Efficient Inner-Product Argument from Compressed Σ-Protocols and Applications

Emanuele Scala[1(✉)] and Leonardo Mostarda[2]

[1] Computer Science, University of Camerino, Camerino, Italy
emanuele.scala@unicam.it
[2] Mathematics and Computer Science, University of Perugia, Perugia, Italy
leonardo.mostarda@unipg.it

Abstract. The Inner-Product Argument (IPA) is a subroutine of well-known zero-knowledge proof systems, such as Bulletproofs and Halo. These proof systems are then applied in large cryptographc protocols for anonymous and private transactions in the public blockchain. Despite its trustless nature and logarithmic communication efficiency, IPA suffers from low computational efficiency. While not specifically aimed at optimizing the IPA, Attema et al. propose the compressed Σ-protocol theory. Their intuition is simple: the prover provides an argument for a single committed vector to the verifier, whose commitment satisfies an arbitrary linear relation. We follow this intuition, but instead we provide an argument for two vectors committed under a single compact commitment, satisfying a linear form that is the inner-product relation. Hence, we propose the compressed Σ-protocol version of the original IPA, namely the compressed Σ-Inner-Product Argument (Σ-IPA). To this end, we prove security and provide a Σ-IPA that is complete and has soundness in standard DLOG setting. Finally, we conduct an efficiency analysis showing that our IPA reduces the computational complexity of prover and verifier algorithms by a factor of 2 compared to the original IPA.

Keywords: Inner-Product Argument · Σ-protocols · Zero-knowledge · Bulletproofs · Blockchain

1 Introduction

The Inner Product Argument (IPA) is an interactive proof between prover $\mathcal{P}$ and verifier $\mathcal{V}$, which engage an argument of knowledge of two vectors of scalars satisfying an *inner product relation*. Bootle et al. in [4] introduce the Inner Product Argument secure under discrete logarithm (DLOG) assumptions. Later, Bünz et al. in [7] propose Bulletproofs (BPs), that are arguments of knowledge for range proofs and general arithmetic circuits and use the IPA as a subroutine. The authors also optimize the IPA using a *folding strategy* with recursive composition, so that the resulting proof has logarithmic size as the size of the statement behind the inner product relation increases. Another interesting property of the BP argument of knowledge is that it does not come with trusted setup, which makes it attractive for trustless cryptographic protocols. From this

L. Barolli (Ed.): AINA 2024, LNDECT 202, pp. 326–337, 2024.
https://doi.org/10.1007/978-3-031-57916-5_28

result, BP has become widely adopted, especially in public blockchain cryptographic protocols, such as anonymous and private transactions. In these contexts, it is worth mentioning Quisquis [12] and Zether [6], which propose private transactions using BP to prove that amounts and balances are non-negative. Similarly, Lelantus [14] and Monero [1] hide the values of input coins, and prove that the outputs in a spend transaction are in the range of admissible values. However, the trade-off of gaining a trustless protocol results in worse scalability and higher fees. This is because trustless comes with non-constant size of the proofs and linear verification time. An amortization strategy is proposed in ZeroMT [10] and in MTproof [17]. Here, the transaction fees can be amortized by performing multiple transfers in a single transaction equipped with an aggregate *Zero-Knowledge proof* (ZK-proof), obtained by combining the aggregation technique of BP and Σ-protocol theory. Hence, the cost that a private transaction would have for a single transfer is now spread across multiple transfers. However, it turns out that the IPA subroutine still weighs significantly on the overall ZK-proof verification time. With the aim of optimizing the IPA's verification time, some proposals arises such as the one of Bowe et al. in [5]. Here, the authors introduce a new inner-product relation which leads to an argument for polynomial commitment evaluation. Along the same lines, Bünz et al. in [8] propose an amortized succinctness introducing accumulators for polynomial commitment schemes. Applying this technique to the IPA, the verifier asymptotically results in a logarithmic cost barring a single linear time check. In [18], the new inner-product relation is directly applied to the BP, landing into constant-size proofs. In another line of research, there are proposals for the IPA based on pairing-friendly groups, landing into the *inner-pairing product* [9,11,15]. However, they may result in expensive pairing operations once applied. While not specifically aimed at optimizing the IPA, closely related is the compressed Σ-protocol theory of Attema et al. [2,3]. In their notable works, the authors reconcile the BP compression mechanism with the Σ-protocol theory. This allows the design of RPs or arithmetic circuits within an established theory and in DLOG assumptions, with the same communication complexity of BPs. Their intuition is simple: a prover $\mathcal{P}$ provides a proof of knowledge of a committed vector $\mathbf{x}$ to a verifier $\mathcal{V}$, whose commitment satisfies an arbitrary and public linear relation $L(\mathbf{x})$. We follow this intuition, but in our case we want that $\mathcal{P}$ and $\mathcal{V}$ engage a proof of knowledge of two secret vectors committed under a single compact Pedersen commitment, satisfying a public linear form corresponding to the inner-product relation. Therefore, our specific instantiation of compressed Σ-protocol lies on the problem that the IPA tries to solve. Finally, with respect to the original IPA, we observe that our interactive proof algorithms require fewer expensive computations at the same communication complexity. This leads to a more computationally efficient IPA for prover and verifier algorithms.

Our Contribution. In this paper, we propose a new argument to prove that two secret vectors, which are committed under a single compact Pedersen commitment, satisfy the inner-product relation. To this end, we develop the compressed Σ-Inner-Product Argument (Σ-IPA) following the compressed Σ-protocols theory and the BP folding strategy. We prove security and provide a Σ-IPA that is complete and has soundness in standard DLOG setting. Finally, we conduct an efficiency analysis showing that our IPA reduces the computational complexity of prover and verifier algorithms by a factor of 2 compared to the BP's IPA. The paper is organized as follows: Sect. 2 presents related

work; Sect. 3 gives the cryptographic background; Sect. 4 provides an IPA with inefficient communication and related security proofs; Sect. 5 provides an IPA with logarithmic communication, improved computational cost and related security proofs; Sect. 6 analyzes the computational costs of the optimized IPA; Sect. 7 are the conclusions and future work.

2 Related Work

Bootle et al. [4] propose an inner product argument where the soundness relies on discrete logarithm assumption in prime order groups. The authors also present the notion of *witness extended emulation* from which the security property of soundness for the IPA is derived. Bünz et al. [7] optimizes the communication complexity of the IPA by a factor of 3, introducing a *folding strategy* in the recursive composition. The authors also propose Bulletproofs (BPs), zero-knowledge arguments for range proofs and general arithmetic circuits based on DLOG assumptions. Despite the proofs have logarithmic size, verification time is linear with respect to the length of the witnesses. Bowe et al. [5] propose an amortization strategy for the IPA verification time. This strategy follows a different inner-product relation, which turns out to be satisfied by a polynomial evaluation argument. Bünz et al. [8] establish a generalized result from the previous one. The authors demonstrate that any polynomial commitment scheme based on DLOG assumption has an accumulation scheme. With an accumulation scheme, the IPA verifier results in a logarithmic cost barring a final linear time opening check. Here, security is based on random oracle model. In our previous work [18], we apply the new relation of [5] to the IPA of BP and see that it could improve the communication complexity to a constant size. The security inherits that of [5], however the analysis requires further details. Another side of the research is devoted to the IPA based on pairing-friendly groups and universal setup. In that direction, Daza et al. [11] achieve logarithmic verification complexity in the circuit size based on the work of [4]. Bünz et al. [9] achieve a logarithmic-time verifier for a generalized IPA in pairing settings. Lee [15] proposes an argument of knowledge from inner-pairing products with a transparent setup, where the verifier has an asymptotic logarithmic time plus the cost for a number of pairings. However, pairing operations may result expensive once applied. Attema et al. [3] propose compressed Σ-protocols, reconciling the BPs compression mechanism with the theory of Σ-protocols and achieving the same communication complexity. The authors provide a general relation for the proof of knowledge of a vector commitment with arbitrary linear form openings. In our work, we essentially develop a compressed Σ-inner-product argument for the original inner-product relation of BPs, thus considering the security properties from the theory of compressed Σ-protocols.

3 Preliminaries

Notation. We denote with $\lambda \in \mathbb{N}$ the security parameter, PPT means probabilistic polynomial-time, and with $s \xleftarrow{\$} \mathcal{S}$ we indicate a random variable s uniformly sampled from the set $\mathcal{S}$. We consider cyclic groups of large prime order p denoted with $\mathbb{G}$, and

$|\mathbb{G}|$ is the order of the group. In every occurrence, g or h are generators of a cyclic group $\mathbb{G}$. We use the *group-generation* function $\mathcal{G}$ on input the security parameter 1^λ (written in unary) to generate the tuple $(\mathbb{G}, p, g) \leftarrow \mathcal{G}(1^\lambda)$. We use the *multiplicative notation* for group operations and scalar multiplications. Rings of integers modulo prime p are denoted with $\mathbb{Z}_p$, and the invertible elements of $\mathbb{Z}_p$ are in $\mathbb{Z}_p^*$. We denote vectors in bold, e.g., $\mathbf{a} = (a_1, ..., a_n) \in \mathbb{Z}_p^n$ is a vector of scalars and $\mathbf{g} = (g_1, ..., g_n) \in \mathbb{G}^n$ a vector of generators. We denote the *inner-product* between vectors of dimension n with $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^{n} a_i \cdot b_i \in \mathbb{Z}_p$. The *hadamard-product* between vectors of size n with $\mathbf{a} \circ \mathbf{b} = (a_1 \cdot b_1, ..., a_n \cdot b_n) \in \mathbb{Z}_p^n$. Let n be the size of a vector $\mathbf{s}$, then $\mathbf{s}_{lo} = (s_1, ..., s_k)$ of size k and $\mathbf{s}_{hi} = (s_{k+1}, ..., s_n)$ of size $n - k$ are vector slice operations.

Assumptions. We consider groups in which the *discrete logarithm problem* (DLOG problem) is computationally intractable. The following definition is for the *discrete logarithm assumption.*

Definition 1 (DLOG assumption). We say that the discrete-logarithm problem is hard relative to $\mathbb{G}$ if for all PPT algorithm $\mathcal{A}$ there exists a negligible function **negl** such that

$$Pr\left[\begin{matrix}(\mathbb{G}, p, g) \leftarrow \mathcal{G}(1^\lambda),\ y \xleftarrow{\$} \mathbb{G}; \\ x \in \mathbb{Z}_p \leftarrow \mathcal{A}(\mathbb{G}, p, g, y)\end{matrix} \quad : g^x = y \right] \leq \mathbf{negl}(\lambda)$$

An alternative definition is the *non-trivial discrete-logarithm relation* from [7].

Definition 2 (Non-trivial DLOG relation). For all PPT algorithm $\mathcal{A}$ and for all $k \geq 2$ there exists a negligible function **negl** such that

$$Pr\left[\begin{matrix}\mathbb{G} \leftarrow \mathcal{G}(1^\lambda),\ h_1, ..., h_k \xleftarrow{\$} \mathbb{G}; \\ x_1, ..., x_k \in \mathbb{Z}_p \leftarrow \mathcal{A}(\mathbb{G}, h_1, ..., h_k)\end{matrix} \quad : \begin{matrix}\exists x_1, ..., x_k \neq 0 \\ \wedge \prod_{i=1}^{k} h_i^{x_i} = 1\end{matrix} \right] \leq \mathbf{negl}(\lambda)$$

We say that there is a *non-trivial DLOG relation* between uniformly random group elements $h_1, ..., h_k$ when $\prod_{i=1}^{k} h_i^{x_i} = 1$ and each $x_1, ..., x_k \in \mathbb{Z}_p$ is non-zero. Thus, the DLOG relation assumption states that it is hard to find a non-trivial relation between randomly chosen group elements.

Commitments. We use the form of *Pedersen commitments* which can be defined over prime order cyclic groups $\mathbb{G}$. In particular, let g and h be two distinct generators and β a randomly chosen blinding factor, we compute a Pedersen commitment T to the value $t \in \mathbb{Z}_p$ as $T = g^t h^\beta$. We can also commit to multiple values at once using the *Pedersen vector commitment* variant. Here, values and generators are gathered in vectors of size n and the commitment is computed as $T = \mathbf{g}^{\mathbf{t}} h^\beta = \prod_{i=1}^{n} g_i^{t_i} \cdot h^\beta$. Pedersen commitments are computationally binding under DLOG assumption and perfectly hiding. Moreover, Pedersen commitments are *additive homomorphic*.

Zero-knowledge Relations. In the following, we give a definition for *zero-knowledge relation* and the relative notation.

Definition 3 (Zero-knowledge relation). A relation is a binary relation $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{W}$, where $\mathcal{X}$, $\mathcal{W}$, and $\mathcal{R}$ are finite sets. Elements $\mathbf{x} = \{x_1, ..., x_n\}$ of $\mathcal{X}$ are called instances and $\mathbf{w} = \{w_1, ..., w_n\}$ of $\mathcal{W}$ witnesses. A relation $\mathcal{R}$ formally specifies some statements as a function $f(\mathbf{x}, \mathbf{w})$ which are satisfied if and only if $(\mathbf{x}, \mathbf{w}) \in \mathcal{R}$. We use the notation

$$\mathcal{R} : \{(\mathbf{x}; \mathbf{w}) : f(\mathbf{x}, \mathbf{w})\}$$

to specify a relation for an interactive proof between prover $\mathcal{P}$ and verifier $\mathcal{V}$, where elements in $\mathbf{x}$ are public and are known to both $\mathcal{P}$ and $\mathcal{V}$, while those in $\mathbf{w}$ are only known to $\mathcal{P}$. We say that the relation is zero-knowledge if $\mathcal{P}$ convinces $\mathcal{V}$ that the statements are true, without revealing information about $\mathbf{w}$.

Interactive Proofs and Σ-Protocols. Let $\mathcal{R}$ be a relation and $\mathcal{L}$ the corresponding NP language such that $\mathcal{L} = \{x \mid \exists w : \ (x, w) \in \mathcal{R}\}$. An interactive proof between prover $\mathcal{P}$ and verifier $\mathcal{V}$ is a conversation where $\mathcal{P}$ tries to convince $\mathcal{V}$ that an instance x belongs to the language $\mathcal{L}$ according to the specified relation $\mathcal{R}$. Such conversation is called *transcript* of the interactive proof, and the verifier can `accept` or `reject` the transcript. When the verifier $\mathcal{V}$ outputs `accept` we call the conversation an *accepting transcript* for x. An interactive proof also requires the parties to execute some algorithms, we call these algorithms *interactive protocol algorithms*.

Σ-Protocols are a class of interactive proofs well established also in the context of zero-knowledge proofs. We now give a general definition for Σ-protocol:

Definition 4 (Σ-protocol). Let $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{W}$ be a binary relation. A Σ-protocol for $\mathcal{R}$ is an interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ where:

- $\mathcal{P}$ is an interactive protocol algorithm which takes as input an instance-witness pair $(x, w) \in \mathcal{R}$.
- $\mathcal{V}$ is an interactive protocol algorithm which takes as input an instance $x \in \mathcal{X}$ and outputs `accept` or `reject`.
- The interactive proof between $\mathcal{P}$ and $\mathcal{V}$ is structured so that it always works as follows:
  - $\mathcal{P}$ starts the protocol by computing a message a, called announcement, and sends a to $\mathcal{V}$;
  - Upon receiving $\mathcal{P}$'s announcement a, $\mathcal{V}$ chooses a challenge c at random from a finite challenge space $\mathcal{C}$, and sends c to $\mathcal{P}$.
  - Upon receiving $\mathcal{V}$'s challenge c, $\mathcal{P}$ computes a response z, and sends z to $\mathcal{V}$.
  - Upon receiving $\mathcal{P}$'s response z, $\mathcal{V}$ outputs `accept` or `reject`. The $\mathcal{V}$'s output must be computed strictly as a function of the instance x and the conversation (a, c, z). In particular, all $\mathcal{V}$ computations are completely deterministic except the random choice of the challenge.

We require that for all $(x, w) \in \mathcal{R}$, when $P(x, w)$ and $V(x)$ interact and follow the prescribed protocol, $V(x)$ always outputs `accept`.

Definition (4) highlights that Σ-protocols are 3-round protocols. When we execute multiple protocol instances, this leads to a multi-round Σ-protocol. Interactions between the $\mathcal{V}$ and an honest $\mathcal{P}$ produce accepting transcripts; this suggests how to verify the correctness of a conversation between $\mathcal{P}$ and $\mathcal{V}$. We can generalize the above concept with the definition of *perfect completeness* for any Σ-protocol.

Definition 5 (Perfect completeness). Let $(\mathcal{P}, \mathcal{V})$ be a Σ-protocol for relation $\mathcal{R}$. If the prover $\mathcal{P}$ follows the protocol then the verifier $\mathcal{V}$ will accept with probability 1.

Of course, non-accepting transcripts can occur if $\mathcal{V}$ interacts with a "dishonest" $\mathcal{P}^*$ who does not follow the protocol. To prevent this, we require the security properties of *special-soundness*. Furthermore, Σ-protocols are often required to have a large challenge space.

Definition 6 (k-Special-Soundness). Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a Σ-protocol for relation $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{W}$. We say that Π is k-special-sound if there exists a polynomial-time deterministic algorithm $\mathcal{E}$, called witness extractor, which is given as input an instance $x \in \mathcal{X}$ and k accepting transcripts (a, c_1, z_1), ...,(a, c_k, z_k) with a the common first $\mathcal{P}$'s message, $c_1, ..., c_k$ pairwise distinct $\mathcal{V}$'s challenges, $z_1, ..., z_k$ the final $\mathcal{P}$'s messages, and always outputs a witness $w \in \mathcal{W}$ satisfying $(x, w) \in \mathcal{R}$, i.e., w is a witness for x. When $k = 2$, it is simply said that Π is special-sound.

Suppose $\Pi = (\mathcal{P}, \mathcal{V})$ is a *special-sound* Σ-protocol that has a large challenge space, we say that Π acts as a *proof of knowledge*. There are alternative notions of special-soundness known in the literature: *knowledge-soundness* and *witness extended emulation* [3,16]. In the *knowledge-soundness* the difference is that the extractor only has *oracle access* to $\mathcal{P}^*$. In the *witness extended emulation*, the extractor with oracle access to $\mathcal{P}^*$ is also required to output a transcript that is indistinguishable from a conversation between $\mathcal{P}^*$ and an honest $\mathcal{V}$. In [4], it is shown that multi-round spacial-soundness implies *witness-extended emulation*. Essentially, a multi-round Σ-protocol is a $(2\mu+1)$-round interactive protocol where the verifier sends μ challenges. The special-soundness definition for multi-round Σ-protocol is a generalization of definition (6) and is given below.

Definition 7 ($(k_1, ..., k_\mu)$-Special-Soundness). Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a $(2\mu+1)$-round Σ-protocol with μ verifier's challenges and for relation $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{W}$. We say that Π is $(k_1, ..., k_\mu)$-special-sound if there exists a polynomial-time deterministic algorithm $\mathcal{E}$, called witness extractor, which is given as input an instance $x \in \mathcal{X}$ and a $(k_1, ..., k_\mu)$-tree of accepting transcripts and always outputs a witness $w \in \mathcal{W}$ satisfying $(x, w) \in \mathcal{R}$, i.e., w is a witness for x.

For the definition of $(k_1, ..., k_\mu)$-tree of transcripts we refer the reader to [3,4]. In this paper, we focus on $(k_1, ..., k_\mu)$-special-soundness protocols with some μ (challenges) and some set of k_i's (transcripts). Then, from [4] it follows that our protocols are *proof of knowledge*.

The following definition is useful when composing interactive proofs; the definition is revised from [2].

Definition 8 (Composable interactive proofs). Let Π_1 for relation $\mathcal{R}_1$ and Π_2 for relation $\mathcal{R}_2$ be two interactive proofs with $2\mu_1 + 1$ and $2\mu_2 + 1$ rounds respectively. Then, Π_1 and Π_2 are composable if, for an efficient computation ψ, the transcript $(\alpha_1, c_1, \alpha_2, ..., c_{\mu_1}, \alpha_{\mu_1+1})$ of Π_1 for statement x_1 is accepting if and only if the prover's final message α_{μ_1+1} is a witness for statement $x_2 = \psi(\alpha_1, c_1, \alpha_2, ..., c_{\mu_1})$ and $(x_2; \alpha_{\mu_1+1}) \in \mathcal{R}_2$. If the verifier of Π_2 accepts the proof for $\mathcal{R}_2$, then the composition $\Pi = \Pi_2 \diamond \Pi_1$ is accepted.

Moreover, if Π_1 is $\mathbf{k}_1$-special-sound and Π_2 is $\mathbf{k}_2$-special-sound, where $\mathbf{k}_1 = (k_1,...,k_{\mu_1})$ and $\mathbf{k}_2 = (k_1,...,k_{\mu_2})$, then $\Pi = \Pi_2 \diamond \Pi_1$ is $(\mathbf{k}_1,\mathbf{k}_2)$-special-sound [2].

Σ-protocols are commonly expected to be *public-coin* and *special Honest-Verifier Zero-Knowledge* (sHVZK). In public-coin Σ-protocols, all the messages sent by $\mathcal{V}$ to $\mathcal{P}$ are sampled uniformly at random and are independent of $\mathcal{P}$ messages. Moreover, $\mathcal{V}$ random choices are made public. In this paper, we deal with Σ-protocols that are not necessarily sHVZK and show only the interactive version of those protocols. Using the Fiat-Shamir heuristic [13], it is possible to convert an interactive proof into a *non-interactive* proof.

Inner-Product Argument relation. The *inner-product relation* follows the general definition (3). The standard IPA protocol is an interactive proof for the following relation, with $\mathbf{g},\mathbf{h},u,T$ public paramenters:

$$\mathcal{R}_{IPA} = \{(\mathbf{g},\mathbf{h} \in \mathbb{G}^n,\ u,T \in \mathbb{G}\ ;\ \mathbf{a},\mathbf{b} \in \mathbb{Z}_p^n)\ :\ T = \mathbf{g}^{\mathbf{a}}\mathbf{h}^{\mathbf{b}} \cdot u^{\langle \mathbf{a},\mathbf{b} \rangle}\} \tag{1}$$

where $\mathbf{g}$ and $\mathbf{h}$ are vectors of (independent) generators, u is group element, T a vector commitment and group element, $\mathbf{a}$ and $\mathbf{b}$ are vectors of scalar elements. The goal by the prover is to convince the verifier that he knows the two vectors $\mathbf{a}$ and $\mathbf{b}$ for the statement $T = \mathbf{g}^{\mathbf{a}}\mathbf{h}^{\mathbf{b}} \wedge c = \langle \mathbf{a},\mathbf{b} \rangle$, where c is the resulting value from the inner-product of the two vectors $\mathbf{a},\mathbf{b}$. Morever, this value c is given as a part of the vector commitment T by means of the additional group element u. We refer the reader to [7] for the relation where c is not given as a part of the vector commitment and call that relation $\mathcal{R}_{BP}$.

4 Sigma Inner-Product with Constant Rounds

In this section we present a 3-move IPA protocol denoted by Π_1 with inefficient communication complexity in protocol 1. Before engaging in Π_1, the prover and verifier run in a 2-move protocol Π_0 where, upon the verifier samples and sends $y \xleftarrow{\$} \mathbb{Z}_p^*$, both compute

$$T' = T \cdot u^{y \cdot c}$$

with $T = \mathbf{g}^{\mathbf{a}}\mathbf{h}^{\mathbf{b}}$, $c = \langle \mathbf{a},\mathbf{b} \rangle$ and generators $\mathbf{g},\mathbf{h}$ public parameters. Then, prover and verifier engage in Π_1 with the inputs substitution $u^y \rightarrow u$ and $T' \rightarrow T$, as specified in [7].

Protocol 1. Σ-IPA Π_1 for relation 1

input: $(\mathbf{g},\mathbf{h} \in \mathbb{G}^n, u,T \in \mathbb{G};\ \mathbf{a},\mathbf{b} \in \mathbb{Z}_p^n)$
  $\mathcal{P}$'s input: $(\mathbf{g},\mathbf{h},u,T,\mathbf{a},\mathbf{b})$
  $\mathcal{V}$'s input: $(\mathbf{g},\mathbf{h},u,T)$
output: $\mathcal{V}$ accepts or rejects

  $\mathcal{P}$ computes:
    $L = \mathbf{g}_{lo}^{\mathbf{a}_{hi}} \cdot \mathbf{h}_{hi}^{\mathbf{b}_{lo}} \cdot u^{\langle \mathbf{a}_{hi},\mathbf{b}_{lo} \rangle} \in \mathbb{G}$
    $R = \mathbf{g}_{hi}^{\mathbf{a}_{lo}} \cdot \mathbf{h}_{lo}^{\mathbf{b}_{hi}} \cdot u^{\langle \mathbf{a}_{lo},\mathbf{b}_{hi} \rangle} \in \mathbb{G}$
  end $\mathcal{P}$
  $\mathcal{P} \rightarrow \mathcal{V} : L,R$

$\mathcal{V} : x \xleftarrow{\$} \mathbb{Z}_p^*$
$\mathcal{V} \rightarrow \mathcal{P} : x$
$\mathcal{P}$ computes:
  $\mathbf{a}' = x \cdot \mathbf{a}_{lo} + \mathbf{a}_{hi} \in \mathbb{Z}_p^{n/2}$
  $\mathbf{b}' = \mathbf{b}_{lo} + x \cdot \mathbf{b}_{hi} \in \mathbb{Z}_p^{n/2}$
end $\mathcal{P}$
$\mathcal{P} \rightarrow \mathcal{V} : \mathbf{a}', \mathbf{b}'$
$\mathcal{V}$ computes and checks:
  $T' = L \cdot T^x \cdot R^{x^2} \in \mathbb{G}$
  $T' \stackrel{?}{=} \mathbf{g}_{lo}^{\mathbf{a}'} \cdot \mathbf{g}_{hi}^{x\mathbf{a}'} \cdot \mathbf{h}_{lo}^{x\mathbf{b}'} \cdot \mathbf{h}_{hi}^{\mathbf{b}'} \cdot u^{\langle \mathbf{a}', \mathbf{b}' \rangle}$
end $\mathcal{V}$

The protocol Π_1 is inefficient since the communication cost is only reduced by a factor of 2, i.e., the prover sends two vectors $\mathbf{a}', \mathbf{b}'$ of size $n/2$ with respect to the witness size of n. The protocol Π_1 is not required to be zero-knowledge, but it is complete and special-sound as stated in Theorem 1.

Theorem 1 (Σ-IPA). *The Σ-protocol Π_1 is a 3-move perfectly complete and 3-special-sound argument for the relation (1).*

Proof. **Completeness.** Following the definition (5) of *perfect completeness*, if the prover follows the protocol the proof is always accepted. Hence, it is sufficient to show that the following holds: $T' = \mathbf{g}_{lo}^{\mathbf{a}'} \cdot \mathbf{g}_{hi}^{x\mathbf{a}'} \cdot \mathbf{h}_{lo}^{x\mathbf{b}'} \cdot \mathbf{h}_{hi}^{\mathbf{b}'} \cdot u^{\langle \mathbf{a}', \mathbf{b}' \rangle} = L \cdot T^x \cdot R^{x^2}$.

3**-Special-Soundness.** We follow the definition (6) of *k-special-soundness*. Let $((L,R), x_i, (\mathbf{a}'_i, \mathbf{b}'_i,))$ for $i = 1, \ldots, 3$, be the accepting transcripts obtained by rewinding the prover three times after the prover sends L, R. Assuming $x_1, x_2, x_3 \in \mathbb{Z}_p$ are pairwise distinct challanges, we can find three values $v_1, v_2, v_3 \in \mathbb{Z}_p$ by inverting a Vandermonde matrix with non-zero determinant. Thus, it can be shown that for each $\mathbf{a}', \mathbf{b}'$ the tuples $\bar{\mathbf{a}} := \sum_{i=1}^{3} (v_i \mathbf{a}', v_i x_i \mathbf{a}')$, $\bar{\mathbf{b}} := \sum_{i=1}^{3} (v_i x_i \mathbf{b}', v_i \mathbf{b}')$ and $c = \sum_{i=1}^{3} v_i \cdot \langle \mathbf{a}', \mathbf{b}' \rangle$ are valid extracted witnesses for relation (1). Given that the statement $\mathbf{g}^{\bar{\mathbf{a}}} \mathbf{h}^{\bar{\mathbf{b}}} \cdot u^c = T$ holds, this completes the proof.

We now need one additional rewind for the 2-move protocol Π_0. By the soundness of protocol Π_1 the extractor can obtain witnesses $\mathbf{a}$ and $\mathbf{b}$ such that $T \cdot u^{y \cdot c} = \mathbf{g}^{\mathbf{a}} \mathbf{h}^{\mathbf{b}} u^{y \cdot \langle \mathbf{a}, \mathbf{b} \rangle}$. The extractor rewinds and runs the prover with a different challenge y', thus obtaining $T \cdot u^{y' \cdot c} = \mathbf{g}^{\mathbf{a}'} \mathbf{h}^{\mathbf{b}'} u^{y' \cdot \langle \mathbf{a}', \mathbf{b}' \rangle}$. Then, by combining the two equalities we get $\mathbf{g}^{\mathbf{a}-\mathbf{a}'} \mathbf{h}^{\mathbf{b}-\mathbf{b}'} u^{y \cdot \langle \mathbf{a}, \mathbf{b} \rangle - y' \cdot \langle \mathbf{a}', \mathbf{b}' \rangle} = u^{c \cdot (y - y')}$. Hence, either we have found a non-trivial DLOG relation from definition (2), or $\mathbf{a} = \mathbf{a}'$ and $\mathbf{b} = \mathbf{b}'$. In the latter case, it fallows that $u^{(y-y') \langle \mathbf{a}, \mathbf{b} \rangle} = u^{c \cdot (y - y')}$, which implies $c = \langle \mathbf{a}, \mathbf{b} \rangle$. Hence, we have found valid witnesses $\mathbf{a}$ and $\mathbf{b}$ for the statement $T = \mathbf{g}^{\mathbf{a}} \mathbf{h}^{\mathbf{b}} \wedge c = \langle \mathbf{a}, \mathbf{b} \rangle$.

5 Compressed Sigma Inner-Product with LOG Rounds

Following the definition (8) of composable proofs, it turns out that $\Pi_0 \diamond \Pi_1$ is $(2,3)$-special-sound for the Σ-IPA. However, the size of the proof is only reduced by a factor

of 2. We can further reduce the proof to a logarithmic size with respect to the witness, following the recursive composition of the *folding strategy* of BP. Furthermore, this is a similar strategy that leads the standard Σ-protocols to the *compressed* form of Attema et al. [2,3]. Thus, we can apply recursion multiple times until the two vectors $\mathbf{a}, \mathbf{b}$ have constant size. Such recursive composition is shown in protocol 2, denoted with Π_2, and considers vectors whose initial size is a power of two, i.e., $n = 2^\mu$ for some $\mu \in \mathbb{N}$.

Protocol 2. Compressed Σ-IPA Π_2 for relation 1

input: $(\mathbf{g}, \mathbf{h} \in \mathbb{G}^n, u, T \in \mathbb{G};\ \mathbf{a}, \mathbf{b} \in \mathbb{Z}_p^n)$
  $\mathcal{P}$'s input: $(\mathbf{g}, \mathbf{h}, u, T, \mathbf{a}, \mathbf{b})$
  $\mathcal{V}$'s input: $(\mathbf{g}, \mathbf{h}, u, T)$
output: $\mathcal{V}$ accepts or rejects

  $\mathcal{P}$ computes:
    $n = 2^\mu \in \mathbb{N}$
    $L_1 = \mathbf{g}_{lo}^{\mathbf{a}_{hi}} \cdot \mathbf{h}_{hi}^{\mathbf{b}_{lo}} \cdot u^{\langle \mathbf{a}_{hi}, \mathbf{b}_{lo} \rangle} \in \mathbb{G}$
    $R_1 = \mathbf{g}_{hi}^{\mathbf{a}_{lo}} \cdot \mathbf{h}_{lo}^{\mathbf{b}_{hi}} \cdot u^{\langle \mathbf{a}_{lo}, \mathbf{b}_{hi} \rangle} \in \mathbb{G}$
  end $\mathcal{P}$
  $\mathcal{P} \rightarrow \mathcal{V} : L_1, R_1$
  $\mathcal{V} : x_1 \xleftarrow{\$} \mathbb{Z}_p^*$
  $\mathcal{V} \rightarrow \mathcal{P} : x_1$
  $\mathcal{P}$ and $\mathcal{V}$ compute:
    $\mathbf{g}^{(2)} = \mathbf{g}_{lo} \circ \mathbf{g}_{hi}^{x_1} \in \mathbb{G}^{n/2}$
    $\mathbf{h}^{(2)} = \mathbf{h}_{lo}^{x_1} \circ \mathbf{h}_{hi} \in \mathbb{G}^{n/2}$
    $T_2 = L_1 \cdot T^{x_1} \cdot R_1^{x_1^2} \in \mathbb{G}$
  end $\mathcal{P}$ and $\mathcal{V}$
  $\mathcal{P}$ computes:
    $\mathbf{a}^{(2)} = x_1 \cdot \mathbf{a}_{lo} + \mathbf{a}_{hi} \in \mathbb{Z}_p^{n/2}$
    $\mathbf{b}^{(2)} = \mathbf{b}_{lo} + x_1 \cdot \mathbf{b}_{hi} \in \mathbb{Z}_p^{n/2}$
  end $\mathcal{P}$
  ⋮
  $\mathcal{P}$ computes:
    $\mathbf{a}' := \mathbf{a}^{(\mu-1)}$, $\mathbf{b}' := \mathbf{b}^{(\mu-1)} \in \mathbb{Z}_p^2$
    $\mathbf{g}' := \mathbf{g}^{(\mu-1)}$, $\mathbf{h}' := \mathbf{h}^{(\mu-1)} \in \mathbb{G}^2$
    $L_{\mu-1} = \mathbf{g}'^{\mathbf{a}'_{hi}}_{lo} \cdot \mathbf{h}'^{\mathbf{b}'_{lo}}_{hi} \cdot u^{\langle \mathbf{a}'_{hi}, \mathbf{b}'_{lo} \rangle} \in \mathbb{G}$
    $R_{\mu-1} = \mathbf{g}'^{\mathbf{a}'_{lo}}_{hi} \cdot \mathbf{h}'^{\mathbf{b}'_{hi}}_{lo} \cdot u^{\langle \mathbf{a}'_{lo}, \mathbf{b}'_{hi} \rangle} \in \mathbb{G}$
  end $\mathcal{P}$
  $\mathcal{P} \rightarrow \mathcal{V} : L_{\mu-1}, R_{\mu-1}$
  $\mathcal{V} : x_\mu \xleftarrow{\$} \mathbb{Z}_p^*$
  $\mathcal{V} \rightarrow \mathcal{P} : x_\mu$
  $\mathcal{P}$ and $\mathcal{V}$ compute:
    $g := g^{(\mu)} = \mathbf{g}'_{lo} \circ \mathbf{g}'^{x_\mu}_{hi} \in \mathbb{G}$
    $h := h^{(\mu)} = \mathbf{h}'^{x_\mu}_{lo} \circ \mathbf{h}'_{hi} \in \mathbb{G}$
    $T_\mu = L_{\mu-1} \cdot T_{\mu-1}^{x_\mu} \cdot R_{\mu-1}^{x_\mu} \in \mathbb{G}$
  end $\mathcal{P}$ and $\mathcal{V}$
  $\mathcal{P}$ computes:

$a := a^{(\mu)} = x_\mu \cdot \mathbf{a}'_{lo} + \mathbf{a}'_{hi} \in \mathbb{Z}_p$
$b := b^{(\mu)} = \mathbf{b}'_{lo} + x_\mu \cdot \mathbf{b}'_{hi} \in \mathbb{Z}_p$
end $\mathcal{P}$
$\mathcal{P} \rightarrow \mathcal{V} : a, b$
$\mathcal{V}$ computes and checks:
$c \stackrel{?}{=} \langle a, b \rangle$
$T_\mu \stackrel{?}{=} g^a \cdot h^b \cdot u^c$
end $\mathcal{V}$

The protocol Π_2 is efficient in terms of communication costs, that is, it is a $(2\mu+1)$-round protocol with $\mu = log_2(n)$ and n is the witness length. This implies the proof is logarithmic-sized considering that the prover exchanges $(L_1, R_1)...(L_{\mu-1}, R_{\mu-1})$ group element and 2 elements of $\mathbb{Z}_p$ with the verifier, while the verifier sends $log_2(n)$ elements of $\mathbb{Z}_p$. The protocol Π_2 is complete $(3,...,3)$-special-sound argument, always satisfying relation (1) as stated in theorem 2, but this time with halved-size witnesses in each round.

Theorem 2 (Compressed Σ-IPA). *The* Σ*-protocol* Π_2 *is a* $(2\mu+1)$*-move perfectly complete and* $(3,...,3)$*-special-sound argument for the relation (1).*

Proof. **Completeness.** It follows directly.

$(3,...,3)$-Special-Soundness. Following the definition (7), it can be shown that Π_2 is $(k_1,...,k_\mu)$-special-sound, where $k_i = 3$ for all $i \in [1,\mu]$. That is a generalization of the extractor analysis of Π_1, where now we have $\mu = log_2(n)$. It follows that we can use the same extractor, but this time it takes in total a 3-ary tree of accepting transcripts of depth $log_2(n)$, thus it runs in polynomial time in n. This complete the proof.

We now can replace the Π_1 protocol with Π_2 and compose $\Pi_0 \diamond \Pi_2$ to obtain a $(2\mu+3)$-move complete and $(2,3,...,3)$-special-sound IPA with logarithmic-size proof. From definition (8) of composable proofs, the composition is well-defined: the transcript $(y,(\mathbf{a},\mathbf{b}))$ for Π_0 on public input $(\mathbf{g},\mathbf{h},u,T,c)$, is accepting if and only if the tuple $(\mathbf{a},\mathbf{b})$ is a witness for statement $x = \psi(\mathbf{g},\mathbf{h},u,T,c,y) \mapsto (T \cdot u^{y \cdot c})$ with ψ an efficient computation, and thus $(x;(\mathbf{a},\mathbf{b})) \in \mathcal{R}_{BP}$. Similarly, Π_2 has an efficient computation ψ' and considers accepting transcripts if and only if the witnesses $(\mathbf{a},\mathbf{b})$ belong to a new relation $\mathcal{R}_{IPA}$ (1) with halved-size witnesses at each recursive step.

6 Efficiency Analysis

We now give an analysis of the efficiency of Σ-protocols Π_1 and Π_2 (protocol 1 and 2 respectively) with regards to communication and computational complexity. The protocol Π_1 has constant rounds but inefficient communication complexity. Indeed, prover sends to the verifier: 2 elements of $\mathbb{G}$ and 2 elements of $\mathbb{Z}_p^{n/2}$ of size $n/2$ with respect to the witness size n. The verifier sends only 1 element of $\mathbb{Z}_p^*$ to the prover. The protocol Π_2 reduces the communication complexity to logarithmic. Indeed, the prover sends to the verifier: $2 \cdot (\log(n) - 1)$ elements of $\mathbb{G}$ and 2 elements of $\mathbb{Z}_p$, with respect to the witness size n. The veirifer sends $\log(n)$ elements of $\mathbb{Z}_p^*$ to the prover. Finally, protocol

Π_2 improves the computational complexity of the BP's IPA by a factor of 2. Indeed, our prover and verifier algorithms execute $\sum_{j=1}^{\log_2(n)} \frac{n}{2^{j-1}}$ exponentiations and 0 inversions for computing the new generators $\mathbf{g}, \mathbf{h}$ in each j-th round. Instead, prover and verifier algorithms of the BP's IPA execute $\sum_{j=1}^{\log_2(n)} \frac{2n}{2^{j-1}}$ exponentiations and 2 inversions in each j-th round.

7 Conclusions and Future Work

In this paper, we propose compressed Σ-IPA, an argument of knowledge of two commited vectors following the compressed Σ-protocols theory and the original IPA relation. Our Σ-IPA maintains the same logarithmic communication complexity of the original IPA, while reducing the computational complexity by a factor of 2. As a future work, we want to introduce *accumulators* to enhance the overall verification time with logarithmic efficiency.

References

1. Alonso, K.M., et al. Zero to monero (2020)
2. Attema, T.: *Compressed Σ-protocol theory*. PhD thesis, Leiden University (2023)
3. Attema, T., Cramer, R.: Compressed-protocol theory and practical application to plug & play secure algorithmics. In: Annual International Cryptology Conference, pp. 513–543. Springer (2020)
4. Bootle, J., Cerulli, A., Chaidos, P., Groth, J., Petit, C.: Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 327–357. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_12
5. Bowe, S., Grigg, J., Hopwood, D.: Recursive proof composition without a trusted setup. Cryptology ePrint Archive (2019)
6. Bünz, B., Agrawal, S., Zamani, M., Boneh, D.: Zether: towards privacy in a smart contract world. In: Bonneau, J., Heninger, N. (eds.) FC 2020. LNCS, vol. 12059, pp. 423–443. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51280-4_23
7. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: 2018 IEEE Symposium on Security and Privacy (SP), pp. 315–334. IEEE (2018)
8. Bünz, B., Chiesa, A., Mishra, P., Spooner, N.: Proof-carrying data from accumulation schemes. Cryptology ePrint Archive (2020)
9. Bünz, B., Maller, M., Mishra, P., Tyagi, N., Vesely, P.: Proofs for inner pairing products and applications. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021. LNCS, vol. 13092, pp. 65–97. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92078-4_3
10. Corradini, F., Mostarda, L., Scala, E.: ZeroMT: multi-transfer protocol for enabling privacy in off-chain payments. In: Barolli, L., Hussain, F., Enokido, T. (eds.) AINA 2022. LNNS, vol. 450, pp. 611–623. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-99587-4_52
11. Daza, V., Ràfols, C., Zacharakis, A.: Updateable inner product argument with logarithmic verifier and applications. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020. LNCS, vol. 12110, pp. 527–557. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45374-9_18

12. Fauzi, P., Meiklejohn, S., Mercer, R., Orlandi, C.: Quisquis: a new design for anonymous cryptocurrencies. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019. LNCS, vol. 11921, pp. 649–678. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34578-5_23
13. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_12
14. Jivanyan, A.: Lelantus: towards confidentiality and anonymity of blockchain transactions from standard assumptions. In: IACR Cryptol. ePrint Arch., p. 373 (2019)
15. Lee, J.: Dory: efficient, transparent arguments for generalised inner products and polynomial commitments. In: Nissim, K., Waters, B. (eds.) TCC 2021. LNCS, vol. 13043, pp. 1–34. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-90453-1_1
16. Lindell: Parallel coin-tossing and constant-round secure two-party computation. J. Cryptol. **16**(3), 143–184 (2003). https://doi.org/10.1007/s00145-002-0143-7
17. Scala, E., Dong, C., Corradini,F., Mostarda, L.: Zero-knowledge multi-transfer based on range proofs and homomorphic encryption. In: International Conference on Advanced Information Networking and Applications, pp. 461–472. Springer (2023)
18. Scala, E., Mostarda, L.: Range proofs with constant size and trustless setup. In: International Conference on Advanced Information Networking and Applications, pp. 301–310. Springer (2023)