Range Proofs with Constant Size and Trustless Setup

Emanuele Scala(✉) and Leonardo Mostarda

Computer Science, University of Camerino, Camerino, Italy
{emanuele.scala,leonardo.mostarda}@unicam.it

Abstract. Range proofs are widely adopted in practice in many privacy-preserving cryptographic protocols in the public blockchain. The performances known in the literature for range proofs are logarithmic-sized proofs and linear verification time. In contexts where the proof verification is left to the ledger maintainers and proofs are stored in blocks, one might expect higher transaction fees and blockchain space when the size of the relation over the proof grows. With this paper, we improve Bulletproofs, a zero-knowledge argument of knowledge for range proofs, by modifying its Inner Product Argument (IPA) subroutine. In particular, we adopt a new relation from the polynomial commitment scheme of Halo, based on standard groups and assumptions (DLOG and RO) with a trustless setup. We design a *two-step reduction* algorithm and we obtain a constant number of two rounds in the IPA and a constant-sized proof composed of 5 $\mathbb{G}_1$ points and 2 $\mathbb{Z}_p$ scalars.

1 Introduction

Bootle et al. [3] develop an Inner Product Argument (IPA) system, in which computational soundness relies on discrete logarithm (DLOG) assumption in standard groups. The IPA consists of an argument of knowledge of the openings of Pedersen commitments satisfying an inner product relation. Bünz et al. [7] adopt the system of Bootle and propose Bulletproofs, a zero-knowledge proof system optimized for *range proofs*. Such proofs are useful in confidential transactions where a sender wants to prove that a value is in a particular range, without revealing the value to the receiver of the transaction. Bulletproofs optimizes the communication complexity through a logarithmic number of rounds in the IPA protocol used as a subroutine in the range proof. From these results, many cryptographic protocols have been applied with range proofs in blockchain contexts: Quisquis [13] and Zether [6] are privacy-preserving payment schemes using range proofs to prove that transfer amounts and balances over homomorphic encryptions are non-negatives; ZeroMT [10] extends the Zether's relation to many transfer amounts and balances, proving that a batch of aggregated values are non-negatives; Lelantus [14] and Monero [1] are private cryptocurrencies that hide the coin values through Pedersen commitments, and prove that output commitments in a spend transaction are in the range of admissible values.

L. Barolli (Ed.): AINA 2023, LNNS 655, pp. 301–310, 2023.
https://doi.org/10.1007/978-3-031-28694-0_28

However, due to the complexity of the IPA protocol, range proofs are logarithmically sized and proof verification time is linear in the bit length of the range. It follows that many works try to optimize the IPA protocol with solutions from standard groups or pairing-friendly groups, e.g., the inner-pairing products (a complete description can be found in the related works Sect. 4). In our work, we consider the optimizations proposed by Bowe et al. in [5], and we show how Halo's modified IPA can be applied to Bulletproofs, keeping the trustless setup and avoiding expensive pairing checks.

Our Contribution. We present a new *two-step reduction* algorithm for the IPA of Bulletproofs. The reduction exploits the structure of the polynomial commitment of Bulletproofs and a new IPA relation presented in Sect. 3. Surprisingly, this adaptation yields a constant number of two rounds in the IPA and a constant-sized proof composed of 5 $\mathbb{G}_1$ points and 2 $\mathbb{Z}_p$ scalars. As a part of the contribution, we implement and evaluate concretely our solution in the `arkworks` [2] Rust ecosystem.

2 Preliminaries

Groups. Let $(\mathbb{G}, p, g)$ be a description of a cyclic group $\mathbb{G}$, where p is the order of the group and is a prime number, $g \in \mathbb{G}$ is a generator of the group, we consider groups in which the *discrete logarithm problem* is computationally hard. In particular, we refer to the *Discrete Logarithm* (DLOG) and *Decisional Diffie-Hellman* (DDH) security assumptions for such groups.

Pedersen Commitment. A Pedersen commitment can be defined over a cyclic group $\mathbb{G}$ of prime order. A *binding* and *hiding* commitment for a message $m \in \mathbb{Z}_p$, from the set of integers modulo p, can be generated by applying the Commit function such that: $\text{Commit}(m;r) = (g^m h^r) \in \mathbb{G}$, where g and h are two distinct generators of the group $\mathbb{G}$, and r is a randomly chosen blinding factor. One variant is *Pedersen vector commitment* which allows multiple messages to be committed at once. Pedersen commitments are *homomorphic additive* when the group operator $\cdot$ is applied between commitments.

Zero-Knowledge Proofs. Let $\mathcal{R}$ be a relation between an instance x and a witness w such that $(x, w) \in \mathcal{R}$ and $\mathcal{L}$ be the language for that relation such that $\mathcal{L} = \{x \mid \exists w : \ (x, w) \in \mathcal{R}\}$. An interactive zero-knowledge proof is a protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ in which $\mathcal{P}$ convinces $\mathcal{V}$ that $x \in \mathcal{L}$ for the given relation $\mathcal{R}$ without revealing the witness. From the transcript of the protocol, the verifier can accept or reject the proof, which essentially reveals nothing beyond the validity of the proof. A proof system is *honest-verifier perfect zero-knowledge* (HVZK) if it has the properties of *perfect completeness*, *special soundness* and *honest-verifier perfect zero-knowledge*. The HVZK protocol is defined *public coin* if the messages from the verifier are uniformly random and are independent of the messages of the prover.

Bulletproofs Notation. The notations we use are those of Bulletproofs [7]. In summary, we denote:

- $\mathbb{Z}_p$ is a ring of integers modulo p prime and $\mathbb{G}_p$ is a cyclic group of prime order p.
- g and h are generators of $\mathbb{G}_p$.
- *Pedersen commitment* to the value a with blinding factor α is: $A = g^a h^\alpha$.
- Bold letters are vectors, e.g., $\mathbf{a} = (a_1, ..., a_n)$ with $\mathbf{a} \in \mathbb{Z}_p^n$.
- The *inner-product* of two vectors is $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^{n} a_i \cdot b_i$.
- *Pedersen vector commitment* to a vector $\mathbf{a} \in \mathbb{Z}_p^n$: $A = \mathbf{g}^\mathbf{a} = \prod_{i=1}^{n} g_i^{a_i}$ is *binding* (but not *hiding*) commitment, where $\mathbf{g} = (g_1, ..., g_n) \in \mathbb{G}^n$ is a vector of generators.
- *Vector polynomial* is defined as $p(X) = \sum_{i=0}^{d} \mathbf{p_i} \cdot X^i \in \mathbb{Z}_p^n[X]$, meaning that each coefficient of the polynomial p is a vector of field elements in $\mathbb{Z}_p^n$.

For the full notation of Bulletproofs, refer to Sect. 2.3 of [7].

Bulletproofs Proof System. Bulletproofs is zero-knowledge argument of knowledge in which a prover demonstrates that a value v is in a specific range, between zero and 2^{n-1}, where n is the range domain. Given as public parameters the tuple $(g, h, V = g^v h^\gamma)$, where g and h are generators of a group $\mathbb{G}_p$ and V is a Pedersen commitment of the value v, with a hiding factor from the randomness γ, the system ends by proving the equality $\hat{t} = \langle \mathbf{l}, \mathbf{r} \rangle$, i.e., that the inner-product of two committed vectors $\mathbf{l}$, $\mathbf{r}$ is a certain $\hat{t}$. In what follows, we first present the steps of the range proof prior to the inner-product. The prover generates two vectors $\mathbf{a}_L$ and $\mathbf{a}_R$ where $\langle \mathbf{a}_L, \mathbf{2}^n \rangle = v$ and $\mathbf{a}_R = \mathbf{a}_L - \mathbf{1}^n$, and commits to these vectors producing one commitment A. Further, the prover generates a second commitment S to blinding terms $\mathbf{s}_L$ and $\mathbf{s}_R$. The verifier generates and sends to the prover two random challenges y and z. The prover defines a polynomial $t(X)$ from the inner product of two vector polynomials $l(X)$ and $r(X)$, which in turn are derived from a linear combination of the vectors $\mathbf{a}_L$ and $\mathbf{a}_R$, the blinding vectors $\mathbf{s}_L$ and $\mathbf{s}_R$ and the two verifier challenges y and z. This results in a degree-two polynomial $t(X)$ with coefficients t_0, t_1 and t_2, where t_0 is the constant term, t_1 is the degree-one term and t_2 is the degree-two term. Then, the prover does not commit to the coefficient t_0, instead creates and sends to the verifier the commitments T_1 and T_2 to the coefficients t_1 and t_2. The prover convinces the verifier that it has the knowledge of the coefficients by proving that the polynomial $t(X)$ evaluates to a specific value $\hat{t}$ at a random point x. After receiving the challenge x, the prover sends to the verifier a blinding term τ_x for $\hat{t}$, a blinding factor μ for the commitments A and S and the two blinded vectors $\mathbf{l} = l(x)$ and $\mathbf{r} = r(x)$.

Prover		Verifier
$A, S \rightarrow$		
	$\leftarrow y,z$	
$T_1, T_2 \rightarrow$		
	$\leftarrow x$	
$\tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r} \rightarrow$		
		verify

Finally, the verifier can check the commitment V (which is public) of the value v, that the two vectors $\mathbf{l}$ and $\mathbf{r}$ are valid and that $\hat{t} = \langle \mathbf{l}, \mathbf{r} \rangle$. In order to reduce the size of the range proof from linear to logarithmic size in n (bits of the range), instead of transmitting the vectors $\mathbf{l}$ and $\mathbf{r}$, the prover and verifier engage in an Inner-Product-Argument (IPA) protocol with the two vectors becoming witnesses. In the next section, instead, we present how we modify the IPA protocol following the IPA relation of Halo [5], to reduce the size of the proof to a constant size.

3 Two-Step Reduction Inner-Product-Argument

Bulletproofs [7] implements an Inner-Product-Argument in which the prover proves the knowledge of two vectors $\mathbf{a}$ and $\mathbf{b}$ to the verifier for the relation:

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

Halo [5] introduce a new relation considering the following intuitions: by fixing the vector $\mathbf{b} = (1, x, x^2, ..., x^{d-1})$, where d is a fixed polynomial degree, we can claim that an evaluation v of a polynomial $t(x) = \langle \mathbf{a}, \mathbf{b} \rangle = v$ at random point x, where a is the vector with the coefficients of the polynomial t. With this variant, the vector $\mathbf{h}$ is no longer necessary, and we rewrite the new relation:

$$\{(\mathbf{g} \in \mathbb{G}^n,\ h \in \mathbb{G},\ u, T \in \mathbb{G},\ x, v \in \mathbb{Z}_p\ ;\ \mathbf{a} \in \mathbb{Z}_p^n,\ r \in \mathbb{Z}_p)\ :\ T = \mathbf{g}^{\mathbf{a}} h^r \cdot u^{\langle \mathbf{a}, \mathbf{b} \rangle}\} \tag{2}$$

where the additional generator h serves for the purpose of blinding the commitment T through the randomness r, x is the evaluation point used to construct the vector $\mathbf{b}$, and $v = \langle \mathbf{a}, \mathbf{b} \rangle$. Given the relation (2), in the following we design a *two-step reduction* IPA, adding the new relation to the range proof protocol of Bulletproofs.

From the definition of the polynomial $t(X) = t_0 + t_1 X + t_2 X^2$, we oberve that its evaluation at point x is

$$t(x) = \langle \mathbf{t}, \mathbf{b} \rangle = \hat{t}$$

where $\mathbf{t} = (t_0, t_1, t_2)$ is the vector of coefficients of $t(X)$ and $\mathbf{b} = (1, x, x^2)$. This means that if the prover proves the knowledge of $\mathbf{a} = \mathbf{t}$ and r for relation (2) also the relation (1) holds given

$$t(x) = \langle \mathbf{l}, \mathbf{r} \rangle = \hat{t}$$

with $\mathbf{a} = \mathbf{l}$ and $\mathbf{b} = \mathbf{r}$ for relation (1).

However, the length of $\mathbf{t}$ and $\mathbf{b}$ vectors is not a power of two, and so we cannot use them directly inside the Halo IPA. Then, we add an extra round into the protocol before the actual reduction step occurs.

In the first move, prover $\mathcal{P}$ and verifier $\mathcal{V}$ initialize a commitment

$$T' = T \cdot u^{\hat{t}}$$

where $T = \mathbf{g}^{\mathbf{t}} h^r$, with $\mathbf{t} = (t_0, t_1, t_2)$, and $u \in \mathbb{G}$ is a random group element sent by $\mathcal{V}$. Then, $\mathcal{P}$ and $\mathcal{V}$ engage in an IPA for relation (2). Assuming $d = 2$ the degree of $t(X)$, the protocol proceeds in $k = 2$ rounds, from one extra round (at $j = k - 1$) to one reduction step (at $j = 0$).

In round $j = 1$, the prover sets three vector:

$$\mathbf{t}^{(1)} = (t_0, t_1) \ , \ \mathbf{b}^{(1)} = (1, x) \ , \ \mathbf{g}^{(1)} = (g_0, g_1)$$

Then, the prover samples at random $l_1, r_1 \in \mathbb{Z}_p$ and computes and sends to the verifier:

$$L_1 = g_1^{t_0} \cdot h^{l_1} \cdot u^{t_0 \cdot x}$$

$$R_1 = g_0^{t_1} \cdot h^{r_1} \cdot u^{t_1 \cdot 1}$$

The verifier samples and sends to the prover a random challange $\mu_1 \in \mathbb{Z}_p$. Then, the prover computes $t^{(1)} \in \mathbb{Z}_p$, $b^{(1)} \in \mathbb{Z}_p$ and $g^{(1)} \in \mathbb{G}$, such that:

$$t^{(1)} = t_1 \cdot \mu_1^{-1} + t_0 \cdot \mu_1$$

$$b^{(1)} = 1 \cdot \mu_1^{-1} + x \cdot \mu_1$$

$$g^{(1)} = g_0^{\mu_1^{-1}} \cdot g_1^{\mu_1}$$

Now the prover prepares for the next round ($j = 0$) three other vectors (note that in this way an effective reduction step does not occur):

$$\mathbf{t}^{(0)} = (t^{(1)}, t_2) \ , \ \mathbf{b}^{(0)} = (b^{(1)}, x^2) \ , \ \mathbf{g}^{(0)} = (g^{(1)}, g_2)$$

In round $j = 0$, the prover samples at random $l_0, r_0 \in \mathbb{Z}_p$ and computes and sends to the verifier:

$$L_0 = g_2^{t^{(1)}} \cdot h^{l_0} \cdot u^{t^{(1)} \cdot x^2}$$

$$R_0 = g^{(1) t_2} \cdot h^{r_0} \cdot u^{t_2 \cdot b^{(1)}}$$

The verifier samples and sends to the prover a random challenge $\mu_0 \in \mathbb{Z}_p$. Then, the prover computes $t^{(0)} \in \mathbb{Z}_p$, $b^{(0)} \in \mathbb{Z}_p$ and $g^{(0)} \in \mathbb{G}$, such that:

$$t^{(0)} = t_2 \cdot \mu_0^{-1} + t^{(1)} \cdot \mu_0$$

$$b^{(0)} = b^{(1)} \cdot \mu_0^{-1} + x^2 \cdot \mu_0$$

$$g^{(0)} = g^{(1) \mu_0^{-1}} \cdot g_2^{\mu_0}$$

After this final round, the verifier computes:

$$T^{(0)} = \prod_{j=0}^{k-1}(L_j^{\mu_j^2}) \cdot T' \cdot \prod_{j=0}^{k-1}(R_j^{\mu_j^{-2}})$$

And the verifier wants to check that:

$$T^{(0)} \stackrel{?}{=} g^{(0)t^{(0)}} \cdot h^{r'} \cdot u^{(t^{(0)} \cdot b^{(0)})} \tag{3}$$

where $r' = \sum_{j=0}^{k-1}(l_j\mu_j^2) + r + \sum_{j=0}^{k-1}(r_j\mu_j^{-2})$.
Note that the verifier can compute $g^{(0)}$ and $b^{(0)}$ by itself from the following inner-products:

$$g^{(0)} = \langle \mathbf{s}, \mathbf{g} \rangle = \langle (\mu_0^{-1}, \mu_0), (g^{(1)}, g_2) \rangle \text{ with } g^{(1)} = \langle (\mu_1^{-1}, \mu_1), (g_0, g_1) \rangle \tag{4}$$

$$b^{(0)} = \langle \mathbf{s}, \mathbf{b} \rangle = \langle (\mu_0^{-1}, \mu_0), (b^{(1)}, x^2) \rangle \text{ with } b^{(1)} = \langle (\mu_1^{-1}, \mu_1), (1, x) \rangle \tag{5}$$

To check the equality (3), we first rewrite the right side:

$$T^{(0)} = (g^{(0)} \cdot u^{b^{(0)}})^{t^{(0)}} \cdot h^{r'} \tag{6}$$

Hence, the prover and verifier engage in a Schnorr protocol in which the prover proves to the verifier the knowledge of $t^{(0)}$ and r'.

The prover samples at random d,$s \in \mathbb{Z}_p$, computes and sends to the verifier a new commitment R:

$$R = (g^{(0)} \cdot u^{b^{(0)}})^d \cdot h^s$$

The verifier samples and sends to the prover a random $c \in \mathbb{Z}_p$.
The prover computes and sends to the verifier the scalars z_1 and z_2:

$$z_1 = t^{(0)}c + d$$

$$z_2 = r'c + s$$

Finally, the verifier accepts or rejects the proof if and only if:

$$T^{(0)c} \cdot R \stackrel{?}{=} (g^{(0)} \cdot u^{b^{(0)}})^{z_1} \cdot h^{z_2}$$

Two-Step Reduction IPA Proof Size. A zero-knowledge proof is composed of all the scalars (elements in $\mathbb{Z}_p$), and elliptic curve points (elements in $\mathbb{G}$) that the prover forwards to the verifier. Our *two-step reduction* IPA generates two collections of elliptic curve points (L_1, L_0) and (R_1, R_0) at each j-th round, one group element R and two scalar field elements z_1, z_2 in the Schnorr protocol. The proof size is constant given the constant number of rounds in the IPA and the total proof consists of 5 $\mathbb{G}_1$ points and 2 $\mathbb{Z}_p$ scalars.

4 Related Work

Bowe et al. [5] propose Halo, a recursive proof composition from the notion of Incrementally Verifiable Computation (IVC), i.e. a method to inductively prove within a single proof the validity of past proofs. The recursion is made via a cycle of normal prime-order elliptic curves, such that proofs over one curve can verify proofs over the other curve. An interesting technique is the *amortized succinctness* for *polynomial commitments*: by the structure of the two vectors behind the IPA, the linear-time work of the verifier is amortized across many proofs. This is done by an untrusted third party who executes the linear-time operations for each step proof and then proves the correctness of a batch of that proofs to the verifier. The verifier performs the same operations once for the entire batch. This batch of proofs is handled via an *accumulator* which does not grow in size with each step proof. With this amortization strategy, the IPA verifier results in a logarithmic cost barring the single linear time check.

Bünz et al. [8] establish an exciting result that generalizes the Halo's recursive composition to a class of non-interactive arguments which do not necessarily have succinct verification. The authors provide theoretical efficiency and security proofs for constructing *accumulation schemes* for any SNARK, which yields to Proof-Carrying-Data (PCD) scheme. Moreover, the authors prove a second theorem stating that if the SNARK verifier is succinct except for a specific predicate, and has an accumulation scheme for that predicate, it is possible to derive an accumulation scheme for the SNARK. Further, the authors prove that two polynomial commitment schemes have accumulation schemes in Random Oracle: (i) PC_{DL}, polynomial commitment scheme based on discrete logarithm assumption; (ii) PC_{AGM}, polynomial commitment based on knowledge assumption in bilinear groups. From this follows an interesting open question of whether constructions exist in the standard assumption instead of in the "trivial" knowledge assumption. From the efficiency perspective, PC_{DL} achieves an asymptotic logarithmic cost to check accumulation steps and a linear cost in the polynomial degree during the final opening check. Instead, PC_{AGM} has an asymptotic linear cost to check accumulation steps, while only one pairing is required in the final check.

Xiong et al. [19] propose VERI-ZEXE, an improvement of Zexe's Decentralized Private Computation (DPC) scheme [4], translating the circuit-specific trusted setup into a universal setup where a structured reference string (SRS) is reused for different circuits. Universal SNARKs built on Polynomial Interactive Oracle Proofs (PIOP) are often instantiated with pairing-based Polynomial Commitment Schemes (PCS) that require expensive pairing operations in the verifier circuit. To lighten the cost of pairing checks, VERI-ZEXE relies on the generalized *accumulation scheme* of Bünz et al. in PCD [8], designing a *two-step* IVC. Hence, with this algorithm, their goal is to delay the final bilinear pairing check, by attaching $2\mathbb{G}_1$ points to the transaction validity proof to be verified by the ledger maintainers.

Daza et al. [11] propose an optimization of the IPA protocol of Bootle et al. [3] on the verifier side. In particular, the authors try to achieve a logarithmic

verification complexity in the circuit size. Their scheme is based on bilinear groups, secure under the standard assumption and Random Oracle, with an updatable and universal setup.

Bünz et al. [9] present a Generalized Inner Product Argument (GIPA) in pairing-based groups. With GIPA, the authors achieve a logarithmic-time verifier for a polynomial commitment scheme with a universal setup. This comes at the cost of square root complexity for prover time bounded to the polynomial degree and square root SRS size.

Lee [15] proposes Dory, an argument of knowledge system from inner-pairing products with a transparent setup. This result is established in the standard SXDH (Symmetric eXternal Diffie-Hellman) assumption which implies DLOG. The verifier work has an asymptotic logarithmic cost of n multi-exponentiation with respect to the length n of the IPA vectors, plus a constant number of pairings.

5 Implementation and Evaluation

In this section, we present an implementation and proof size evaluation of our *two-step reduction* IPA presented in Sect. 3, compared to the IPA of Bulletproofs [7]. The source code, available on GitHub [12], is written in Rust and is based on the `arkworks` [2] libraries. The elliptic curve we use for all group operations is the Barreto-Naehrig curve (BN-254). In Table 1, we report the evaluations in bytes of the size of the proofs, considering a fixed range domain of $n = 16$ bits and a variable number of aggregate range values m, from 2 up to 64 values. Measurements are executed on a machine running the Rust compiler with an Intel Core i7-10750H CPU and 16 GB of RAM.

Table 1. Inner-Product-Argument (IPA) proof size comparison. **BP-IPA** is the IPA of Bulletproofs [7]. **TS-IPA** is our *two-step* reduction IPA presented in this work. **n** and **m** are respectively the bit-range domain and the number of aggregate range values.

n	m	BP-IPA proof size (bytes)	TS-IPA proof size (bytes)
16	2	720	400
16	4	848	400
16	8	976	400
16	16	1,104	400
16	32	1,232	400
16	64	1,360	400

The results in Table 1 highlight that as the number of aggregated values increases, hence m, the proof size of BP-IPA grows logarithmically while in our TS-IPA the proof size is clearly constant. This is in line with the theoretical results: the aggregated Bulletproofs (in [7], Sect. 4.3) shows a logarithmic proof size asymptotically equal to $O(log_2(m \cdot n))$, considering that at each

IPA round there are two collections of group elements $(L_1, \ldots, L_{log_2(m \cdot n)})$ and $(R_1, \ldots, R_{log_2(m \cdot n)})$. Instead, in TS-IPA the two collections end up having only 4 group elements (L_1, L_0) and (R_1, R_0). Hence, the proof size is constant and the total proof consists of 5 $\mathbb{G}_1$ points and 2 $\mathbb{Z}_p$ scalars. Figure 1 shows the asymptotic sizes of the BP-IPA and TS-IPA proofs.

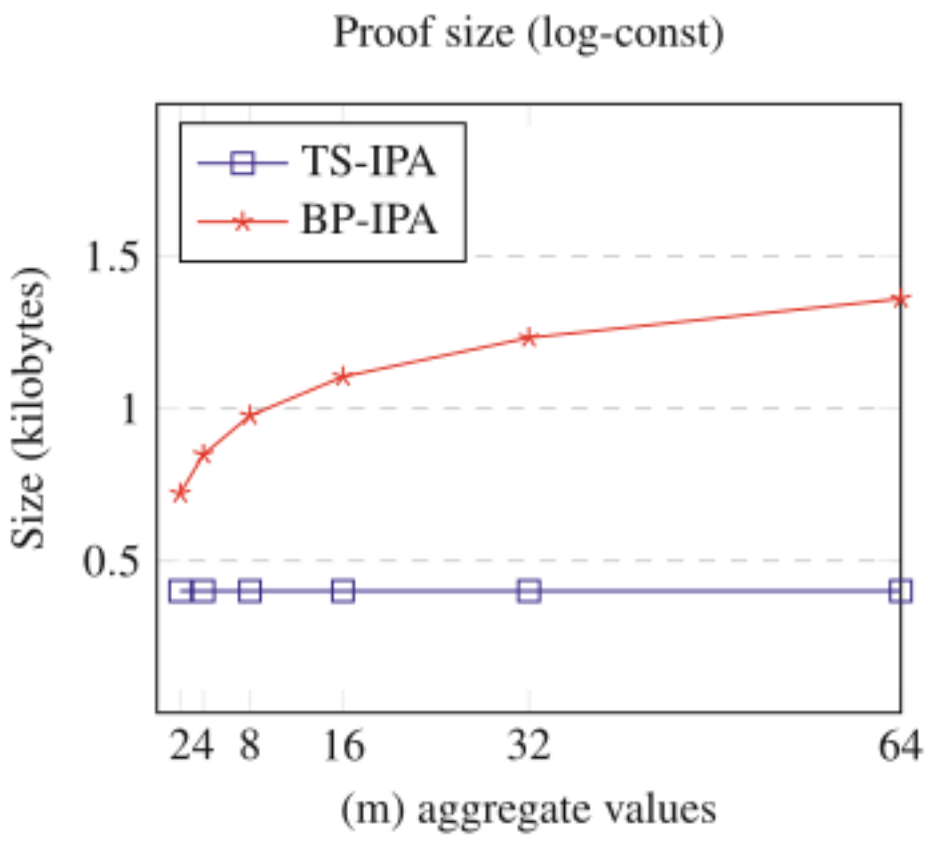

Fig. 1. Proof size comparison TS-IPA and BP-IPA

6 Conclusion and Future Work

Range proofs in standard security assumptions, standard groups and without trusted setup are attractive in confidential transaction protocols. However, range proofs lack succinct verification and proof size. We presented a modified Inner-Product-Argument protocol for range proof systems such as Bulletproofs, and our two-step reduction algorithm keeps the size of the proof constant. Moreover, we also reduce the communication complexity since the proof size is in the order of bytes. In this work, we assumed that the new relation for IPA introduced by Halo is sound and has zero-knowledge, however, further investigations are needed. As future work, we will validate our approach in real case studies involving streams of sensor data [16–18].

References

1. Alonso, K.M., et al.: Zero to Monero (2020)
2. arkworks rs. arkworks
3. 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

4. Bowe, S., Chiesa, A., Green, M., Miers,I., Mishra, P., Wu, H.: Zexe: enabling decentralized private computation. In: 2020 IEEE Symposium on Security and Privacy (SP), pp. 947–964. IEEE (2020)
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. EmanueleSc. Zeromt
13. 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
14. Jivanyan, A.: Lelantus: towards confidentiality and anonymity of blockchain transactions from standard assumptions. IACR Cryptol. ePrint Arch. **2019**, 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. Mehmood, N.Q., Culmone, R., Mostarda, L.: Modeling temporal aspects of sensor data for MongoDB NoSQL database. J. Big Data **4**(1), (2017)
17. Russello, G., Mostarda, L., Dulay, N.: A policy-based publish/subscribe middleware for sense-and-react applications. J. Syst. Softw. **84**(4), 638–654 (2011)
18. Vannucch, C., et al.: Symbolic verification of event–condition–action rules in intelligent environments. J. Reliable Intell. Environ. **3**(2), 117–130 (2017)
19. Xiong, A.I., Chen, B., Zhang, Bünz, B., Fisch, B., Krell, F., Camacho. P.: Verizexe: decentralized private computation with universal setup. *Cryptology ePrint Archive* (2022)