 2025-03-11

(element-wise) product.[1]

Bulletproofs for R1CS. The argument system constructed by Bünz [Bün23] enables to prove that a given R1CS system is satisfiable by a witness such that a commitment to a part (or to all) of the witness is provided together with the R1CS instance. Specifically, Bünz constructed logarithmic-length arguments for the following relation $\mathcal{R}_{\mathrm{R1CS}}$ which is defined with respect to parameters $m, r, n \in \mathbb{N}$, where $m \geq 1$ and $1 \leq r \leq n$, and a cyclic group $\mathbb{G}$ of prime order q with generators $\boldsymbol{G}, \boldsymbol{H} \in \mathbb{G}^{n+m}$. The relation $\mathcal{R}_{\mathrm{R1CS}}$ consists of all pairs $((T, A, B, C), (\boldsymbol{x}, \boldsymbol{y}))$, where

$$T \in \mathbb{G}, \quad A, B, C \in \mathbb{Z}_q^{m \times n}, \quad \boldsymbol{x} \in \mathbb{Z}_q^r, \quad \boldsymbol{y} \in \mathbb{Z}_q^{n-r},$$

which satisfy the following two requirements[2]:

1. $T = \langle ((\boldsymbol{x} || 0^{n-r}) \mid\mid 0^m), \boldsymbol{G} \rangle$.
2. $A\boldsymbol{z} \circ B\boldsymbol{z} = C\boldsymbol{z}$ for $\boldsymbol{z} = (\boldsymbol{x} || \boldsymbol{y}) \in \mathbb{Z}_q^n$.

That is, the group element T is a commitment to $\boldsymbol{x}$, and the matrices A, and B and C define an R1CS system that is satisfied by $\boldsymbol{z} = (\boldsymbol{x} || \boldsymbol{y})$. In his description of the relation $\mathcal{R}_{\mathrm{R1CS}}$, Bünz allows the group element T to be of the more general form

$$T = \langle ((\boldsymbol{x} || \boldsymbol{u}) \mid\mid 0^m), \boldsymbol{G} \rangle + \langle (\boldsymbol{v} \mid\mid 0^m), \boldsymbol{H} \rangle$$

for some $\boldsymbol{u} \in \mathbb{Z}_q^{n-r}$ and $\boldsymbol{v} \in \mathbb{Z}_q^n$ which, as noted by Bünz, can be set to the all-zero vectors. However, his argument system does not provide completeness for such instances (i.e., when $\boldsymbol{u}$ and $\boldsymbol{v}$ may be arbitrary vectors instead of the all-zero vectors). In addition, when analyzing the soundness of his argument system, Bünz presented an algorithm that extracts a witness $\boldsymbol{z} = (\boldsymbol{x} || \boldsymbol{y})$ for which $A\boldsymbol{z} \circ B\boldsymbol{z} = C\boldsymbol{z}$ as required, but for which T is of the even more general form

$$T = \langle ((\boldsymbol{x} || \boldsymbol{t_2}) \mid\mid \boldsymbol{t_3}), \boldsymbol{G} \rangle + \langle ((\boldsymbol{t'_1} || \boldsymbol{t'_2}) \mid\mid \boldsymbol{t'_3}), \boldsymbol{H} \rangle$$

for some $\boldsymbol{t'_1} \in \mathbb{Z}_q^r$, $\boldsymbol{t_2}, \boldsymbol{t'_2} \in \mathbb{Z}_q^{n-r}$ and $\boldsymbol{t'_3} \in \mathbb{Z}_q^m$. As the vectors $\boldsymbol{t_3}$ and $\boldsymbol{t'_3}$ may not be the all-zero vectors, this is not a valid witness even with respect to instances of the general form $T = \langle ((\boldsymbol{x} || \boldsymbol{u}) \mid\mid 0^m), \boldsymbol{G} \rangle + \langle (\boldsymbol{v} \mid\mid 0^m), \boldsymbol{H} \rangle$. In fact, as we demonstrate in Section 4, there are instances in which $\boldsymbol{t_3}$ and $\boldsymbol{t'_3}$ are not the all-zero vectors, and for which an efficient prover can always convince the verifier to accept. Thus, this gap cannot be bridged by only refining the analysis provided by Bünz.

The gap between the completeness and soundness in the argument system presented by Bünz can be formalized by viewing his argument system as an argument system for a "gap" relation: Given an "outer" relation $\mathcal{R}_{\mathsf{out}}$ and an "inner" relation $\mathcal{R}_{\mathsf{in}} \subsetneq \mathcal{R}_{\mathsf{out}}$, the honest prover can convince the verifier to accept any instance x when provided with an inner-witness w such that $(x, w) \in \mathcal{R}_{\mathsf{in}}$, whereas any efficient malicious prover that convinces the verifier to accept an instance x with a non-negligible probability can be used to extract an outer-witness w such that $(x, w) \in \mathcal{R}_{\mathsf{out}}$. That is, completeness is provided for instances of the inner relation $\mathcal{R}_{\mathsf{in}}$, whereas soundness is guaranteed for instances of the outer relation $\mathcal{R}_{\mathsf{out}}$. Although argument systems for such gap relations nevertheless provide clear and concrete guarantees, the gaps they introduce may lead to various inconsistencies or undesirable gaps within proofs of security, especially when used as building blocks within larger systems.

[1] For any $\ell \geq 1$, $\boldsymbol{x} = (x_1, \ldots, x_\ell) \in \mathbb{Z}_q^\ell$ and $\boldsymbol{y} = (y_1, \ldots, y_\ell) \in \mathbb{Z}_q^\ell$, the Hadamard product $\boldsymbol{x} \circ \boldsymbol{y} \in \mathbb{Z}_q^\ell$ is defined as $\boldsymbol{x} \circ \boldsymbol{y} = (x_1 \cdot y_1, \ldots, x_\ell \cdot y_\ell) \in \mathbb{Z}_q^\ell$.

[2] For any $\ell \geq 1$, $\boldsymbol{c} = (c_1, \ldots, c_\ell) \in \mathbb{Z}_q^\ell$ and $\boldsymbol{G} = (G_1, \ldots, G_\ell) \in \mathbb{G}^\ell$, we let $\langle \boldsymbol{c}, \boldsymbol{G} \rangle = \sum_{i=1}^{\ell} c_i \cdot G_i \in \mathbb{G}$.

Our contributions. In this work we show that the argument system presented by Bünz can be extended to bridge the above-discussed gap between its completeness and soundness, and to additionally provide honest-verifier zero-knowledge. Specifically, we introduce a relation $\mathcal{R}_{\mathrm{R1CS}^*}$ that lies in between the outer and inner relations resulting from the analysis presented by Bünz, and extend his argument system to provide completeness, soundness and honest-verifier zero-knowledge for $\mathcal{R}_{\mathrm{R1CS}^*}$ without considering gap relations. The extended argument system preserves the performance guarantees of the argument system presented by Bünz, where for instances in which $T = \langle((\boldsymbol{x}||0^{n-r}) \mid\mid 0^m), \boldsymbol{G}\rangle$ the two argument systems coincide. Furthermore, the extended argument system is public coin and yields a non-interactive argument system using the Fiat-Shamir transform [FS87, AFK23].

2 Preliminaries

Let $\mathbb{G}$ be a cyclic group of prime order q that is generated by $G \in \mathbb{G}$. Throughout this document we denote scalars via lower-case letters (e.g., $x \in \mathbb{Z}_q$), vectors of scalars via boldface lower-case letters (e.g., $\boldsymbol{x} = (x_1, \ldots, x_\ell) \in \mathbb{Z}_q^\ell$), group elements via upper-case letters (e.g., $G \in \mathbb{G}$), and vectors of group elements via boldface upper-case letters (e.g., $\boldsymbol{G} = (G_1, \ldots, G_\ell) \in \mathbb{G}^\ell$). For $\boldsymbol{x} = (x_1, \ldots, x_\ell) \in \mathbb{Z}_q^\ell$, $\boldsymbol{y} = (y_1, \ldots, y_\ell) \in \mathbb{Z}_q^\ell$ and $\boldsymbol{G} = (G_1, \ldots, G_\ell) \in \mathbb{G}^\ell$, we let $\langle \boldsymbol{x}, \boldsymbol{G}\rangle = \sum_{i=1}^{\ell} x_i \cdot G_i \in \mathbb{G}$ and $\boldsymbol{x} \circ \boldsymbol{y} = (x_1 \cdot y_1, \ldots, x_\ell \cdot y_\ell) \in \mathbb{Z}_q^\ell$. For $x \in \mathbb{Z}_q^*$ and integer $\ell \geq 1$ we let $\boldsymbol{x}^\ell = (x, \ldots, x^\ell) \in \mathbb{Z}_q^\ell$.

2.1 Interactive Argument Systems

An interactive argument system for a relation $\mathcal{R} = \{\mathcal{R}_\sigma\}_{\sigma \in \{0,1\}^*}$ is a triplet $\Pi = (\mathcal{K}, \mathcal{P}, \mathcal{V})$ of probabilistic polynomial-time algorithms. The common-reference string generation algorithm $\mathcal{K}$ receives as input the unary representation 1^κ of the security parameter $\kappa \in \mathbb{N}$, and outputs a common-reference string σ. For any $\kappa \in \mathbb{N}$ and for any σ produced by $\mathcal{K}(1^\kappa)$, the prover algorithm $\mathcal{P}$ and the verifier algorithm $\mathcal{V}$ define an interactive protocol $\langle \mathcal{P}(\sigma, \cdot, \cdot), \mathcal{V}(\sigma, \cdot)\rangle$. The input of the prover consists of a common-reference string σ, an instance x and a witness w, and the input of the verifier consists of the common-reference string σ and the instance x. We assume without loss of generality that any common-reference string σ produced by $\mathcal{K}(1^\kappa)$ is of length of least κ bits (thus, we do not need to provide $\mathcal{P}$ and $\mathcal{V}$ with 1^κ as part of their input). We denote by $\mathsf{tr} \leftarrow \langle \mathcal{P}(\sigma, x, w), \mathcal{V}(\sigma, x)\rangle$ the probabilistic process of producing a transcript of the protocol, and denote by $\mathsf{Out}_{\mathcal{V}}\langle \mathcal{P}(\sigma, x, w), \mathcal{V}(\sigma, x)\rangle$ the random variable corresponding to the output of the verifier, in both cases when the prover and verifier follow the instructions of the protocol.

In what follows we present the standard notions of completeness, honest-verifier zero-knowledge and witness-extended emulation for interactive argument systems.

Definition 1 (Perfect Completeness)**.** An interactive argument system $\Pi = (\mathcal{K}, \mathcal{P}, \mathcal{V})$ for a relation $\mathcal{R} = \{\mathcal{R}_\sigma\}_{\sigma \in \{0,1\}^*}$ has *perfect completeness* if for any algorithm $\mathcal{A}$ and for any $\kappa \in \mathbb{N}$ it holds that

$$\Pr\left[(x, w) \notin \mathcal{R}_\sigma \text{ or } \mathsf{Out}_{\mathcal{V}}\langle \mathcal{P}(\sigma, x, w), \mathcal{V}(\sigma, x)\rangle = 1 \;\middle|\; \begin{array}{c} \sigma \leftarrow \mathcal{K}(1^\kappa) \\ (x, w) \leftarrow \mathcal{A}(\sigma) \end{array}\right] = 1.$$

Definition 2 (Public Coin)**.** An interactive argument system $\Pi = (\mathcal{K}, \mathcal{P}, \mathcal{V})$ is *public coin* if all messages sent by the honest verifier are uniformly distributed and independent of the honest verifier's input and of all messages previously sent by the prover.

Definition 3 (Perfect Special Honest-Verifier Zero-Knowledge)**.** An interactive argument system $\Pi = (\mathcal{K}, \mathcal{P}, \mathcal{V})$ for a relation $\mathcal{R} = \{\mathcal{R}_\sigma\}_{\sigma \in \{0,1\}^*}$ has *perfect special honest-verifier* *zero-knowledge* if there exists a probabilistic polynomial-time simulator $\mathcal{S}$ such that for any algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ it holds that

$$\begin{aligned}&\Pr\left[\begin{array}{c}(x,w)\in\mathcal{R}_\sigma\\ \text{and } \mathcal{A}_2(\mathsf{tr})=1\end{array}\middle|\begin{array}{c}\sigma\leftarrow\mathcal{K}(1^\kappa),(x,w,\rho)\leftarrow\mathcal{A}_1(\sigma)\\ \mathsf{tr}\leftarrow\langle\mathcal{P}(\sigma,x,w),\mathcal{V}(\sigma,x;\rho)\rangle\end{array}\right]\\ &\qquad=\Pr\left[\begin{array}{c}(x,w)\in\mathcal{R}_\sigma\\ \text{and } \mathcal{A}_2(\mathsf{tr})=1\end{array}\middle|\begin{array}{c}\sigma\leftarrow\mathcal{K}(1^\kappa),(x,w,\rho)\leftarrow\mathcal{A}_1(\sigma)\\ \mathsf{tr}\leftarrow\mathcal{S}(\sigma,x,\rho)\end{array}\right]\end{aligned}$$

for all $\kappa\in\mathbb{N}$, where $\rho\in\{0,1\}^*$ denotes the randomness of the verifier $\mathcal{V}$.

Definition 4 (Witness-Extended Emulation [Lin03])**.** An interactive argument system $\Pi=(\mathcal{K},\mathcal{P},\mathcal{V})$ for a relation $\mathcal{R}=\{\mathcal{R}_\sigma\}_{\sigma\in\{0,1\}^*}$ has *statistical witness-extended emulation* if for any algorithm $\mathcal{P}^*$ there exists an expected polynomial-time emulator $\mathcal{E}$ such that for any algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ there exists a negligible function $\nu(\cdot)$ such that

$$\begin{aligned}&\left|\Pr\left[\mathcal{A}_2(\mathsf{tr})=1\middle|\begin{array}{c}\sigma\leftarrow\mathcal{K}(1^\kappa),(x,u)\leftarrow\mathcal{A}_1(\sigma)\\ \mathsf{tr}\leftarrow\langle\mathcal{P}^*(\sigma,x,u),\mathcal{V}(\sigma,x)\rangle\end{array}\right]\right.\\ &\qquad\left.-\Pr\left[\begin{array}{c}\mathcal{A}_2(\mathsf{tr})=1\text{ and}\\ \mathsf{tr}\text{ is accepting }\implies(x,w)\in\mathcal{R}_\sigma\end{array}\middle|\begin{array}{c}\sigma\leftarrow\mathcal{K}(1^\kappa),(x,u)\leftarrow\mathcal{A}_1(\sigma)\\ (\mathsf{tr},w)\leftarrow\mathcal{E}^{\mathcal{O}}(\sigma,x)\end{array}\right]\right|\leq\nu(\kappa)\end{aligned}$$

for all $\kappa\in\mathbb{N}$, where the oracle $\mathcal{O}=\langle\mathcal{P}^*(\sigma,x,u),\mathcal{V}(\sigma,x)\rangle$ permits rewinding to a specific point and resuming with fresh randomness for the verifier. Such an argument system has *computational* witness-extended emulation if the above holds when restricting $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{P}^*$ to probabilistic polynomial time.

In order to prove that the argument systems we present provide witness-extended emulation, we rely on the general forking lemma of Bootle, Cerulli, Chaidos, Groth and Petit [BCC+16] that we now state by following their presentation. Let Π be a $(2\mu+1)$-move argument system with μ challenges $c_1,\ldots,c_\mu$. Let $n_1,\ldots,n_\mu\geq 1$ and consider $\Pi_{i=1}^{\mu}n_i$ accepting transcripts in the following tree structure: The tree is labeled with a statement x, it has depth μ and $\Pi_{i=1}^{\mu}n_i$ leaves, where each node at depth i has n_i children labeled with distinct values for the ith challenge c_i. We refer to such transcripts as an $(n_1,\ldots,n_\mu)$-tree of accepting transcripts. For simplicity, in the following lemma Bootle et al. assumed that the challenges are chosen uniformly from $\mathbb{Z}_q$, for a λ-bit prime q such that $\lambda=\Omega(\kappa)$, where κ is the security parameter (we note that the lemma holds also when some or all of the challenges are chosen uniformly from $\mathbb{Z}_q^*$).

Lemma 1 ([BCC+16])**.** *Let $\mu=\mu(\kappa)$ be a function of the security parameter $\kappa\in\mathbb{N}$, let Π be a $(2\mu+1)$-move public-coin argument system for a relation $\mathcal{R}$, and for each $i\in[\mu]$ let $n_i=n_i(\kappa)\geq 1$ such that $\Pi_{i=1}^{\mu}n_i$ is upper bounded by a polynomial in κ. If there exists a probabilistic polynomial-time algorithm that always succeeds in extracting a valid witness from any $(n_1,\ldots,n_\mu)$-tree of accepting transcripts, then Π has statistical witness-extended emulation.*

2.2 Inner-Product Arguments

As a building block for constructing an R1CS argument system, following Bünz [Bün23] we rely on an argument system for the inner-product relation $\mathcal{R}_{\mathrm{IP}}$ defined as follows:

The Relation $\mathcal{R}_{\mathbf{IP}}$

Let $d\geq 1$ be an integer, and let $\mathbb{G}$ be a cyclic group of prime order q. The relation $\mathcal{R}_{\mathrm{IP}}$ consists of all pairs $((\boldsymbol{G},\boldsymbol{H},G,H,P,\omega),(\boldsymbol{u},\boldsymbol{v},\alpha))$, where

$$\boldsymbol{G},\boldsymbol{H}\in\mathbb{G}^d,\quad G,H,P\in\mathbb{G},\qquad \boldsymbol{u},\boldsymbol{v}\in\mathbb{Z}_q^d,\quad \omega,\alpha\in\mathbb{Z}_q,$$

which satisfy the following requirements:

1. $P = \langle \boldsymbol{u}, \boldsymbol{G} \rangle + \langle \boldsymbol{v}, \boldsymbol{H} \rangle + \alpha \cdot H$.
2. $\omega = \langle \boldsymbol{u}, \boldsymbol{v} \rangle$.

The relation $\mathcal{R}_{\text{IP}}$ slightly differs from the inner-product relation considered by Bünz [Bün23] and by Bünz, Bootle, Boneh, Poelstra, Wuille and Maxwell [BBB+18], which did not allow the group element P to contain the additional term $\alpha \cdot H$ (in our setting this additional term is used for additionally providing honest-verifier zero-knowledge). Nevertheless, we show that the approach of Bünz et al. for constructing an argument system for the inner-product relation they considered applies also to the more subtle relation $\mathcal{R}_{\text{IP}}$. Specifically, we show that an argument system for the above-defined relation $\mathcal{R}_{\text{IP}}$ can be obtained from an argument system for the following “multiplicative” inner-product relation $\mathcal{R}_{\text{mIP}}$ that is defined as follows:

The Relation $\mathcal{R}_{\text{mIP}}$

Let $d \geq 1$ be an integer, and let $\mathbb{G}$ be a cyclic group of prime order q. The relation $\mathcal{R}_{\text{mIP}}$ consists of all pairs $((\boldsymbol{G}, \boldsymbol{H}, G, H, P), (\boldsymbol{u}, \boldsymbol{v}, \alpha))$, where

$$\boldsymbol{G}, \boldsymbol{H} \in \mathbb{G}^d, \quad G, H, P \in \mathbb{G}, \qquad \boldsymbol{u}, \boldsymbol{v} \in \mathbb{Z}_q^d, \quad \alpha \in \mathbb{Z}_q,$$

which satisfy

$$P = \langle \boldsymbol{u}, \boldsymbol{G} \rangle + \langle \boldsymbol{v}, \boldsymbol{H} \rangle + \langle \boldsymbol{u}, \boldsymbol{v} \rangle \cdot G + \alpha \cdot H \ .$$

We construct an argument system for the relation $\mathcal{R}_{\text{IP}}$ based on the argument system constructed by Chung, Han, Ju, Kim and Seo [CHJ+22] for the relation $\mathcal{R}_{\text{mIP}}$.[3] Specifically, given a probabilistic polynomial-time group-generation algorithm that on input the security parameter $\kappa \in \mathbb{N}$ produces a description of a cyclic group $\mathbb{G}$ of a κ-bit prime order q and $2d + 2$ uniformly and independently sampled generators $\boldsymbol{G}, \boldsymbol{H} \in \mathbb{G}^d$ and $G, H \in \mathbb{G}$, we prove the following theorem in Appendix A:

Theorem 1. *Let $t : \mathbb{N} \to \mathbb{N}$ be any function of the security parameter $\kappa \in \mathbb{N}$ such that $d = d(\kappa) = 2^{t(\kappa)}$ is polynomial. Assuming the hardness of the DL problem for expected polynomial-time algorithms, there exists an argument system Π_{IP} for the d-dimensional inner-product relation $\mathcal{R}_{IP}$ that has perfect completeness, perfect special honest-verifier zero-knowledge, and computational witness-extended emulation. Furthermore, the argument system is public coin, and the prover communicates $2 \cdot \log_2 d + 2$ group elements and 3 field elements.*

As noted in Section 2.1, in order to prove that the R1CS* argument system we present provides witness-extended emulation, we rely on the general forking lemma of Bootle, Cerulli, Chaidos, Groth and Petit [BCC+16] (see Lemma 1 above). For this purpose, we rely on the following lemma that we prove in Appendix A for establishing the witness-extended emulation property of the argument system Π_{IP}:

Lemma 2. *There exists a probabilistic polynomial-time algorithm* Ext *that, on input any $\mathcal{R}_{IP}$ instance $(\boldsymbol{G}, \boldsymbol{H}, G, H, P, \omega)$ together with any corresponding $(2, 4, \ldots, 4, 5)$-transcript tree of depth $\log_2 d + 2$ for the argument system Π_{IP}, where $d = 2^t \geq 1$, produces either a witness $(\boldsymbol{u}, \boldsymbol{v}, \alpha)$ such that $((\boldsymbol{G}, \boldsymbol{H}, G, H, P, \omega), (\boldsymbol{u}, \boldsymbol{v}, \alpha)) \in \mathcal{R}_{IP}$ or a non-trivial discrete-logarithm relation for $(\boldsymbol{G}, \boldsymbol{H}, G, H)$.*

Finally, we note that in our above formulation we have included the public generators $(\boldsymbol{G}, \boldsymbol{H}, G, H)$ as part of the instance for the relations $\mathcal{R}_{\text{IP}}$ and $\mathcal{R}_{\text{mIP}}$, whereas for the relation $\mathcal{R}_{\text{R1CS}^*}$ which will be defined in the Section 3 section we have included them as public parameters. In our setting, this is a matter of convenience due to the recursive structure

[3] The argument system of Chung et al. [CHJ+22] in fact supports a more general, weighted, form of the multiplicative inner-product relation which is not needed for our purposes

of the considered argument systems for the relations $\mathcal{R}_{\mathrm{IP}}$ and $\mathcal{R}_{\mathrm{mIP}}$, where essentially a different sequence of generators is used for each recursive invocation. Therefore, we have included these generators as part of the instance. Conversely, our argument system for the relation $\mathcal{R}_{\mathrm{R1CS}^*}$ utilizes a fixed sequence of generators. This fixed nature aligns with the conventional understanding of a public parameters, where parameters are established once and remain constant. This distinction does not affect the security of our overall argument system. Our witness-extended emulation relies on the ability to extract witnesses from any transcript tree, which includes the generators, whether fixed or non-fixed. In particular, this ensures witness-extended emulation when the generators are honestly generated as intended.

3 The R1CS* Relation

In this section we introduce the $\mathcal{R}_{\mathrm{R1CS}^*}$ relation which, as discussed above, lies in between the outer and inner relations resulting from the analysis presented by Bünz. The $\mathcal{R}_{\mathrm{R1CS}^*}$ relation is defined as follows:

The Relation $\boldsymbol{\mathcal{R}_{\mathrm{R1CS}^*}}$

Let $m, r, n \in \mathbb{N}$ be such that $m \geq 1$ and $1 \leq r \leq n$, let $\mathbb{G}$ be a cyclic group of prime order q, and let $\boldsymbol{G}, \boldsymbol{H} \in \mathbb{G}^{n+m}$ and $G, H \in \mathbb{G}$ be $2(n+m)+2$ generators. The relation $\mathcal{R}_{\mathrm{R1CS}^*}$ consists of all pairs $((T, A, B, C), (\boldsymbol{x}, \boldsymbol{x'}, \boldsymbol{y}, \boldsymbol{y'}, \eta))$, where

$$T \in \mathbb{G}, \quad A, B, C \in \mathbb{Z}_q^{m \times n}, \quad \boldsymbol{x}, \boldsymbol{x'} \in \mathbb{Z}_q^r, \quad \boldsymbol{y}, \boldsymbol{y'} \in \mathbb{Z}_q^{n-r}, \quad \eta \in \mathbb{Z}_q$$

which satisfy the following requirements for $\boldsymbol{z} = (\boldsymbol{x} || \boldsymbol{y}) \in \mathbb{Z}_q^n$ and $\boldsymbol{z'} = (\boldsymbol{x'} || \boldsymbol{y'}) \in \mathbb{Z}_q^n$:

1. $T = \langle((\boldsymbol{x}||\boldsymbol{y'}) \mid\mid A\boldsymbol{z'}), \boldsymbol{G}\rangle + \langle(0^n \mid\mid B\boldsymbol{z'}), \boldsymbol{H}\rangle + \eta \cdot H$.
2. $A\boldsymbol{z} \circ B\boldsymbol{z} = C\boldsymbol{z}$.
3. $A\boldsymbol{z'} \circ B\boldsymbol{z'} = 0^m$.
4. $A\boldsymbol{z} \circ B\boldsymbol{z'} + B\boldsymbol{z} \circ A\boldsymbol{z'} = C\boldsymbol{z'}$.
5. $A_{[1:r]}\boldsymbol{x'} = B_{[1:r]}\boldsymbol{x'} = C_{[1:r]}\boldsymbol{x'} = 0^m$, where $A_{[1:r]}, B_{[1:r]}, C_{[1:r]} \in \mathbb{Z}^{m \times r}$ denote the leftmost r columns of the matrices A, B and C, respectively.

The relation $\mathcal{R}_{\mathrm{R1CS}^*}$ differs from the relation considered by Bünz in two aspects. First, the relation $\mathcal{R}_{\mathrm{R1CS}^*}$ considers an additional generator H which is used both for providing honest-verifier zero-knowledge and for supporting instances in which T already consists of an additional randomizing element $\eta \cdot H$ (where η is made part of the witness).

Second, when comparing the relation $\mathcal{R}_{\mathrm{R1CS}^*}$ to the outer and inner relations resulting from the analysis presented by Bünz (while ignoring the additional randomizing element $\eta \cdot H$ for the purpose of this comparison), note that the requirement $T = \langle((\boldsymbol{x}||\boldsymbol{y'}) \mid\mid A\boldsymbol{z'}), \boldsymbol{G}\rangle + \langle(0^n \mid\mid B\boldsymbol{z'}), \boldsymbol{H}\rangle$ indeed lies between the requirement $T = \langle((\boldsymbol{x}||0^{n-r}) \mid\mid 0^m), \boldsymbol{G}\rangle$ to which the argument system presented by Bünz provides completeness, and the requirement $T = \langle((\boldsymbol{x}||\boldsymbol{t_2}) \mid\mid \boldsymbol{t_3}), \boldsymbol{G}\rangle + \langle((\boldsymbol{t'_1}||\boldsymbol{t'_2}) \mid\mid \boldsymbol{t'_3}), \boldsymbol{H}\rangle$ to which it provides soundness. Furthermore, note that the additional requirements (i.e., requirements 3–5) are always satisfied for $\boldsymbol{z'} = 0^n$, and thus the relation $\mathcal{R}_{\mathrm{R1CS}^*}$ indeed contains all instances to which the argument system presented by Bünz provides completeness. Thus, we expect that for most applications an argument system for the $\mathcal{R}_{\mathrm{R1CS}^*}$ relation may be used directly for proving R1CS instances in which $T = \langle((\boldsymbol{x}||0^{n-r}) \mid\mid 0^m), \boldsymbol{G}\rangle$.

If, for some applications, it is nevertheless required that soundness holds specifically for $T = \langle((\boldsymbol{x}||0^{n-r}) \mid\mid 0^m), \boldsymbol{G}\rangle$, this can be resolved by additionally providing an argument of knowledge for the relation that consists of all instances $((G_1, \dots G_r, T), \boldsymbol{x})$ where $T = \langle \boldsymbol{x}, (G_1, \dots, G_r)\rangle$ and $\boldsymbol{x} \in \mathbb{Z}_q^r$.[4] As shown by Attema and Cramer [AC20], such an

[4] Note that if $T = \langle((\boldsymbol{x}||\boldsymbol{y'}) \mid\mid A\boldsymbol{z'}), \boldsymbol{G}\rangle + \langle(0^n \mid\mid B\boldsymbol{z'}), \boldsymbol{H}\rangle + \eta \cdot H$ and $T = \langle((\hat{\boldsymbol{x}}||0^{n-r}) \mid\mid 0^m), \boldsymbol{G}\rangle$ for

argument can be provided either via a Σ-protocol in which the prover sends a single group element and r field elements, or by extending the Σ-protocol to a $\lceil \log_2(r+1) \rceil$-round public-coin protocol in which the prover sends $2 \cdot \lceil \log_2(r+1) \rceil$ group elements and 3 field elements (thus, the classic Σ-protocol is a good practical fit for applications in which r is comparable to $\log_2(m+n)$). However, given that providing such an additional proof increases the concrete overhead (potentially even doubling it), it may be preferable to first examine whether the security analysis of application under consideration can be refined to rely on the relation $\mathcal{R}_{\mathrm{R1CS}^*}$ and thus avoid the additional proof.

4 An R1CS* ZK-Argument System

In this section we present and analyze an argument system Π_{R1CS^*} for the $\mathcal{R}_{\mathrm{R1CS}^*}$ relation. Following the approach of Bünz, Bootle, Boneh, Poelstra, Wuille and Maxwell [BBB+18], the key idea observed by Bünz [Bün23] is that an R1CS instance (T, A, B, C) with respect to a given set of generators, where $T \in \mathbb{G}$ and $A, B, C \in \mathbb{Z}_q^{m \times n}$, can be transformed in a probabilistic manner into a single $\mathcal{R}_{\mathrm{IP}}$ instance with respect to a modified set of generators (see Section 2.2 for the definition of the relation $\mathcal{R}_{\mathrm{IP}}$). That is, the m constraints defined by the instance (T, A, B, C) can combined into a single inner-product constraint.

Specifically, note that a solution $\boldsymbol{z} \in \mathbb{Z}_q^n$ to the system $A\boldsymbol{z} \circ B\boldsymbol{z} = C\boldsymbol{z}$ exists if and only if there exist vectors $\boldsymbol{w_A}, \boldsymbol{w_B} \in \mathbb{Z}_q^n$ such that: (1) $A\boldsymbol{z} - \boldsymbol{w_A} = 0^m$, (2) $B\boldsymbol{z} - \boldsymbol{w_B} = 0^m$, and (3) $C\boldsymbol{z} - \boldsymbol{w_A} \circ \boldsymbol{w_B} = 0^m$. On the one hand, this increases the number of constraints from m to $3m$. On the other hand, however, this "uncouples" the Hadamard product $A\boldsymbol{z} \circ B\boldsymbol{z}$ involving the solution $\boldsymbol{z}$. Such an uncoupling is crucial since the given commitment T commits to the individual entries of $\boldsymbol{z}$ and not to the products of each two individual entries of $\boldsymbol{z}$. Now, for proving that $A\boldsymbol{z} - \boldsymbol{w_A} = 0^m$, following an appropriate commitment from the prover, the verifier can uniformly sample a scalar $\alpha \leftarrow \mathbb{Z}_q^*$, and the inner-product argument system can be used for proving that $\langle \boldsymbol{\alpha}^m, A\boldsymbol{z} - \boldsymbol{w_A} \rangle = 0$, where $\boldsymbol{\alpha}^m = (\alpha, \ldots, \alpha^m) \in \mathbb{Z}_q^m$. If the prover successfully convinces the verifier with a non-negligible probability over the choice of α, then this holds for $m+1$ distinct values of α, and we are ensured that $A\boldsymbol{z} - \boldsymbol{w_A} = 0^m$. The same approach can be taken for proving that $B\boldsymbol{z} - \boldsymbol{w_B} = 0^m$ and $C\boldsymbol{z} - \boldsymbol{w_A} \circ \boldsymbol{w_B} = 0^m$, resulting in three inner-product instances. For ensuring that the same $\boldsymbol{w_A}$ and $\boldsymbol{w_B}$ are used, and for further compressing the three inner-product instances into a single instance, the verifier samples $\alpha, \beta, \gamma \leftarrow \mathbb{Z}_q^*$, and the argument system includes a wide variety of technical complications for enabling the parties to essentially construct an inner-product instance roughly equivalent to the constraint

$$\langle \boldsymbol{\alpha}^m, A\boldsymbol{z} - \boldsymbol{w_A} \rangle + \langle \boldsymbol{\beta}^m, B\boldsymbol{z} - \boldsymbol{w_B} \rangle + \langle \boldsymbol{\gamma}^m, C\boldsymbol{z} - \boldsymbol{w_A} \circ \boldsymbol{w_B} \rangle = 0 \ .$$

In what follows we present the argument system, and then state and prove its completeness, soundness and zero-knowledge guarantees. The argument system Π_{R1CS^*}, which uses as a building block the inner-product argument system Π_{IP} provided by Theorem 1 (which we prove in Appendix A), is defined as follows (our modifications to the argument system presented by Bünz, which are discussed below, are colored red for convenience):

The Argument System $\Pi_{\mathbf{R1CS}^*}$

- **Public parameters:**
  1. Integers $m, r, n \in \mathbb{N}$ such that $m \geq 1$, $1 \leq r \leq n$, and $n + m = 2^t$ for some integer $t \geq 1$.

$\boldsymbol{x} \neq \hat{\boldsymbol{x}}$, then this provides a non-trivial discrete-logarithm relation for $(\boldsymbol{G}, \boldsymbol{H}, H)$. Therefore, assuming the hardness of the DL problem for expected polynomial-time algorithms, the additional argument guarantees that $\boldsymbol{x} = \hat{\boldsymbol{x}}$.

2. Cyclic group $\mathbb{G}$ of prime order q and $2(n+m)+2$ generators $\boldsymbol{G}, \boldsymbol{H} \in \mathbb{G}^{n+m}$ and $G, H \in \mathbb{G}$.

- **Inputs:**
  1. $\mathcal{P}$: Instance (T, A, B, C) and witness $(\boldsymbol{x}, \boldsymbol{x'}, \boldsymbol{y}, \boldsymbol{y'}, \eta) \in \mathbb{Z}_q^r \times \mathbb{Z}_q^r \times \mathbb{Z}_q^{n-r} \times \mathbb{Z}_q^{n-r} \times \mathbb{Z}_q$.
  2. $\mathcal{V}$: Instance (T, A, B, C).
- **Execution:**
  1. The prover $\mathcal{P}$ samples $r \leftarrow \mathbb{Z}_q$, lets $\boldsymbol{z} = (\boldsymbol{x}||\boldsymbol{y}) \in \mathbb{Z}_q^n$, computes
  $$S = \langle((\boldsymbol{x'}||\boldsymbol{y}) \;||\; A\boldsymbol{z}), \boldsymbol{G}\rangle + \langle(0^n \;||\; B\boldsymbol{z}), \boldsymbol{H}\rangle + r \cdot H \in \mathbb{G}$$
  and sends S to the verifier $\mathcal{V}$.
  2. The verifier $\mathcal{V}$ samples $\alpha, \beta, \gamma, \delta \leftarrow \mathbb{Z}_q^*$ independently and uniformly, and sends $(\alpha, \beta, \gamma, \delta)$ to the prover $\mathcal{P}$.
  3. Letting $\boldsymbol{G} = (G_1, \ldots, G_{n+m})$, each party computes
  $$\begin{aligned}
  \mu &= \alpha \cdot \gamma \in \mathbb{Z}_q\\
  \boldsymbol{\delta} &= (\delta, \ldots, \delta, 1^{n-r}) \in \mathbb{Z}_q^n\\
  \boldsymbol{\delta^{-1}} &= (\delta^{-1}, \ldots, \delta^{-1}, 1^{n-r}) \in \mathbb{Z}_q^n\\
  \boldsymbol{G'} &= (G_1, \ldots, G_n, \gamma^{-1} \cdot G_{n+1}, \ldots, \gamma^{-m} \cdot G_{n+m}) \in \mathbb{G}^{n+m}\\
  \boldsymbol{c} &= \boldsymbol{\mu}^m A + \boldsymbol{\beta}^m B - \boldsymbol{\gamma}^m C \in \mathbb{Z}_q^n\\
  \omega &= \langle \boldsymbol{\alpha}^m, \boldsymbol{\beta}^m \rangle + \delta^2 \cdot \langle \boldsymbol{\alpha}^n, \boldsymbol{c} \circ \boldsymbol{\delta} \rangle \in \mathbb{Z}_q\\
  P &= \delta^{-1} \cdot T + S + \langle(\delta^2 \cdot \boldsymbol{\alpha}^n \;||\; -\boldsymbol{\beta}^m), \boldsymbol{G'}\rangle + \langle(\boldsymbol{c} \circ \boldsymbol{\delta} \;||\; -\boldsymbol{\alpha}^m), \boldsymbol{H}\rangle \in \mathbb{G}
  \end{aligned}$$
  and the prover $\mathcal{P}$ additionally lets $\boldsymbol{z'} = (\boldsymbol{x'}||\boldsymbol{y'}) \in \mathbb{Z}_q^n$ and computes
  $$\begin{aligned}
  \boldsymbol{u} &= ((\boldsymbol{x'}||\boldsymbol{y}) + \delta^{-1} \cdot (\boldsymbol{x}||\boldsymbol{y'}) + \delta^2 \cdot \boldsymbol{\alpha}^n \;||\; (A\boldsymbol{z} + \delta^{-1} \cdot A\boldsymbol{z'}) \circ \boldsymbol{\gamma}^m - \boldsymbol{\beta}^m) \in \mathbb{Z}_q^{n+m}\\
  \boldsymbol{v} &= (\boldsymbol{c} \circ \boldsymbol{\delta} \;||\; B\boldsymbol{z} - \boldsymbol{\alpha}^m + \delta^{-1} \cdot B\boldsymbol{z'}) \in \mathbb{Z}_q^{n+m}\\
  \eta' &= r + \delta^{-1} \cdot \eta
  \end{aligned}$$
  4. The parties invoke the inner-product argument Π_{IP} with the instance $(\boldsymbol{G'}, \boldsymbol{H}, G, H, P, \omega)$, where the prover $\mathcal{P}$ takes the role of the prover using the witness $(\boldsymbol{u}, \boldsymbol{v}, \eta')$, and the verifier $\mathcal{V}$ takes the role of the verifier and then outputs its output.

The argument system Π_{R1CS^*} is obtained from the argument system presented by Bünz by first introducing the additional vectors $\boldsymbol{x'}$ and $\boldsymbol{y'}$ for supporting the specific structure of the group element T as specified by the relation $\mathcal{R}_{\mathrm{R1CS}^*}$, and by introducing the element $\eta \cdot H$ for randomizing the group element S in order to provide semi-honest zero-knowledge. Then, by additionally modifying the group element S and the vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, as described above in red font for supporting the additional vectors $\boldsymbol{x'}$ and $\boldsymbol{y'}$, this enables to provide both completeness and soundness for the relation $\mathcal{R}_{\mathrm{R1CS}^*}$ (setting $\boldsymbol{x'} = 0^r$, $\boldsymbol{y'} = 0^{n-r}$ and $\eta = 0$ yields the argument system presented by Bünz).

The following theorem captures the completeness, soundness, zero-knowledge and prover communication complexity of the argument system Π_{R1CS^*}. For formalizing the soundness of the argument system we assume that the cyclic group $\mathbb{G}$ is produced by a group-generation algorithm, and that the generators $\boldsymbol{G}$, $\boldsymbol{H}$, G and H are uniformly and independently sampled. This enables us to prove that the argument system provides computational witness-extended emulation assuming the hardness of the DL problem for expected polynomial-time algorithms. Specifically, following Bootle et al. [BCC+16] and Bünz et al. [BBB+18], when including the description of the group and generators as part of the instance to the relation, we show that the argument system provides statistical witness-extended emulation for extracting either a valid witness or a non-trivial discrete-logarithm relation for $(\boldsymbol{G}, \boldsymbol{H}, G, H)$.

Theorem 2. *Let $t : \mathbb{N} \to \mathbb{N}$ be any function of the security parameter $\kappa \in \mathbb{N}$ such that $2^{t(\kappa)}$ is polynomial. Assuming the hardness of the DL problem for expected polynomial-time algorithms, then for any polynomials $m = m(\kappa)$, $r = r(\kappa)$ and $n = n(\kappa)$ such that $m \geq 1$, $1 \leq r \leq n$ and $n + m = 2^t$, it holds that Π_{R1CS^*} is an argument system for the relation $\mathcal{R}_{R1CS^*}$ with perfect completeness, perfect special honest-verifier zero-knowledge and computational witness-extended emulation. Furthermore, the argument system is public coin, and the prover communicates $2 \cdot \log_2(n + m) + 3$ group elements and 3 field elements.*

The proof of Theorem 2 is provided in Sections 4.1, 4.2 and 4.3 and which consider the completeness, zero-knowledge and witness-extended emulation properties, respectively. As for the communication complexity, note that the prover communicates a single group element, and then additionally communicates $2 \cdot \log_2(n + m) + 2$ group elements and 3 field elements when taking the role of the prover in the inner-product argument Π_{IP} (see Theorem 1 for the properties of the inner-product argument system Π_{IP}).

Padding R1CS* instances. Note that we presented the argument system Π_{R1CS^*} for parameters n and m such that $n + m$ is a power of 2, since we presented the argument system Π_{IP} for any dimension which is a power of 2. Dealing with the more general case in which $n + m$ may not be a power of 2 can be done by padding any R1CS* instance (T, A, B, C) with "empty constraints". That is, by adding $m' < n + m$ all-zero rows to the matrices $A, B, C \in \mathbb{Z}_q^{m \times n}$ to obtain matrices $A', B', C' \in \mathbb{Z}_q^{(m+m') \times n}$ for which $n + m + m'$ is a power of 2. As a result, this requires including $2m'$ additional generators $G_{n+m+1}, H_{n+m+1}, \ldots, G_{n+m+m'}, H_{n+m+m'} \in \mathbb{G}$ in the public parameters. In terms of communication complexity, since the prover sends $2 \cdot \log_2(n + m) + 3$ group elements and 3 field elements, then this would increase the prover's communication by at most two group elements (and would increase the number of rounds by at most one).

We observe that for any $\boldsymbol{w} = (\boldsymbol{x}, \boldsymbol{x'}, \boldsymbol{y}, \boldsymbol{y'}, \eta) \in \mathbb{Z}_q^r \times \mathbb{Z}_q^r \times \mathbb{Z}_q^{n-r} \times \mathbb{Z}_q^{n-r} \times \mathbb{Z}_q$, it holds that $\boldsymbol{w}$ is a valid witness for an instance (T, A, B, C) if and only if it is a valid witness for the padded instance (T, A', B', C'). Specifically, for $\boldsymbol{z} = (\boldsymbol{x}||\boldsymbol{y})$ and $\boldsymbol{z'} = (\boldsymbol{x'}||\boldsymbol{y'})$, it holds that $A'\boldsymbol{z'} = \left(A\boldsymbol{z'}||0^{m'}\right)$ and $B'\boldsymbol{z'} = \left(B\boldsymbol{z'}||0^{m'}\right)$, and therefore

$$T = \langle((\boldsymbol{x}||\boldsymbol{y'}) \;||\; A\boldsymbol{z'}), \boldsymbol{G}\rangle + \langle(0^n \;||\; B\boldsymbol{z'}), \boldsymbol{H}\rangle + \eta \cdot H$$

if and only if

$$\begin{aligned} T &= \langle((\boldsymbol{x}||\boldsymbol{y'}) \;||\; A'\boldsymbol{z'}), (\boldsymbol{G}||G_{n+m+1}, \ldots, G_{n+m+m'})\rangle \\ &\quad + \langle(0^n \;||\; B'\boldsymbol{z'}), (\boldsymbol{H}||H_{n+m+1}, \ldots, H_{n+m+m'})\rangle + \eta \cdot H \ . \end{aligned}$$

Similarly, it holds that:

- $A\boldsymbol{z} \circ B\boldsymbol{z} = C\boldsymbol{z}$ if and only if $A'\boldsymbol{z} \circ B'\boldsymbol{z} = C'\boldsymbol{z}$.
- $A\boldsymbol{z'} \circ B\boldsymbol{z'} = 0^m$ if and only if $A'\boldsymbol{z'} \circ B'\boldsymbol{z'} = 0^{m+m'}$.
- $A\boldsymbol{z} \circ B\boldsymbol{z'} + B\boldsymbol{z} \circ A\boldsymbol{z'} = C\boldsymbol{z'}$ if and only if $A'\boldsymbol{z} \circ B'\boldsymbol{z'} + B'\boldsymbol{z} \circ A'\boldsymbol{z'} = C'\boldsymbol{z'}$.
- $A_{[1:r]}\boldsymbol{x'} = B_{[1:r]}\boldsymbol{x'} = C_{[1:r]}\boldsymbol{x'} = 0^m$ if and only if $A'_{[1:r]}\boldsymbol{x'} = B'_{[1:r]}\boldsymbol{x'} = C'_{[1:r]}\boldsymbol{x'} = 0^{m+m'}$.

4.1 Completeness

We prove the following lemma by showing that the completeness of the argument system Π_{R1CS^*} is directly inherited from that of the inner-product argument system Π_{IP}.

Lemma 3. *The argument system Π_{R1CS^*} has perfect completeness.*

Proof. Let $\mathbb{G}$ be a cyclic group of prime order q with $2(m+n)+2$ generators $\boldsymbol{G}, \boldsymbol{H} \in \mathbb{G}^{n+m}$ and $G, H \in \mathbb{G}$, and let $((T, A, B, C), (\boldsymbol{x}, \boldsymbol{x'}, \boldsymbol{y}, \boldsymbol{y'}, \eta)) \in \mathcal{R}_{\text{R1CS}^*}$. We show that for the inner-product instance $(\boldsymbol{G'}, \boldsymbol{H}, G, H, P, \omega)$ and witness $(\boldsymbol{u}, \boldsymbol{v}, \eta')$, as computed by the prover and verifier, it holds that $((\boldsymbol{G'}, \boldsymbol{H}, G, H, P, \omega), (\boldsymbol{u}, \boldsymbol{v}, \eta')) \in \mathcal{R}_{\text{IP}}$. That is, we have to show that $\omega = \langle \boldsymbol{u}, \boldsymbol{v} \rangle$ and that $P = \langle \boldsymbol{u}, \boldsymbol{G'} \rangle + \langle \boldsymbol{v}, \boldsymbol{H} \rangle + \eta' \cdot H$. First, $\boldsymbol{u}$ and $\boldsymbol{v}$ are defined by the prover as

$$\begin{aligned}
\boldsymbol{u} &= ((\boldsymbol{x'} || \boldsymbol{y}) + \delta^{-1} \cdot (\boldsymbol{x} || \boldsymbol{y'}) + \delta^2 \cdot \boldsymbol{\alpha}^n \;||\; (A\boldsymbol{z} + \delta^{-1} \cdot A\boldsymbol{z'}) \circ \boldsymbol{\gamma}^m - \boldsymbol{\beta}^m) \in \mathbb{Z}_q^{n+m} \\
\boldsymbol{v} &= (\boldsymbol{c} \circ \boldsymbol{\delta} \;||\; B\boldsymbol{z} - \boldsymbol{\alpha}^m + \delta^{-1} \cdot B\boldsymbol{z'}) \in \mathbb{Z}_q^{n+m} \ ,
\end{aligned}$$

and therefore

$$\begin{aligned}
\langle \boldsymbol{u}, \boldsymbol{v} \rangle &= \langle (\boldsymbol{x'} || \boldsymbol{y}) + \delta^{-1} \cdot (\boldsymbol{x} || \boldsymbol{y'}) + \delta^2 \cdot \boldsymbol{\alpha}^n, \boldsymbol{c} \circ \boldsymbol{\delta} \rangle \\
&\quad + \langle (A\boldsymbol{z} + \delta^{-1} \cdot A\boldsymbol{z'}) \circ \boldsymbol{\gamma}^m - \boldsymbol{\beta}^m, B\boldsymbol{z} - \boldsymbol{\alpha}^m + \delta^{-1} \cdot B\boldsymbol{z'} \rangle \ . \qquad (1)
\end{aligned}$$

Focusing on each of the two inner products on the right-hand side of Eq. (1), for the first one we obtain

$$\begin{aligned}
&\langle (\boldsymbol{x'} || \boldsymbol{y}) + \delta^{-1} \cdot (\boldsymbol{x} || \boldsymbol{y'}) + \delta^2 \cdot \boldsymbol{\alpha}^n, \boldsymbol{c} \circ \boldsymbol{\delta} \rangle \\
&\qquad = \langle (\delta^{-1}\boldsymbol{x} \;||\; \boldsymbol{y}), \boldsymbol{c} \circ \boldsymbol{\delta} \rangle + \langle (\boldsymbol{x'} \;||\; \delta^{-1}\boldsymbol{y'}), \boldsymbol{c} \circ \boldsymbol{\delta} \rangle + \delta^2 \cdot \langle \boldsymbol{\alpha}^n, \boldsymbol{c} \circ \boldsymbol{\delta} \rangle \\
&\qquad = \langle \boldsymbol{\delta}^{\boldsymbol{-1}} \circ \boldsymbol{z}, \boldsymbol{c} \circ \boldsymbol{\delta} \rangle + \delta \cdot \langle \boldsymbol{x'}, \boldsymbol{c}_{[1:r]} \rangle + \delta^{-1} \cdot \langle \boldsymbol{y'}, \boldsymbol{c}_{[r+1:n]} \rangle + \delta^2 \cdot \langle \boldsymbol{\alpha}^n, \boldsymbol{c} \circ \boldsymbol{\delta} \rangle \\
&\qquad = \langle \boldsymbol{z}, \boldsymbol{c} \rangle + \delta^{-1} \cdot \langle \boldsymbol{y'}, \boldsymbol{c}_{[r+1:n]} \rangle + \delta^2 \cdot \langle \boldsymbol{\alpha}^n, \boldsymbol{c} \circ \boldsymbol{\delta} \rangle \ , \qquad (2)
\end{aligned}$$

where Eq. (2) relies on the fact that $A_{[1:r]}\boldsymbol{x'} = B_{[1:r]}\boldsymbol{x'} = C_{[1:r]}\boldsymbol{x'} = 0^m$ that combined with $\boldsymbol{c} = \boldsymbol{\mu}^m A + \boldsymbol{\beta}^m B - \boldsymbol{\gamma}^m C$ implies

$$\langle \boldsymbol{x'}, \boldsymbol{c}_{[1:r]} \rangle = \langle \boldsymbol{x'}, \boldsymbol{\mu}^m A_{[1:r]} + \boldsymbol{\beta}^m B_{[1:r]} - \boldsymbol{\gamma}^m C_{[1:r]} \rangle = 0 \ .$$

For the second inner-product on the right-hand side of Eq. (1), we obtain

$$\begin{aligned}
&\langle (A\boldsymbol{z} + \delta^{-1} \cdot A\boldsymbol{z'}) \circ \boldsymbol{\gamma}^m - \boldsymbol{\beta}^m, B\boldsymbol{z} - \boldsymbol{\alpha}^m + \delta^{-1} \cdot B\boldsymbol{z'} \rangle \\
&\qquad = \langle \boldsymbol{\alpha}^m, \boldsymbol{\beta}^m \rangle + \delta^{-1} \cdot \boldsymbol{\gamma}^m (A\boldsymbol{z} \circ B\boldsymbol{z'} + B\boldsymbol{z} \circ A\boldsymbol{z'}) \\
&\qquad\quad - \delta^{-1} \cdot (\boldsymbol{\alpha}^m \circ \boldsymbol{\gamma}^m) A\boldsymbol{z'} - \delta^{-1} \boldsymbol{\beta}^m B\boldsymbol{z'} + \ \boldsymbol{\gamma}^m (A\boldsymbol{z} \circ B\boldsymbol{z}) \\
&\qquad\quad - (\boldsymbol{\alpha}^m \circ \boldsymbol{\gamma}^m) A\boldsymbol{z} - \boldsymbol{\beta}^m B\boldsymbol{z} + \delta^{-2} \cdot \boldsymbol{\gamma}^m (A\boldsymbol{z'} \circ B\boldsymbol{z'}) \\
&\qquad = \langle \boldsymbol{\alpha}^m, \boldsymbol{\beta}^m \rangle + \delta^{-1} \cdot (\boldsymbol{\gamma}^m C\boldsymbol{z'} - \boldsymbol{\mu}^m A\boldsymbol{z'} - \boldsymbol{\beta}^m B\boldsymbol{z'}) \\
&\qquad\quad + \boldsymbol{\gamma}^m C\boldsymbol{z} - \boldsymbol{\mu}^m A\boldsymbol{z} - \boldsymbol{\beta}^m B\boldsymbol{z} \qquad (3) \\
&\qquad = \langle \boldsymbol{\alpha}^m, \boldsymbol{\beta}^m \rangle - \delta^{-1} \cdot \langle \boldsymbol{z'}, \boldsymbol{c} \rangle - \langle \boldsymbol{z}, \boldsymbol{c} \rangle \\
&\qquad = \langle \boldsymbol{\alpha}^m, \boldsymbol{\beta}^m \rangle - \delta^{-1} \cdot \langle \boldsymbol{y'}, \boldsymbol{c}_{[r+1:n]} \rangle - \langle \boldsymbol{z}, \boldsymbol{c} \rangle \qquad (4)
\end{aligned}$$

where Eq. (3) follows from the facts that $A\boldsymbol{z} \circ B\boldsymbol{z} = C\boldsymbol{z}$, $A\boldsymbol{z'} \circ B\boldsymbol{z'} = 0^m$ and $A\boldsymbol{z} \circ B\boldsymbol{z'} + B\boldsymbol{z} \circ A\boldsymbol{z'} = C\boldsymbol{z'}$, and Eq. (4) follows from the fact that $A_{[1:r]}\boldsymbol{x'} = B_{[1:r]}\boldsymbol{x'} = C_{[1:r]}\boldsymbol{x'} = 0^m$ implies $\langle \boldsymbol{z'}, \boldsymbol{c} \rangle = \langle \boldsymbol{y'}, \boldsymbol{c}_{[r+1:n]} \rangle$. Combining Eq. (1), (2) and (4), we obtain

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \langle \boldsymbol{\alpha}^m, \boldsymbol{\beta}^m \rangle + \delta^2 \cdot \langle \boldsymbol{\alpha}^n, \boldsymbol{c} \circ \boldsymbol{\delta} \rangle = \omega \ .$$

Second, by the guarantee $T = \langle ((\boldsymbol{x} || \boldsymbol{y'}) \;||\; A\boldsymbol{z'}), \boldsymbol{G} \rangle + \langle (0^n \;||\; B\boldsymbol{z'}), \boldsymbol{H} \rangle + \eta \cdot H$, and by definitions of the group elements S and P as computed in the argument system, we obtain

$$\begin{aligned}
P &= \delta^{-1} \cdot T + S + \langle (\delta^2 \cdot \boldsymbol{\alpha}^n \;|| - \boldsymbol{\beta}^m), \boldsymbol{G'} \rangle + \langle (\boldsymbol{c} \circ \boldsymbol{\delta} \;|| - \boldsymbol{\alpha}^m), \boldsymbol{H} \rangle \\
&= \delta^{-1} \cdot \langle ((\boldsymbol{x} || \boldsymbol{y'}) \;||\; A\boldsymbol{z'}), \boldsymbol{G} \rangle + \delta^{-1} \cdot \langle (0^n \;||\; B\boldsymbol{z'}), \boldsymbol{H} \rangle + \delta^{-1} \cdot \eta \cdot H \\
&\quad + \langle ((\boldsymbol{x'} || \boldsymbol{y}) \;||\; A\boldsymbol{z}), \boldsymbol{G} \rangle + \langle (0^n \;||\; B\boldsymbol{z}), \boldsymbol{H} \rangle + r \cdot H \\
&\quad + \langle (\delta^2 \cdot \boldsymbol{\alpha}^n \;|| - \boldsymbol{\beta}^m), \boldsymbol{G'} \rangle + \langle (\boldsymbol{c} \circ \boldsymbol{\delta} \;|| - \boldsymbol{\alpha}^m), \boldsymbol{H} \rangle \\
&= \langle ((\boldsymbol{x'} || \boldsymbol{y}) + \delta^{-1} \cdot (\boldsymbol{x} || \boldsymbol{y'}) + \delta^2 \cdot \boldsymbol{\alpha}^n \;||\; (A\boldsymbol{z} + \delta^{-1} \cdot A\boldsymbol{z'}) \circ \boldsymbol{\gamma}^m - \boldsymbol{\beta}^m), \boldsymbol{G'} \rangle \\
&\quad + \langle (\boldsymbol{c} \circ \boldsymbol{\delta} \;||\; B\boldsymbol{z} - \boldsymbol{\alpha}^m + \delta^{-1} \cdot B\boldsymbol{z'}), \boldsymbol{H} \rangle + (r + \delta^{-1} \cdot \eta) \cdot H \\
&= \langle \boldsymbol{u}, \boldsymbol{G'} \rangle + \langle \boldsymbol{v}, \boldsymbol{H} \rangle + \eta' \cdot H \ .
\end{aligned}$$

Overall, we showed that $\omega = \langle \boldsymbol{u}, \boldsymbol{v} \rangle$ and $P = \langle \boldsymbol{u}, \boldsymbol{G'} \rangle + \langle \boldsymbol{v}, \boldsymbol{H} \rangle + \eta' \cdot H$, and therefore $((\boldsymbol{G'}, \boldsymbol{H}, G, H, P, \omega), (\boldsymbol{u}, \boldsymbol{v}, \eta')) \in \mathcal{R}_{\mathrm{IP}}$. Thus, the perfect completeness of the argument system Π_{IP} implies that the argument system Π_{R1CS^*} has perfect completeness. ■

4.2 Honest-Verifier Zero-Knowledge

We prove the following lemma by showing that the honest-verifier zero-knowledge of the argument system Π_{R1CS^*} is directly inherited from that of the inner-product argument system Π_{IP}.

Lemma 4. *The argument system Π_{R1CS^*} has perfect special honest-verifier zero-knowledge.*

Proof. Let $\mathcal{S}$ be the simulator that is defined as follows on input $((\boldsymbol{G}, \boldsymbol{H}, G, H), (T, A, B, C), \rho)$:

1. Uniformly sample $S \leftarrow \mathbb{G}$.
2. Parse $\rho = (\alpha, \beta, \gamma, \delta, \rho_{\mathsf{IP}})$, where $\alpha, \beta, \gamma, \delta \in \mathbb{Z}_q^*$ is the verifier's randomness for Step 2 of the argument system Π_{R1CS^*}, and ρ_{IP} is the verifier's randomness for Step 4 of the argument system Π_{R1CS^*} (i.e., ρ_{IP} is the verifier's randomness for the argument system Π_{IP}).
3. Compute

$$\begin{aligned}
\mu &= \alpha \cdot \gamma \in \mathbb{Z}_q \\
\boldsymbol{\delta} &= (\delta, \ldots, \delta, 1^{n-r}) \in \mathbb{Z}_q^n \\
\boldsymbol{G'} &= (G_1, \ldots, G_n, \gamma^{-1} \cdot G_{n+1}, \ldots, \gamma^{-m} \cdot G_{n+m}) \in \mathbb{G}^{n+m} \\
\boldsymbol{c} &= \boldsymbol{\mu}^m A + \boldsymbol{\beta}^m B - \boldsymbol{\gamma}^m C \in \mathbb{Z}_q^n \\
\omega &= \langle \boldsymbol{\alpha}^m, \boldsymbol{\beta}^m \rangle + \delta^2 \cdot \langle \boldsymbol{\alpha}^n, \boldsymbol{c} \circ \boldsymbol{\delta} \rangle \in \mathbb{Z}_q \\
P &= \delta^{-1} \cdot T + S + \langle (\delta^2 \cdot \boldsymbol{\alpha}^n \mid\mid -\boldsymbol{\beta}^m), \boldsymbol{G'} \rangle + \langle (\boldsymbol{c} \circ \boldsymbol{\delta} \mid\mid -\boldsymbol{\alpha}^m), \boldsymbol{H} \rangle \in \mathbb{G}
\end{aligned}$$

4. Invoke the zero-knowledge simulator of the argument system Π_{IP} on input $((\boldsymbol{G'}, \boldsymbol{H'}, G, H, P, \omega), \rho_{\mathsf{IP}})$ for obtaining a transcript $\mathsf{tr}_{\mathsf{IP}}$.
5. Output $(S, \alpha, \beta, \gamma, \delta, \mathsf{tr}_{\mathsf{IP}})$.

Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be any two algorithms as in Definition 3, fix any common-reference string $\sigma = (\boldsymbol{G}, \boldsymbol{H}, G, H)$, and fix any triplet $((T, A, B, C), (\boldsymbol{x}, \boldsymbol{x'}, \boldsymbol{y}, \boldsymbol{y'}, \eta), \rho)$ produced by $\mathcal{A}_1(\sigma)$ such that $((T, A, B, C), (\boldsymbol{x}, \boldsymbol{x'}, \boldsymbol{y}, \boldsymbol{y'}, \eta)) \in \mathcal{R}_{\mathrm{R1CS}^*}$. We need to show that, conditioned on any such fixed values, the distribution of the transcript produced by the simulator $\mathcal{S}$ is identical to the distribution of an honestly-generated transcript (thus, $\mathcal{A}_2$ will have no advantage in distinguishing the two cases – as required by Definition 3).

First, in both cases, the group element S is uniformly distributed: For the transcript produced by the simulator $\mathcal{S}$ this follows directly from the fact that $\mathcal{S}$ uniformly samples $S \leftarrow \mathbb{G}$, and for an honestly-generated transcript this follows from the fact that the honest prover uniformly samples $r \leftarrow \mathbb{Z}_q$ and computes

$$S = \langle ((\boldsymbol{x'} \| \boldsymbol{y}) \mid\mid A\boldsymbol{z}), \boldsymbol{G} \rangle + \langle (0^n \mid\mid B\boldsymbol{z}), \boldsymbol{H} \rangle + r \cdot H \ .$$

Second, in both cases, the values α, β, γ, and δ are uniquely determined by the verifier's randomness $\rho = (\alpha, \beta, \gamma, \delta, \rho_{\mathsf{IP}})$. Finally, in both cases the inner-product instance $(\boldsymbol{G'}, \boldsymbol{H}, G, H, P, \omega)$ is computed as the same deterministic function of the given generators $(\boldsymbol{G}, \boldsymbol{H}, G, H)$ and of $(S, \alpha, \beta, \gamma, \delta)$. Given that $((T, A, B, C), (\boldsymbol{x}, \boldsymbol{x'}, \boldsymbol{y}, \boldsymbol{y'}, \eta)) \in \mathcal{R}_{\mathrm{R1CS}^*}$, the perfect completeness of the argument system Π_{R1CS^*} guarantees that there exists a witness

$(\boldsymbol{u}, \boldsymbol{v}, \eta')$ (which would be computed by the prover in an honest execution as instructed) such that $((\boldsymbol{G'}, \boldsymbol{H}, G, H, P, \omega), (\boldsymbol{u}, \boldsymbol{v}, \eta')) \in \mathcal{R}_{\mathrm{IP}}$. Thus, the perfect special honest-verifier zero-knowledge of the argument system Π_{IP} guarantees that the transcript $\mathsf{tr}_{\mathsf{IP}}$ produced by its corresponding simulator is distributed identically to an honestly-generated transcript when conditioned on all previously-fixed values. ■

4.3 Witness-Extended Emulation

We prove that the argument system Π_{R1CS^*} provides computational witness-extended emulation based on the sufficient condition established by the general forking lemma of Bootle et al. (as discussed in Section 2.1 above). That is, we present a probabilistic polynomial-time algorithm that when provided with a transcript tree (of a suitable polynomial size) extracts either a valid witness or a non-trivial discrete-logarithm relation for $(\boldsymbol{G}, \boldsymbol{H}, G, H)$.

Lemma 5. *There exists a probabilistic polynomial-time algorithm* Ext *that, on input any* $(\boldsymbol{G}, \boldsymbol{H}, G, H) \in \mathbb{G}^{2(n+m)+2}$ *and any* $\mathcal{R}_{R1CS^*}$ *instance* (T, A, B, C) *together with any corresponding* $(n+1, m+1, m+1, 5, 2, 4, \ldots, 4, 5)$*-transcript tree of depth* $\log_2(n+m)+6$ *for the argument system* Π_{R1CS^*}*, produces either a witness* $(\boldsymbol{x}, \boldsymbol{x'}, \boldsymbol{y}, \boldsymbol{y'}, \eta)$ *such that* $((T, A, B, C), (\boldsymbol{x}, \boldsymbol{x'}, \boldsymbol{y}, \boldsymbol{y'}, \eta)) \in \mathcal{R}_{R1CS^*}$ *or a non-trivial discrete-logarithm relation for* $(\boldsymbol{G}, \boldsymbol{H}, G, H)$*.*

Proof. Let $m, r, n \in \mathbb{N}$ be such that $m \geq 1$, $1 \leq r \leq n$ and $m + n = 2^t$, let $\mathbb{G}$ be a cyclic group of prime order q, and let $\boldsymbol{G}, \boldsymbol{H} \in \mathbb{G}^{n+m}$ and $G, H \in \mathbb{G}$ be $2(n+m)+2$ generators. Then, any $(n+1, m+1, m+1, 5, 2, 4, \ldots, 4, 5)$-transcript tree of depth $\log_2(n+m)+6$ for an $\mathcal{R}_{\mathrm{R1CS}^*}$ instance (T, A, B, C) has the following form:

- The root of the tree is a group element $S \in \mathbb{G}$, and the first level consists of $n+1$ nodes corresponding to distinct values $\{\alpha_i\}_{i \in [n+1]}$.
- For each first-level node α_i, the second level consists of $m+1$ children corresponding to distinct values $\{\beta_{i,j}\}_{j \in [m+1]}$.
- For each second-level node $\beta_{i,j}$, the third level consists of $m+1$ children corresponding to distinct values $\{\gamma_{i,j,k}\}_{k \in [m+1]}$.
- For each third-level node $\gamma_{i,j,k}$, the forth level consists of 5 children corresponding to distinct values $\{\delta_{i,j,k,\ell}\}_{\ell \in [5]}$.
- Finally, each forth level node $\delta_{i,j,k,\ell}$ serves as the root of a $(2, 4, \ldots, 4, 5)$-transcript sub-tree of depth $\log_2(n+m)+2$ for the inner product argument with a corresponding instance $(\boldsymbol{G'_{i,j,k}}, \boldsymbol{H}, G, H, P_{i,j,k,\ell}, \omega_{i,j,k,\ell})$.

Our extractor Ext invokes the probabilistic polynomial-time extractor of the inner-product argument system (recall Lemma 2) on each of the $(2, 4, \ldots, 4, 5)$-transcript sub-trees (note that the number of such sub-trees is polynomial). For each such sub-tree it obtains either a witness $(\boldsymbol{u_{i,j,k,\ell}}, \boldsymbol{v_{i,j,k,\ell}}, \eta'_{i,j,k,\ell})$ such that $((\boldsymbol{G'_{i,j,k}}, \boldsymbol{H}, G, H, P_{i,j,k,\ell}, \omega_{i,j,k,\ell}),$ $(\boldsymbol{u_{i,j,k,\ell}}, \boldsymbol{v_{i,j,k,\ell}}, \eta'_{i,j,k,\ell})) \in \mathcal{R}_{\mathrm{IP}}$, or a non-trivial discrete-logarithm relation for $(\boldsymbol{G'_{i,j,k}},$ $\boldsymbol{H}, G, H)$. Any such relation yields a corresponding relation for $(\boldsymbol{G}, \boldsymbol{H}, G, H)$, and therefore for the remainder of the proof we assume that the extractor obtains a witness $(\boldsymbol{u_{i,j,k,\ell}}, \boldsymbol{v_{i,j,k,\ell}}, \eta'_{i,j,k,\ell})$ for each such sub-tree.

For every $i \in [n+1]$, $j, k \in [m+1]$ and $\ell \neq \ell' \in [5]$, note that for the $\mathcal{R}_{\mathrm{IP}}$ instances corresponding to the two paths $(\alpha_i, \beta_{i,j}, \gamma_{i,j,k}, \delta_{i,j,k,\ell})$ and $(\alpha_i, \beta_{i,j}, \gamma_{i,j,k}, \delta_{i,j,k,\ell'})$ it holds that

$$\begin{aligned}
S + \delta_{i,j,k,\ell}^{-1} \cdot T &= P_{i,j,k,\ell} - \langle ((\delta_{i,j,k,\ell}^{2} \cdot \boldsymbol{\alpha_i^n} \ || - \boldsymbol{\beta_{i,j}^m}), \boldsymbol{G'_{i,j,k}} \rangle \\
&\quad - \langle ((\boldsymbol{c_{i,j,k}} \circ \boldsymbol{\delta_{i,j,k,\ell}} \ || - \boldsymbol{\alpha_i^m}), \boldsymbol{H} \rangle \\
&= \langle \boldsymbol{u_{i,j,k,\ell}}, \boldsymbol{G'_{i,j,k}} \rangle + \langle \boldsymbol{v_{i,j,k,\ell}}, \boldsymbol{H} \rangle + \eta'_{i,j,k,\ell} \cdot H \\
&\quad - \langle ((\delta_{i,j,k,\ell}^{2} \cdot \boldsymbol{\alpha_i^n} \ || - \boldsymbol{\beta_{i,j}^m}), \boldsymbol{G'_{i,j,k}} \rangle - \langle ((\boldsymbol{c_{i,j,k}} \circ \boldsymbol{\delta_{i,j,k,\ell}} \ || - \boldsymbol{\alpha_i^m}), \boldsymbol{H} \rangle \\
S + \delta_{i,j,k,\ell'}^{-1} \cdot T &= P_{i,j,k,\ell'} - \langle ((\delta_{i,j,k,\ell'}^{2} \cdot \boldsymbol{\alpha_i^n} \ || - \boldsymbol{\beta_{i,j}^m}), \boldsymbol{G'_{i,j,k}} \rangle \\
&\quad - \langle ((\boldsymbol{c_{i,j,k}} \circ \boldsymbol{\delta_{i,j,k,\ell'}} \ || - \boldsymbol{\alpha_i^m}), \boldsymbol{H} \rangle \\
&= \langle \boldsymbol{u_{i,j,k,\ell'}}, \boldsymbol{G'_{i,j,k}} \rangle + \langle \boldsymbol{v_{i,j,k,\ell'}}, \boldsymbol{H} \rangle + \eta'_{i,j,k,\ell'} \cdot H \\
&\quad - \langle ((\delta_{i,j,k,\ell'}^{2} \cdot \boldsymbol{\alpha_i^n} \ || - \boldsymbol{\beta_{i,j}^m}), \boldsymbol{G'_{i,j,k}} \rangle - \langle ((\boldsymbol{c_{i,j,k}} \circ \boldsymbol{\delta_{i,j,k,\ell'}} \ || - \boldsymbol{\alpha_i^m}), \boldsymbol{H} \rangle \ ,
\end{aligned}$$

As $\delta_{i,j,k,\ell} \neq \delta_{i,j,k,\ell'}$, this provides two linearly-independent equations in the unknowns S and T, and enables to efficiently compute vectors

$$\begin{gathered}
\boldsymbol{s_{i,j,k,\ell,\ell',1}}, \boldsymbol{s'_{i,j,k,\ell,\ell',1}}, \boldsymbol{t_{i,j,k,\ell,\ell',1}}, \boldsymbol{t'_{i,j,k,\ell,\ell',1}} \in \mathbb{Z}_q^r \\
\boldsymbol{s_{i,j,k,\ell,\ell',2}}, \boldsymbol{s'_{i,j,k,\ell,\ell',2}}, \boldsymbol{t_{i,j,k,\ell,\ell',2}}, \boldsymbol{t'_{i,j,k,\ell,\ell',2}} \in \mathbb{Z}_q^{n-r} \\
\boldsymbol{s_{i,j,k,\ell,\ell',3}}, \boldsymbol{s'_{i,j,j',3}}, \boldsymbol{t_{i,j,k,\ell,\ell',3}}, \boldsymbol{t'_{i,j,k,\ell,\ell',3}} \in \mathbb{Z}_q^m \\
s''_{i,j,k,\ell,\ell'}, t''_{i,j,k,\ell,\ell'} \in \mathbb{Z}_q
\end{gathered}$$

such that

$$\begin{aligned}
S &= \langle \boldsymbol{s_{i,j,k,\ell,\ell',1}} || \boldsymbol{s_{i,j,k,\ell,\ell',2}} || \boldsymbol{s_{i,j,k,\ell,\ell',3}}, \boldsymbol{G} \rangle \\
&\qquad + \langle \boldsymbol{s'_{i,j,k,\ell,\ell',1}} || \boldsymbol{s'_{i,j,k,\ell,\ell',2}} || \boldsymbol{s'_{i,j,k,\ell,\ell',3}}, \boldsymbol{H} \rangle + s''_{i,j,k,\ell,\ell'} \cdot H \\
T &= \langle \boldsymbol{t_{i,j,k,\ell,\ell',1}} || \boldsymbol{t_{i,j,k,\ell,\ell',2}} || \boldsymbol{t_{i,j,k,\ell,\ell',3}}, \boldsymbol{G} \rangle \\
&\qquad + \langle \boldsymbol{t'_{i,j,k,\ell,\ell',1}} || \boldsymbol{t'_{i,j,k,\ell,\ell',2}} || \boldsymbol{t'_{i,j,k,\ell,\ell',3}}, \boldsymbol{H} \rangle + t''_{i,j,k,\ell,\ell'} \cdot H \ .
\end{aligned}$$

If these vectors are not identical for all (i, j, k, ℓ, ℓ'), then we obtain a non-trivial discrete-logarithm relation for $(\boldsymbol{G}, \boldsymbol{H}, H)$ (and thus a non-trivial discrete-logarithm relation for $(\boldsymbol{G}, \boldsymbol{H}, G, H)$). Therefore, for the remainder of the proof we assume that these vectors are identical for all (i, j, k, ℓ, ℓ'), and denote them by $\boldsymbol{s_1}, \boldsymbol{s'_1}, \boldsymbol{t_1}, \boldsymbol{t'_1} \in \mathbb{Z}_q^r$, $\boldsymbol{s_2}, \boldsymbol{s'_2}, \boldsymbol{t_2}, \boldsymbol{t'_2} \in \mathbb{Z}_q^{n-r}$, $\boldsymbol{s_3}, \boldsymbol{s'_3}, \boldsymbol{t_3}, \boldsymbol{t'_3} \in \mathbb{Z}_q^m$ and $s'', t'' \in \mathbb{Z}_q$. Equipped with this notation, we have

$$\begin{aligned}
S &= \langle \boldsymbol{s_1} || \boldsymbol{s_2} || \boldsymbol{s_3}, \boldsymbol{G} \rangle + \langle \boldsymbol{s'_1} || \boldsymbol{s'_2} || \boldsymbol{s'_3}, \boldsymbol{H} \rangle + s'' \cdot H \\
T &= \langle \boldsymbol{t_1} || \boldsymbol{t_2} || \boldsymbol{t_3}, \boldsymbol{G} \rangle + \langle \boldsymbol{t'_1} || \boldsymbol{t'_2} || \boldsymbol{t'_3}, \boldsymbol{H} \rangle + t'' \cdot H \ .
\end{aligned}$$

Next, for every $i \in [n+1]$, $j, k \in [m+1]$ and $\ell \in [5]$, the extracted witness $(\boldsymbol{u_{i,j,k,\ell}}, \boldsymbol{v_{i,j,k,\ell}}, \eta'_{i,j,k,\ell})$ satisfies

$$\begin{aligned}
&\langle \boldsymbol{u_{i,j,k,\ell}}, \boldsymbol{G'_{i,j,k}} \rangle + \langle \boldsymbol{v_{i,j,k,\ell}}, \boldsymbol{H} \rangle + \eta'_{i,j,k,\ell} \cdot H \\
&\quad = P_{i,j,k,\ell} \\
&\quad = S + \delta_{i,j,k,\ell}^{-1} \cdot T + \langle ((\delta_{i,j,k,\ell}^{2} \cdot \boldsymbol{\alpha_i^n} \ || - \boldsymbol{\beta_{i,j}^m}), \boldsymbol{G'_{i,j,k}} \rangle + \langle ((\boldsymbol{c_{i,j,k}} \circ \boldsymbol{\delta_{i,j,k,\ell}} \ || - \boldsymbol{\alpha_i^m}), \boldsymbol{H} \rangle \\
&\quad = \langle \boldsymbol{s_1} || \boldsymbol{s_2} || \boldsymbol{s_3}, \boldsymbol{G} \rangle + \langle \boldsymbol{s'_1} || \boldsymbol{s'_2} || \boldsymbol{s'_3} || \boldsymbol{H} \rangle + s'' \cdot H \\
&\qquad + \delta_{i,j,k,\ell}^{-1} \cdot \big(\langle \boldsymbol{t_1} || \boldsymbol{t_2} || \boldsymbol{t_3}, \boldsymbol{G} \rangle + \langle \boldsymbol{t'_1} || \boldsymbol{t'_2} || \boldsymbol{t'_3}, \boldsymbol{H} \rangle + t'' \cdot H \big) \\
&\qquad + \langle ((\delta_{i,j,k,\ell}^{2} \cdot \boldsymbol{\alpha_i^n} \ || - \boldsymbol{\beta_{i,j}^m}), \boldsymbol{G'_{i,j,k}} \rangle + \langle ((\boldsymbol{c_{i,j,k}} \circ \boldsymbol{\delta_{i,j,k,\ell}} \ || - \boldsymbol{\alpha_i^m}), \boldsymbol{H} \rangle \\
&\quad = \langle (((\boldsymbol{s_1} || \boldsymbol{s_2}) + \delta_{i,j,k,\ell}^{-1} \cdot (\boldsymbol{t_1} || \boldsymbol{t_2}) + \delta_{i,j,k,\ell}^{2} \cdot \boldsymbol{\alpha_i^n}) \ || \ (\boldsymbol{s_3} + \delta_{i,j,k,\ell}^{-1} \cdot \boldsymbol{t_3}) \circ \boldsymbol{\gamma_{i,j,k}^m} - \boldsymbol{\beta_{i,j}^m}), \boldsymbol{G'_{i,j,k}} \rangle \\
&\qquad + \langle (((\boldsymbol{s'_1} || \boldsymbol{s'_2}) + \delta_{i,j,k,\ell}^{-1} \cdot (\boldsymbol{t'_1} || \boldsymbol{t'_2}) + \boldsymbol{c_{i,j,k}} \circ \boldsymbol{\delta_{i,j,k,\ell}}) \ || \ \boldsymbol{s'_3} + \delta_{i,j,k,\ell}^{-1} \cdot \boldsymbol{t'_3} - \boldsymbol{\alpha_i^m}), \boldsymbol{H} \rangle \\
&\qquad + (s'' + \delta_{i,j,k,\ell}^{-1} \cdot t'') \cdot H \ .
\end{aligned}$$

Therefore,

$$\begin{aligned}
\boldsymbol{u_{i,j,k,\ell}} &= (((\boldsymbol{s_1}||\boldsymbol{s_2}) + \delta_{i,j,k,\ell}^{-1} \cdot (\boldsymbol{t_1}||\boldsymbol{t_2}) + \delta_{i,j,k,\ell}^{2} \cdot \boldsymbol{\alpha_i^n}) \;||\; (\boldsymbol{s_3} + \delta_{i,j,k,\ell}^{-1} \cdot \boldsymbol{t_3}) \circ \boldsymbol{\gamma_{i,j,k}^m} - \boldsymbol{\beta_{i,j}^m}) \\
\boldsymbol{v_{i,j,k,\ell}} &= (((\boldsymbol{s'_1}||\boldsymbol{s'_2}) + \delta_{i,j,k,\ell}^{-1} \cdot (\boldsymbol{t'_1}||\boldsymbol{t'_2}) + \boldsymbol{c_{i,j,k}} \circ \boldsymbol{\delta_{i,j,k,\ell}}) \;||\; \boldsymbol{s'_3} + \delta_{i,j,k,\ell}^{-1} \cdot \boldsymbol{t'_3} - \boldsymbol{\alpha_i^m}) \\
\eta'_{i,j,k,\ell} &= s'' + \delta_{i,j,k,\ell}^{-1} \cdot t'' \ ,
\end{aligned}$$

since otherwise we again obtain a non-trivial discrete-logarithm relation for $(\boldsymbol{G'_{i,j,k}}, \boldsymbol{H}, H)$ (and thus a non-trivial discrete-logarithm relation for $(\boldsymbol{G}, \boldsymbol{H}, G, H)$). Then, on the one hand

$$\begin{aligned}
\omega_{i,j,k,\ell} &= \langle \boldsymbol{u_{i,j,k,\ell}}, \boldsymbol{v_{i,j,k,\ell}} \rangle \\
&= \Big(\langle (\boldsymbol{t_1}||\boldsymbol{t_2}), (\boldsymbol{t'_1}||\boldsymbol{t'_2}) \rangle + \langle \boldsymbol{t_3} \circ \boldsymbol{\gamma_{i,j,k}^m}, \boldsymbol{t'_3} \rangle \Big) \cdot \delta_{i,j,k,\ell}^{-2} \\
&\quad + \Big(\langle (\boldsymbol{s_1}||\boldsymbol{s_2}), (\boldsymbol{t'_1}||\boldsymbol{t'_2}) \rangle + \langle (\boldsymbol{s'_1}||\boldsymbol{s'_2}), (\boldsymbol{t_1}||\boldsymbol{t_2}) \rangle + \langle \boldsymbol{t_2}, \boldsymbol{c_{i,j,k,[r+1:n]}} \rangle \\
&\qquad\qquad + \langle \boldsymbol{s'_3} - \boldsymbol{\alpha_i^m}, \boldsymbol{t_3} \circ \boldsymbol{\gamma_{i,j,k}^m} \rangle + \langle \boldsymbol{s_3} \circ \boldsymbol{\gamma_{i,j,k}^m} - \boldsymbol{\beta_{i,j}^m}, \boldsymbol{t'_3} \rangle \Big) \cdot \delta_{i,j,k,\ell}^{-1} \\
&\quad + \Big(\langle \boldsymbol{s_1}, \boldsymbol{s'_1} \rangle + \langle \boldsymbol{s_2}, (\boldsymbol{s'_2} + \boldsymbol{c_{i,j,k,[r+1:n]}}) \rangle + \langle \boldsymbol{t_1}, \boldsymbol{c_{i,j,k,[1:r]}} \rangle \\
&\qquad\qquad + \langle \boldsymbol{s_3} \circ \boldsymbol{\gamma_{i,j,k}^m}, \boldsymbol{s'_3} - \boldsymbol{\alpha_i^m} \rangle - \langle \boldsymbol{\beta_{i,j}^m}, \boldsymbol{s'_3} \rangle + \langle \boldsymbol{\alpha_i^m}, \boldsymbol{\beta_{i,j}^m} \rangle \Big) \\
&\quad + \Big(\langle \boldsymbol{s_1}, \boldsymbol{c_{i,j,k,[1:r]}} \rangle + \langle \boldsymbol{\alpha_i^n}, (\boldsymbol{t'_1}||\boldsymbol{t'_2}) \rangle \Big) \cdot \delta_{i,j,k,\ell} \\
&\quad + \Big(\langle \boldsymbol{\alpha_i^n}, (\boldsymbol{s'_1}||\boldsymbol{s'_2}) \rangle + \langle \alpha_i^r \cdot \boldsymbol{\alpha_i^{n-r}}, \boldsymbol{c_{i,j,k,[r+1:n]}} \rangle \Big) \cdot \delta_{i,j,k,\ell}^{2} \\
&\quad + \langle \boldsymbol{\alpha_i^r}, \boldsymbol{c_{i,j,k,[1:r]}} \rangle \cdot \delta_{i,j,k,\ell}^{3} \ ,
\end{aligned}$$

whereas on the other hand

$$\begin{aligned}
\omega_{i,j,k,\ell} &= \langle \boldsymbol{\alpha_i^m}, \boldsymbol{\beta_{i,j}^m} \rangle + \delta_{i,j,k,\ell}^{2} \cdot \langle \boldsymbol{\alpha_i^n}, \boldsymbol{c_{i,j,k}} \circ \boldsymbol{\delta_{i,j,k,\ell}} \rangle \\
&= \langle \boldsymbol{\alpha_i^m}, \boldsymbol{\beta_{i,j}^m} \rangle + \langle \alpha_i^r \cdot \boldsymbol{\alpha_i^{n-r}}, \boldsymbol{c_{i,j,k,[r+1:n]}} \rangle \cdot \delta_{i,j,k,\ell}^{2} + \langle \boldsymbol{\alpha_i^r}, \boldsymbol{c_{i,j,k,[1:r]}} \rangle \cdot \delta_{i,j,k,\ell}^{3} \ ,
\end{aligned}$$

which together imply

$$\begin{aligned}
0 &= \Big(\langle (\boldsymbol{t_1}||\boldsymbol{t_2}), (\boldsymbol{t'_1}||\boldsymbol{t'_2}) \rangle + \langle \boldsymbol{t_3} \circ \boldsymbol{\gamma_{i,j,k}^m}, \boldsymbol{t'_3} \rangle \Big) \cdot \delta_{i,j,k,\ell}^{-2} \\
&\quad + \Big(\langle (\boldsymbol{s_1}||\boldsymbol{s_2}), (\boldsymbol{t'_1}||\boldsymbol{t'_2}) \rangle + \langle (\boldsymbol{s'_1}||\boldsymbol{s'_2}), (\boldsymbol{t_1}||\boldsymbol{t_2}) \rangle + \langle \boldsymbol{t_2}, \boldsymbol{c_{i,j,k,[r+1:n]}} \rangle \\
&\qquad\qquad + \langle \boldsymbol{s'_3} - \boldsymbol{\alpha_i^m}, \boldsymbol{t_3} \circ \boldsymbol{\gamma_{i,j,k}^m} \rangle + \langle \boldsymbol{s_3} \circ \boldsymbol{\gamma_{i,j,k}^m} - \boldsymbol{\beta_{i,j}^m}, \boldsymbol{t'_3} \rangle \Big) \cdot \delta_{i,j,k,\ell}^{-1} \\
&\quad + \Big(\langle \boldsymbol{s_1}, \boldsymbol{s'_1} \rangle + \langle \boldsymbol{s_2}, (\boldsymbol{s'_2} + \boldsymbol{c_{i,j,k,[r+1:n]}}) \rangle + \langle \boldsymbol{t_1}, \boldsymbol{c_{i,j,k,[1:r]}} \rangle \\
&\qquad\qquad + \langle \boldsymbol{s_3} \circ \boldsymbol{\gamma_{i,j,k}^m}, \boldsymbol{s'_3} - \boldsymbol{\alpha_i^m} \rangle - \langle \boldsymbol{\beta_{i,j}^m}, \boldsymbol{s'_3} \rangle \Big) \\
&\quad + \Big(\langle \boldsymbol{s_1}, \boldsymbol{c_{i,j,k,[1:r]}} \rangle + \langle \boldsymbol{\alpha_i^n}, (\boldsymbol{t'_1}||\boldsymbol{t'_2}) \rangle \Big) \cdot \delta_{i,j,k,\ell} \\
&\quad + \langle \boldsymbol{\alpha_i^n}, (\boldsymbol{s'_1}||\boldsymbol{s'_2}) \rangle \cdot \delta_{i,j,k,\ell}^{2} \ .
\end{aligned}$$

For every $i \in [n+1]$ and $j, k \in [m+1]$, the right-hand side of above equation (when multiplied by $\delta_{i,j,k,\ell}^2$) is a polynomial of degree 4 in the variable δ. Since the above holds for 5 distinct values $\{\delta_{i,j,k,\ell}\}_{\ell \in [5]}$, it is the zero polynomial, and therefore its 5 coefficients are all zeros. That is,

$$\begin{aligned}
0 &= \langle (\boldsymbol{t_1} || \boldsymbol{t_2}), (\boldsymbol{t'_1} || \boldsymbol{t'_2}) \rangle + \langle \boldsymbol{t_3} \circ \boldsymbol{\gamma^m_{i,j,k}}, \boldsymbol{t'_3} \rangle && (5)\\
0 &= \langle (\boldsymbol{s_1} || \boldsymbol{s_2}), (\boldsymbol{t'_1} || \boldsymbol{t'_2}) \rangle + \langle (\boldsymbol{s'_1} || \boldsymbol{s'_2}), (\boldsymbol{t_1} || \boldsymbol{t_2}) \rangle + \langle \boldsymbol{t_2}, \boldsymbol{c_{i,j,k,[r+1:n]}} \rangle \\
&\qquad + \langle \boldsymbol{s'_3} - \boldsymbol{\alpha^m_i}, \boldsymbol{t_3} \circ \boldsymbol{\gamma^m_{i,j,k}} \rangle + \langle \boldsymbol{s_3} \circ \boldsymbol{\gamma^m_{i,j,k}} - \boldsymbol{\beta^m_{i,j}}, \boldsymbol{t'_3} \rangle && (6)\\
0 &= \langle \boldsymbol{s_1}, \boldsymbol{s'_1} \rangle + \langle \boldsymbol{s_2}, (\boldsymbol{s'_2} + \boldsymbol{c_{i,j,k,[r+1:n]}}) \rangle + \langle \boldsymbol{t_1}, \boldsymbol{c_{i,j,k,[1:r]}} \rangle \\
&\qquad + \langle \boldsymbol{s_3} \circ \boldsymbol{\gamma^m_{i,j,k}}, \boldsymbol{s'_3} - \boldsymbol{\alpha^m_i} \rangle - \langle \boldsymbol{\beta^m_{i,j}}, \boldsymbol{s'_3} \rangle && (7)\\
0 &= \langle \boldsymbol{s_1}, \boldsymbol{c_{i,j,k,[1:r]}} \rangle + \langle \boldsymbol{\alpha^n_i}, (\boldsymbol{t'_1} || \boldsymbol{t'_2}) \rangle && (8)\\
0 &= \langle \boldsymbol{\alpha^n_i}, (\boldsymbol{s'_1} || \boldsymbol{s'_2}) \rangle \ . && (9)
\end{aligned}$$

Similarly, since Eq. (9) holds for $n+1$ distinct values of α_i, then $\boldsymbol{s'_1} = 0^r$ and $\boldsymbol{s'_2} = 0^{n-r}$. Eq. (7) now becomes

$$0 = \langle (\boldsymbol{t_1} || \boldsymbol{s_2}), \boldsymbol{c_{i,j,k}} \rangle + \langle \boldsymbol{s_3} \circ \boldsymbol{\gamma^m_{i,j,k}}, \boldsymbol{s'_3} - \boldsymbol{\alpha^m_i} \rangle - \langle \boldsymbol{\beta^m_{i,j}}, \boldsymbol{s'_3} \rangle$$

and recalling that $\boldsymbol{c_{i,j,k}} = \boldsymbol{\mu^m_{i,j,k}} A + \boldsymbol{\beta^m_{i,j}} B - \boldsymbol{\gamma^m_{i,j,k}} C$ and $\mu_{i,j,k} = \alpha_i \cdot \gamma_{i,j,k}$ we obtain

$$\begin{aligned}
0 &= \langle (\boldsymbol{t_1} || \boldsymbol{s_2}), \boldsymbol{\mu^m_{i,j,k}} A + \boldsymbol{\beta^m_{i,j}} B - \boldsymbol{\gamma^m_{i,j,k}} C \rangle + \langle \boldsymbol{s_3} \circ \boldsymbol{\gamma^m_{i,j,k}}, \boldsymbol{s'_3} - \boldsymbol{\alpha^m_i} \rangle - \langle \boldsymbol{\beta^m_{i,j}}, \boldsymbol{s'_3} \rangle \\
&= \langle A \cdot (\boldsymbol{t_1} || \boldsymbol{s_2}) - \boldsymbol{s_3}, \boldsymbol{\alpha^m_i} \circ \boldsymbol{\gamma^m_{i,j,k}} \rangle + \langle B \cdot (\boldsymbol{t_1} || \boldsymbol{s_2}) - \boldsymbol{s'_3}, \boldsymbol{\beta^m_{i,j}} \rangle \\
&\quad - \langle C \cdot (\boldsymbol{t_1} || \boldsymbol{s_2}) - \boldsymbol{s'_3} \circ \boldsymbol{s_3}, \boldsymbol{\gamma^m_{i,j,k}} \rangle \ .
\end{aligned}$$

Applying Lemma 9 to the above, we obtain

$$\begin{aligned}
A \cdot (\boldsymbol{t_1} || \boldsymbol{s_2}) &= \boldsymbol{s_3} \\
B \cdot (\boldsymbol{t_1} || \boldsymbol{s_2}) &= \boldsymbol{s'_3} \\
C \cdot (\boldsymbol{t_1} || \boldsymbol{s_2}) &= \boldsymbol{s'_3} \circ \boldsymbol{s_3}
\end{aligned}$$

Letting $\boldsymbol{x} = \boldsymbol{t_1} \in \mathbb{Z}^r_q$, $\boldsymbol{y} = \boldsymbol{s_2} \in \mathbb{Z}^{n-r}_q$ and $\boldsymbol{z} = (\boldsymbol{x} || \boldsymbol{y}) \in \mathbb{Z}^n_q$, yields $(A\boldsymbol{z}) \circ (B\boldsymbol{z}) = C\boldsymbol{z}$ as required. Focusing now on the group elements S and T, so far we have established that they are of the following form:

$$\begin{aligned}
S &= \langle ((\boldsymbol{s_1} || \boldsymbol{y}) \ || \ A\boldsymbol{z}), \boldsymbol{G} \rangle + \langle (0^n \ || \ B\boldsymbol{z}), \boldsymbol{H} \rangle + s'' \cdot H \\
T &= \langle ((\boldsymbol{x} || \boldsymbol{t_2}) \ || \ \boldsymbol{t_3}), \boldsymbol{G} \rangle + \langle ((\boldsymbol{t'_1} || \boldsymbol{t'_2}) \ || \ \boldsymbol{t'_3}), \boldsymbol{H} \rangle + t'' \cdot H \ .
\end{aligned}$$

Examining the term $\langle \boldsymbol{s_1}, \boldsymbol{c_{i,j,k,[1:r]}} \rangle$ that appears in Eq. (8), given that $\boldsymbol{c_{i,j,k}} = \boldsymbol{\mu^m_{i,j,k}} A + \boldsymbol{\beta^m_{i,j}} B - \boldsymbol{\gamma^m_{i,j,k}} C$, then

$$\boldsymbol{c_{i,j,k,[1:r]}} = \boldsymbol{\mu^m_{i,j,k}} A_{[1:r]} + \boldsymbol{\beta^m_{i,j}} B_{[1:r]} - \boldsymbol{\gamma^m_{i,j,k}} C_{[1:r]} \ ,$$

where $A_{[1:r]}$, $B_{[1:r]}$ and $C_{[1:r]} \in \mathbb{Z}^{m \times r}_q$ denote the leftmost r columns of the matrices A, B and C, respectively. Eq. (8) is thus equivalent to

$$0 = \langle A_{[1:r]} \boldsymbol{s_1}, \boldsymbol{\alpha^m_i} \circ \boldsymbol{\gamma^m_{i,j,k}} \rangle + \langle B_{[1:r]} \boldsymbol{s_1}, \boldsymbol{\beta^m_{i,j}} \rangle - \langle C_{[1:r]} \boldsymbol{s_1}, \boldsymbol{\gamma^m_{i,j,k}} \rangle + \langle \boldsymbol{\alpha^n_i}, (\boldsymbol{t'_1} || \boldsymbol{t'_2}) \rangle \ .$$

Applying Lemma 9 once again, we obtain $A_{[1:r]} \boldsymbol{s_1} = B_{[1:r]} \boldsymbol{s_1} = C_{[1:r]} \boldsymbol{s_1} = 0^m$, $\boldsymbol{t'_1} = 0^r$ and $\boldsymbol{t'_2} = 0^{n-r}$. Letting $\boldsymbol{x'} = \boldsymbol{s_1} \in \mathbb{Z}^r_q$ we thus have $A_{[1:r]} \boldsymbol{x'} = B_{[1:r]} \boldsymbol{x'} = C_{[1:r]} \boldsymbol{x'} = 0^m$ as required.

Similarly, examining the term $\langle \boldsymbol{t_2}, \boldsymbol{c_{i,j,k,[r+1:n]}} \rangle$ that appears in Eq. (6) it holds that

$$\boldsymbol{c_{i,j,k,[r+1:n]}} = \boldsymbol{\mu^m_{i,j,k}} A_{[r+1:n]} + \boldsymbol{\beta^m_{i,j}} B_{[r+1:n]} - \boldsymbol{\gamma^m_{i,j,k}} C_{[r+1:n]} \ ,$$

where $A_{[r+1:n]}$, $B_{[r+1:n]}$ and $C_{[r+1:n]} \in \mathbb{Z}^{m \times (n-r)}_q$ denote the rightmost $n-r$ columns of the matrices A, B and C, respectively. Given that $\boldsymbol{s'_1} = \boldsymbol{t'_1} = 0^r$ and $\boldsymbol{s'_2} = \boldsymbol{t'_2} = 0^{n-r}$, it holds that $\langle(\boldsymbol{s_1}||\boldsymbol{s_2}),(\boldsymbol{t'_1}||\boldsymbol{t'_2})\rangle = 0$ and $\langle(\boldsymbol{s'_1}||\boldsymbol{s'_2}),(\boldsymbol{t_1}||\boldsymbol{t_2})\rangle = 0$, and therefore from Eq. (6) we obtain

$$\begin{aligned}
0 &= \langle \boldsymbol{t_2}, \boldsymbol{c_{i,j,k,[r+1:n]}}\rangle + \langle \boldsymbol{s'_3} - \boldsymbol{\alpha_i^m}, \boldsymbol{t_3} \circ \boldsymbol{\gamma_{i,j,k}^m}\rangle + \langle \boldsymbol{s_3} \circ \boldsymbol{\gamma_{i,j,k}^m} - \boldsymbol{\beta_{i,j}^m}, \boldsymbol{t'_3}\rangle \\
&= \langle A_{[r+1:n]}\boldsymbol{t_2}, \boldsymbol{\alpha_i^m} \circ \boldsymbol{\gamma_{i,j,k}^m}\rangle + \langle B_{[r+1:n]}\boldsymbol{t_2}, \boldsymbol{\beta_{i,j}^m}\rangle - \langle C_{[r+1:n]}\boldsymbol{t_2}, \boldsymbol{\gamma_{i,j,k}^m}\rangle \\
&\quad + \langle \boldsymbol{s'_3} \circ \boldsymbol{t_3} + \boldsymbol{s_3} \circ \boldsymbol{t'_3}, \boldsymbol{\gamma_{i,j,k}^m}\rangle - \langle \boldsymbol{t_3}, \boldsymbol{\alpha_i^m} \circ \boldsymbol{\gamma_{i,j,k}^m}\rangle - \langle \boldsymbol{t'_3}, \boldsymbol{\beta_{i,j}^m}\rangle \\
&= \langle A_{[r+1:n]}\boldsymbol{t_2} - \boldsymbol{t_3}, \boldsymbol{\alpha_i^m} \circ \boldsymbol{\gamma_{i,j,k}^m}\rangle + \langle B_{[r+1:n]}\boldsymbol{t_2} - \boldsymbol{t'_3}, \boldsymbol{\beta_{i,j}^m}\rangle \\
&\quad - \langle C_{[r+1:n]}\boldsymbol{t_2} - \boldsymbol{s'_3} \circ \boldsymbol{t_3} - \boldsymbol{s_3} \circ \boldsymbol{t'_3}, \boldsymbol{\gamma_{i,j,k}^m}\rangle \ .
\end{aligned}$$

Applying Lemma 9 once again, it holds that

$$\begin{aligned}
A_{[r+1:n]}\boldsymbol{t_2} &= \boldsymbol{t_3} \\
B_{[r+1:n]}\boldsymbol{t_2} &= \boldsymbol{t'_3} \\
C_{[r+1:n]}\boldsymbol{t_2} &= \boldsymbol{s'_3} \circ \boldsymbol{t_3} + \boldsymbol{s_3} \circ \boldsymbol{t'_3} = B\boldsymbol{z} \circ A_{[r+1:n]}\boldsymbol{t_2} + A\boldsymbol{z} \circ B_{[r+1:n]}\boldsymbol{t_2} \ .
\end{aligned}$$

Letting $\boldsymbol{y'} = \boldsymbol{t_2} \in \mathbb{Z}_q^{n-r}$ and $\boldsymbol{z'} = (\boldsymbol{x'}||\boldsymbol{y'}) \in \mathbb{Z}_q^n$, we then obtain $C_{[r+1:n]}\boldsymbol{y'} = (B\boldsymbol{z} \circ A_{[r+1:n]}\boldsymbol{y'}) + (A\boldsymbol{z} \circ B_{[r+1:n]}\boldsymbol{y'})$, and thus $C\boldsymbol{z'} = (B\boldsymbol{z} \circ A\boldsymbol{z'}) + (A\boldsymbol{z} \circ B\boldsymbol{z'})$ as required.

Finally, examining Eq. (5), given that $\boldsymbol{t'_1} = 0^r$ and $\boldsymbol{t'_2} = 0^{n-r}$, then for $m+1$ distinct values of $\gamma_{i,j,k}$ it holds that $0 = \langle \boldsymbol{t_3} \circ \boldsymbol{t'_3}, \boldsymbol{\gamma_{i,j,k}^m}\rangle$, and therefore $\boldsymbol{t_3} \circ \boldsymbol{t'_3} = 0^m$. That is, $(A_{[r+1:n]}\boldsymbol{y'}) \circ (B_{[r+1:n]}\boldsymbol{y'}) = 0^m$ and therefore $(A\boldsymbol{z'}) \circ (B\boldsymbol{z'}) = 0^m$. Letting $\eta = t''$, we obtain

$$T = \langle((\boldsymbol{x}||\boldsymbol{y'}) \ || \ A\boldsymbol{z'}), \boldsymbol{G}\rangle + \langle(0^n \ || \ B\boldsymbol{z'}), \boldsymbol{H}\rangle + \eta \cdot H$$

as required. ■

Acknowledgments

We thank Arash Afshar, Dan Boneh, Yi-Hsiu Chen, Yehuda Lindell, Peihan Miao and Samuel Ranellucci for many useful discussions throughout this work.

References

[AC20] Thomas Attema and Ronald Cramer. Compressed Σ-protocol theory and practical application to plug & play secure algorithmics. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 513–543, August 2020. doi:10.1007/978-3-030-56877-1_18.

[AFK23] Thomas Attema, Serge Fehr, and Michael Klooß. Fiat-Shamir transformation of multi-round interactive proofs (extended version). *Journal of Cryptology*, 36(4):36, October 2023. doi:10.1007/s00145-023-09478-y.

[BBB+18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pages 315–334. IEEE Computer Society Press, May 2018. doi:10.1109/SP.2018.00020.

[BCC+16] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 327–357, May 2016. doi:10.1007/978-3-662-49896-5_12.

[BCR+19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent succinct arguments for R1CS. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 103–128, May 2019. doi:10.1007/978-3-030-17653-2_4.

[Bün23] Benedikt Bünz. *Improving the Privacy, Scalability, and Ecological Impact of Blockchains*. PhD thesis, Stanford University, 2023. Available at https://cs.nyu.edu/~bb/papers/thesis.pdf.

[CHJ+22] Heewon Chung, Kyoohyung Han, Chanyang Ju, Myungsun Kim, and Jae Hong Seo. Bulletproofs+: Shorter proofs for a privacy-enhanced distributed ledger. *IEEE Access*, 10:42067–42082, 2022. doi:10.1109/ACCESS.2022.3167806.

[CHM+20] Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Psi Vesely, and Nicholas P. Ward. Marlin: Preprocessing zkSNARKs with universal and updatable SRS. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 738–768, May 2020. doi:10.1007/978-3-030-45721-1_26.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, August 1987. doi:10.1007/3-540-47721-7_12.

[GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 626–645, May 2013. doi:10.1007/978-3-642-38348-9_37.

[Gro16] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 305–326, May 2016. doi:10.1007/978-3-662-49896-5_11.

[Lin03] Yehuda Lindell. Parallel coin-tossing and constant-round secure two-party computation. *Journal of Cryptology*, 16(3):143–184, June 2003. doi:10.1007/s00145-002-0143-7.

[OB22] Alex Ozdemir and Dan Boneh. Experimenting with collaborative zk-SNARKs: Zero-knowledge proofs for distributed secrets. In Kevin R. B. Butler and Kurt Thomas, editors, *USENIX Security 2022: 31st USENIX Security Symposium*, pages 4291–4308. USENIX Association, August 2022. URL: https://www.usenix.org/system/files/sec22fall_ozdemir.pdf.

A From mIP Arguments to IP Arguments

In this section we show that an argument system for the multiplicative inner-product relation $\mathcal{R}_{\mathrm{mIP}}$ can be used to construct an argument system for the inner-product relation $\mathcal{R}_{\mathrm{IP}}$ (see Section 2.2 for the definitions of these two relations). Given an argument system Π_{mIP} for the relation $\mathcal{R}_{\mathrm{mIP}}$, consider the argument system Π_{IP} defined as follows:

The Argument System Π_{IP}

- **Public parameters:**
  1. Integer $d = 2^t \geq 1$.
  2. Cyclic group $\mathbb{G}$ of prime order q.
- **Inputs:**
  1. $\mathcal{P}$: Instance $(\boldsymbol{G}, \boldsymbol{H}, G, H, P, \omega) \in \mathbb{G}^{2d+3} \times \mathbb{Z}_q$ and witness $(\boldsymbol{u}, \boldsymbol{v}, \alpha) \in \mathbb{Z}_q^{2d+1}$.
  2. $\mathcal{V}$: Instance $(\boldsymbol{G}, \boldsymbol{H}, G, H, P, \omega) \in \mathbb{G}^{2d+3} \times \mathbb{Z}_q$.
- **Execution:**
  1. The verifier $\mathcal{V}$ samples $e \leftarrow \mathbb{Z}_q^*$, and sends e to the prover $\mathcal{P}$.
  2. Each party computes
  $$\begin{aligned} G' &= e \cdot G \\ P' &= P + \omega \cdot G' \end{aligned}$$
  3. The parties invoke Π_{mIP} with the instance $(\boldsymbol{G}, \boldsymbol{H}, G', H, P')$, where the prover $\mathcal{P}$ takes the role of the prover using the witness $(\boldsymbol{u}, \boldsymbol{v}, \alpha)$, and the verifier $\mathcal{V}$ takes the role of the verifier and then outputs its output.

In what follows we prove the completeness, zero-knowledge and witness-extended emulation properties of the argument system Π_{IP} based on the corresponding properties of the underlying argument system Π_{mIP}. Theorem 1 then follows by instantiating the underlying argument system Π_{mIP} with the argument system constructed by Chung et al. [CHJ+22].

Lemma 6. *Assuming that Π_{mIP} has perfect completeness, then Π_{IP} has perfect completeness.*

Proof. Let $\mathbb{G}$ be a cyclic group of prime order q, and let $((\boldsymbol{G}, \boldsymbol{H}, G, H, P, \omega), (\boldsymbol{u}, \boldsymbol{v}, \alpha)) \in \mathcal{R}_{\text{IP}}$. We show that, for any $e \in \mathbb{Z}_q^*$ chosen by the verifier, it holds that $((\boldsymbol{G}, \boldsymbol{H}, G', H, P'), (\boldsymbol{u}, \boldsymbol{v}, \alpha)) \in \mathcal{R}_{\text{mIP}}$ where $G' = e \cdot G$ and $P' = P + \omega \cdot G'$.

Given that $((\boldsymbol{G}, \boldsymbol{H}, G, H, P, \omega), (\boldsymbol{u}, \boldsymbol{v}, \alpha)) \in \mathcal{R}_{\text{IP}}$, then $P = \langle \boldsymbol{u}, \boldsymbol{G} \rangle + \langle \boldsymbol{v}, \boldsymbol{H} \rangle + \alpha \cdot H$ and $\omega = \langle \boldsymbol{u}, \boldsymbol{v} \rangle$, and therefore

$$\begin{aligned} P' &= P + \omega \cdot G' \\ &= \langle \boldsymbol{u}, \boldsymbol{G} \rangle + \langle \boldsymbol{v}, \boldsymbol{H} \rangle + \alpha \cdot H + \langle \boldsymbol{u}, \boldsymbol{v} \rangle \cdot G' \end{aligned}$$

This implies that $((\boldsymbol{G}, \boldsymbol{H}, G', H, P'), (\boldsymbol{u}, \boldsymbol{v}, \alpha)) \in \mathcal{R}_{\text{mIP}}$ as required. ■

Lemma 7. *Assuming that Π_{mIP} has perfect special honest-verifier zero-knowledge, then Π_{IP} has perfect special honest-verifier zero-knowledge.*

Proof. Let $\mathcal{S}$ be the simulator that is defined as follows on input $((\boldsymbol{G}, \boldsymbol{H}, G, H, P, \omega), \rho)$:

1. Parse $\rho = (e, \rho_{\mathsf{mIP}})$, where $e \in \mathbb{Z}_q^*$ is the verifier's randomness for Step 1 of the argument system Π_{IP}, and ρ_{mIP} is the verifier's randomness for Step 3 of the argument system Π_{IP} (i.e., ρ_{mIP} is the verifier's randomness for the argument system Π_{mIP}).
2. Compute $G' = e \cdot G$ and $P' = P + \omega \cdot G'$.
3. Invoke the zero-knowledge simulator of the argument system Π_{mIP} on input $((\boldsymbol{G}, \boldsymbol{H}, G', H, P'), \rho_{\mathsf{mIP}})$ for obtaining a transcript $\mathsf{tr}_{\mathsf{mIP}}$.
4. Output $(e, \mathsf{tr}_{\mathsf{mIP}})$.

Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be any two algorithms as in Definition 3, and fix any common-reference string $\sigma = (\mathbb{G}, G, q)$ and any triplet $((\boldsymbol{G}, \boldsymbol{H}, G, H, P, \omega), (\boldsymbol{u}, \boldsymbol{v}, \alpha), \rho)$ produced by $\mathcal{A}_1(\sigma)$ such that $((\boldsymbol{G}, \boldsymbol{H}, G, H, P, \omega), (\boldsymbol{u}, \boldsymbol{v}, \alpha)) \in \mathcal{R}_{\mathrm{IP}}$. We need to show that, conditioned on any such fixed values, the distribution of the transcript produced by the simulator $\mathcal{S}$ is identical to the distribution of an honestly-generated transcript (thus, $\mathcal{A}_2$ will have no advantage in distinguishing the two cases – as required by Definition 3).

First, in both cases, the value e is uniquely determined by the verifier's randomness $\rho = (e, \rho_{\mathsf{mIP}})$. Second, in both cases, the $\mathcal{R}_{\mathrm{mIP}}$ instance $(\boldsymbol{G}, \boldsymbol{H}, G', H, P')$ is computed as the same deterministic function of $(\boldsymbol{G}, \boldsymbol{H}, G, H, P, \omega)$ and e. Given that $((\boldsymbol{G}, \boldsymbol{H}, G, H, P, \omega), (\boldsymbol{u}, \boldsymbol{v}, \alpha)) \in \mathcal{R}_{\mathrm{IP}}$, the perfect completeness of the argument system Π_{IP} guarantees that $((\boldsymbol{G}, \boldsymbol{H}, G', H, P'), (\boldsymbol{u}, \boldsymbol{v}, \alpha)) \in \mathcal{R}_{\mathrm{mIP}}$. Thus, the perfect special honest-verifier zero-knowledge of the argument system Π_{mIP} guarantees that the transcript $\mathsf{tr}_{\mathsf{mIP}}$ produced by its corresponding simulator is distributed identically to an honestly-generated transcript when conditioned on all previously-fixed values. ■

Lemma 8. *Let $\mathbb{G}$ be a cyclic group of prime order q. Assume that Π_{mIP} is a $(2\mu + 1)$-move public-coin argument system, and for each $i \in [\mu]$ let $n_i = n_i(\kappa) \geq 1$ such that $\Pi_{i=1}^{\mu} n_i$ is polynomial in the security parameter $\kappa \in \mathbb{N}$. Assume further that there exists a probabilistic polynomial-time algorithm Ext_{mIP} that when given any $(n_1, \ldots, n_\mu)$-tree of accepting transcripts for an $\mathcal{R}_{mIP}$ instance $(\boldsymbol{G}, \boldsymbol{H}, G', H, P')$ always succeeds in extracting either a witness $(\boldsymbol{u}, \boldsymbol{v}, \alpha)$ such that $((\boldsymbol{G}, \boldsymbol{H}, G', H, P'), (\boldsymbol{u}, \boldsymbol{v}, \alpha)) \in \mathcal{R}_{mIP}$ or a non-trivial discrete-logarithm relation for $(\boldsymbol{G}, \boldsymbol{H}, G', H)$. Then, there exists a probabilistic polynomial-time algorithm Ext_{IP} that when given any $(2, n_1, \ldots, n_\mu)$-tree of accepting transcripts for an $\mathcal{R}_{IP}$ instance $(\boldsymbol{G}, \boldsymbol{H}, G, H, P, \omega)$ always succeeds in extracting either a witness $(\boldsymbol{u}, \boldsymbol{v}, \alpha)$ such that $((\boldsymbol{G}, \boldsymbol{H}, G, H, P, \omega), (\boldsymbol{u}, \boldsymbol{v}, \alpha)) \in \mathcal{R}_{IP}$ or a non-trivial discrete-logarithm relation for $(\boldsymbol{G}, \boldsymbol{H}, G, H)$.*

Proof. Any $(2, n_1, \ldots, n_\mu)$-transcript tree $\mathcal{T}_{\mathrm{IP}}$ for an $\mathcal{R}_{\mathrm{IP}}$ instance $(\boldsymbol{G}, \boldsymbol{H}, G, H, P, \omega)$ has the following form:

- The first level consists of 2 nodes corresponding to distinct values $e_1 \neq e_2$.
- Each second level node serves as the root of an $(n_1, \ldots, n_\mu)$-transcript sub-tree for an $\mathcal{R}_{\mathrm{mIP}}$ instance $(\boldsymbol{G}, \boldsymbol{H}, G'_i, H, P'_i)$, where $G'_i = e_i \cdot G$ and $P' = P + \omega \cdot G'_i$ for $i \in \{1, 2\}$.

Consider the extractor $\mathsf{Ext}_{\mathrm{IP}}$ that invokes the given extractor $\mathsf{Ext}_{\mathrm{mIP}}$ on each of the two $(n_1, \ldots, n_\mu)$-transcript sub-trees. By assumption, for each such sub-tree it obtains either a witness $(\boldsymbol{u_i}, \boldsymbol{v_i}, \alpha_i)$ such that $((\boldsymbol{G}, \boldsymbol{H}, G'_i, H, P'_i), (\boldsymbol{u_i}, \boldsymbol{v_i}, \alpha_i)) \in \mathcal{R}_{\mathrm{mIP}}$, or a non-trivial discrete-logarithm relation for $(\boldsymbol{G}, \boldsymbol{H}, G'_i, H)$. Any such relation yields a corresponding relation for $(\boldsymbol{G}, \boldsymbol{H}, G, H)$, and therefore for the remainder of the proof we assume that the extractor obtains a witness $(\boldsymbol{u_i}, \boldsymbol{v_i}, \alpha_i)$ for each $i \in \{1, 2\}$.

For each $i \in \{1, 2\}$ it thus holds that

$$P + \omega \cdot e_i \cdot G = P'_i = \langle \boldsymbol{u_i}, \boldsymbol{G} \rangle + \langle \boldsymbol{v_i}, \boldsymbol{H} \rangle + \langle \boldsymbol{u_i}, \boldsymbol{v_i} \rangle \cdot e_i \cdot G + \alpha_i \cdot H \tag{10}$$

and therefore

$$\begin{aligned} \omega \cdot (e_1 - e_2) \cdot G =\ & \langle \boldsymbol{u_1} - \boldsymbol{u_2}, \boldsymbol{G} \rangle + \langle \boldsymbol{v_1} - \boldsymbol{v_2}, \boldsymbol{H} \rangle \\ & + (\langle \boldsymbol{u_1}, \boldsymbol{v_1} \rangle \cdot e_1 - \langle \boldsymbol{u_2}, \boldsymbol{v_2} \rangle \cdot e_2) \cdot G + (\alpha_1 - \alpha_2) \cdot H \end{aligned} \tag{11}$$

If $\boldsymbol{u_1} \neq \boldsymbol{u_2}$ or $\boldsymbol{v_1} \neq \boldsymbol{v_2}$ or $\alpha_1 \neq \alpha_2$, then Eq. (11) yields a non-trivial discrete-logarithm relation for $(\boldsymbol{G}, \boldsymbol{H}, G, H)$. Therefore, for the remainder of the proof we assume that $\boldsymbol{u_1} = \boldsymbol{u_2}$, $\boldsymbol{v_1} = \boldsymbol{v_2}$ and $\alpha_1 = \alpha_2$, and denote these values by $\boldsymbol{u}$, $\boldsymbol{v}$ and α, respectively. Eq. (11) now simplifies to

$$\omega \cdot (e_1 - e_2) \cdot G = \langle \boldsymbol{u}, \boldsymbol{v} \rangle \cdot (e_1 - e_2) \cdot G$$

which implies that $\omega = \langle \boldsymbol{u}, \boldsymbol{v} \rangle$ since $e_1 \neq e_2$. Finally, from Eq. (10) we obtain

$$P = \langle \boldsymbol{u}, \boldsymbol{G} \rangle + \langle \boldsymbol{v}, \boldsymbol{H} \rangle + \alpha \cdot H \ .$$

This implies that $((\boldsymbol{G}, \boldsymbol{H}, G, H, P, \omega), (\boldsymbol{u}, \boldsymbol{v}, \alpha)) \in \mathcal{R}_{\mathrm{IP}}$ as required. ■

B An Auxiliary Lemma

Definition 5. Let $n_\alpha, n_\beta, n_\gamma \geq 1$. A set of triplets $\{(\alpha_i, \beta_{i,j}, \gamma_{i,j,k})\}_{i \in [n_\alpha], j \in [n_\beta], k \in [n_\gamma]}$ is *path distinct* if the following hold:

1. The values $\{\alpha_i\}_{i \in [n_\alpha]}$ are distinct.
2. For every $i \in [n_\alpha]$ the values $\{\beta_{i,j}\}_{j \in [n_\beta]}$ are distinct.
3. For every $i \in [n_\alpha]$ and $j \in [n_\beta]$ the values $\{\gamma_{i,j,k}\}_{k \in [n_\gamma]}$ are distinct.

Lemma 9. *Let $q \in \mathbb{N}$ be a prime number, let $m, n \geq 1$, $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{Z}_q^m$, $\boldsymbol{d} \in \mathbb{Z}_q^n$, $e \in \mathbb{Z}_q$ and let $f : \mathbb{Z}_q^3 \to \mathbb{Z}_q$ be defined as*

$$f(\alpha, \beta, \gamma) = \langle \boldsymbol{a}, \boldsymbol{\alpha}^m \circ \boldsymbol{\gamma}^m \rangle + \langle \boldsymbol{b}, \boldsymbol{\beta}^m \rangle + \langle \boldsymbol{c}, \boldsymbol{\gamma}^m \rangle + \langle \boldsymbol{d}, \boldsymbol{\alpha}^n \rangle + e.$$

If there exists a path-distinct set of $(n+1) \cdot (m+1)^2$ triplets $\{(\alpha_i, \beta_{i,j}, \gamma_{i,j,k})\}_{i \in [n+1], j,k \in [m+1]}$ such that $f(\alpha_i, \beta_{i,j}, \gamma_{i,j,k}) = 0$ for every $i \in [n+1]$ and $j, k \in [m+1]$, then $\boldsymbol{a} = \boldsymbol{b} = \boldsymbol{c} = 0^m$, $\boldsymbol{d} = 0^n$ and $e = 0$.

Proof. Fix any $i \in [n+1]$ and $j \in [m+1]$, and let

$$g_{i,j}(\gamma) = \langle \boldsymbol{a} \circ \boldsymbol{\alpha_i}^m + \boldsymbol{c}, \boldsymbol{\gamma}^m \rangle + \langle \boldsymbol{b}, \boldsymbol{\beta_{i,j}^m} \rangle + \langle \boldsymbol{d}, \boldsymbol{\alpha_i^n} \rangle + e \ .$$

Then $g_{i,j}$ is a polynomial of degree m, and it holds that $g(\gamma_{i,j,k}) = 0$ for every $k \in [m+1]$. Therefore, $g_{i,j}$ is the zero polynomial, and thus its $m+1$ coefficients are all zeros. That is, $\langle \boldsymbol{b}, \boldsymbol{\beta_{i,j}^m} \rangle + \langle \boldsymbol{d}, \boldsymbol{\alpha_i^n} \rangle + e = 0$ and $\boldsymbol{a} \circ \boldsymbol{\alpha_i^m} + \boldsymbol{c} = 0^m$. For each $i \in [n+1]$, let

$$h_i(\beta) = \langle \boldsymbol{b}, \boldsymbol{\beta}^m \rangle + \langle \boldsymbol{d}, \boldsymbol{\alpha_i^n} \rangle + e \ .$$

Then, h_i is a polynomial of degree m, and it holds that $h_i(\beta_{i,j}) = 0$ for every $j \in [m+1]$. Therefore, h_i is the zero polynomial, and thus its $m+1$ coefficients are all zeros. That is, $\boldsymbol{b} = 0^m$ and $\langle \boldsymbol{d}, \boldsymbol{\alpha_i^n} \rangle + e = 0$. Next, let

$$t(\alpha) = \langle \boldsymbol{d}, \boldsymbol{\alpha}^n \rangle + e \ ,$$

then t is a polynomial of degree n, and it holds that $g(\alpha_i) = 0$ for every $i \in [n+1]$. Therefore, t is the zero polynomial, and thus its $n+1$ coefficients are all zeros. That is, $\boldsymbol{d} = 0^n$ and $e = 0$. Finally, since for $n+1 \geq 2$ distinct values of α_i it holds that $\boldsymbol{a} \circ \boldsymbol{\alpha_i^m} + \boldsymbol{c} = 0^m$, then $\boldsymbol{a} = \boldsymbol{c} = 0^m$. ■