Generalized Bulletproofs

Cypher Stack*

May 8, 2024

This report describes the Generalized Bulletproofs construction and provides a relevant security proof. As with any such report, it may contain errors and cannot guarantee correctness or security. Further, it cannot guarantee that any particular implementation of the construction is correct, secure, or suitable for intended use cases.

The author asserts no warranty and disclaims liability for its use. The author further expresses no endorsement of any kind. This report has not undergone any further formal or peer review.

Contents

1 Introduction

Curve Trees [2] is a recent design for a cryptographic accumulator. Unlike many other approaches, the Curve Trees approach is built to use the trustless arithmetic circuit satisfiability proving system used in Bulletproofs [1], which has properties useful for ensuring practical communication and computational efficiency.

As described in the Curve Trees preprint, a major component of the desired efficiency requires a modification of the relevant Bulletproofs proving system in order to accommodate Pedersen vector commitments. The authors call this modification Generalized Bulletproofs.

*https://cypherstack.com

The design of Generalized Bulletproofs is partially described as part of a proof-of-concept repository[1] and fully implemented in Rust. However, to our knowledge, it has never been fully documented or proven secure.

In this report, we fully describe the Generalized Bulletproofs design as a standalone protocol for ease of analysis and implementation. We identify an issue whereby the existing description and implementation do not admit a proof of computational witness-extended emulation. To solve this, we provide a modified proving relation that extends the generators used for the added Pedersen vector commitments, along with a corresponding change to the protocol. We are able to take advantage of the modified relation to extend the protocol in a way that may be useful for efficiency, by allowing for the use of additional weighting matrices that allow for more arithmetic circuit constraints within vector commitments, albeit at the cost of increased proof size. We then produce a proof of security analogous to that of Theorem 4 from the Bulletproofs preprint. This proof shows that, like its parent proving system, the (modified) Generalized Bulletproofs design has perfect completeness, perfect special honest-verifier zero knowledge, and computational witness-extended emulation.

We note carefully that while the Bulletproofs preprint uses multiplicative notation for group operations, we use additive notation here. This is both to improve readability and aid implementation, as most popular elliptic curve libraries use additive conventions for group operations.

We generally follow similar notation as the Bulletproofs preprint, and endeavor to use the same symbols where feasible, with a few exceptions. While the preprint uses $\mathbb{Z}_p$ to represent the scalar field of the group $\mathbb{G}$, we use the symbol $\mathbb{F}$ for clarity. However, while the preprint uses lowercase letters to represent certain group elements, we use the corresponding capital letters in order to more consistently differentiate group elements from scalars. Like the preprint, we continue to use capital letters for matrices.

The author thanks Luke Parker for helpful review and discussion, especially leading to discovery of an indexing issue with an earlier version of this report.

2 Protocol

We now describe a generalization of the Generalized Bulletproofs design as an interactive protocol between a prover and a verifier, including a modification whose requirement we discuss later. The protocol can be made non-interactive using the strong Fiat-Shamir technique, which replaces verifier challenges with the output of a cryptographic hash function (modeled as a random oracle) using the proof transcript as input.

Let n_c be the number of Pedersen vector commitments $\vec{C}$ used by the protocol, where for $k \in [1, n_c]$ we have $C_k = \vec{c}_{k,L}\vec{G} + \vec{c}_{k,R}\vec{H} + c'_k H$. We require that $n_c = O(\lambda)$ for security parameter λ. For each $k \in [1, n_c]$, let $\vec{W}_{k,L}$ be the corresponding weighting matrix for the $\vec{G}$ components of C_k, and let $\vec{W}_{k,R}$ be the

[1] `https://github.com/simonkamp/curve-trees/blob/main/bulletproofs/generalized-bulletproofs.md`

weighting matrix for the $\vec{H}$ components of C_k. Note that this differs from the existing Generalized Bulletproofs description and implementation, where vector commitments have no $\vec{H}$ component or corresponding weighting matrices.

The protocol is a proving system for the following relation:

$$\Bigg\{ G, H \in \mathbb{G}, \vec{G}, \vec{H} \in \mathbb{G}^n,$$
$$\vec{V} \in \mathbb{G}^m, \vec{C} \in \mathbb{G}^{n_c}, \vec{c} \in \mathbb{F}^Q,$$
$$\vec{W}_L, \vec{W}_R, \vec{W}_O, \{(\vec{W}_{k,L}, \vec{W}_{k,R})\}_{k=1}^{n_c} \in \mathbb{F}^{Q \times n}, \vec{W}_V \in \mathbb{F}^{Q \times m};$$
$$\vec{a}_L, \vec{a}_R, \vec{a}_O, \{(\vec{c}_{k,L}, \vec{c}_{k,R})\}_{k=1}^{n_c} \in \mathbb{F}^n, \vec{v}, \vec{\gamma} \in \mathbb{F}^m, \vec{c}' \in \mathbb{F}^{n_c} \mid$$
$$\forall j \in [1, m] : V_j = v_j G + \gamma_j H,$$
$$\forall k \in [1, n_c] : C_k = \vec{c}_{k,L} \vec{G} + \vec{c}_{k,R} \vec{H} + c'_k H,$$
$$\vec{a}_L \circ \vec{a}_R = \vec{a}_O,$$
$$\vec{W}_L \vec{a}_L + \vec{W}_R \vec{a}_R + \vec{W}_O \vec{a}_O + \sum_{k=1}^{n_c} \left(\vec{W}_{k,L} \vec{c}_{k,L} + \vec{W}_{k,R} \vec{c}_{k,R} \right) = \vec{W}_V \vec{v} + \vec{c} \Bigg\} \quad (1)$$

We require that $\vec{W}_V$ have rank m. The relation is organized such that the first line contains fixed parameters, the next two contain the statement, the next contains the witness, and the remaining contain the conditions.

Observe that if we allow $n_c = 0$ (a slight misuse of notation), the original Bulletproofs arithmetic circuit satisfiability relation is recovered. In practice, we require $n_c > 0$ as indicated to ensure the modification is nontrivial.

A key component to the modification to the original Bulletproofs proving system is in the construction of vector polynomials $\vec{l}(X)$ and $\vec{r}(X)$ to accommodate the added Pedersen vector commitments and associated weighting matrices. This change involves carefully including certain elements as specific coefficients of these polynomials.

To make more clear how these coefficients are arranged, let $n' = 2(n_c + 1)$. Define the following pairs of indices, which we will use later:

$$\begin{array}{ll} i_{LR} = n'/2 & j_{LR} = i_{LR} \\ i_O = n' & j_O = 0 \\ i_S = n' + 1 & j_S = i_S \\ i_k = k & j_k = n' - k \end{array}$$

Here k takes on each value in the range $[1, n_c]$. These indices are assigned such that pairs (aside from the S pair, which functions differently for masking purposes) sum to n', which will be important in the protocol.

Overall, the protocol closely mirrors Protocol 3 in the Bulletproofs preprint. To execute the interactive protocol, the prover does the following:

- Samples $\alpha, \beta, \rho \in \mathbb{F}$ uniformly at random.
- Computes
$$A_I = \vec{a_L} \vec{G} + \vec{a}_R \vec{H} + \alpha H$$

and

$$A_O = a_O^{\rightarrow}\vec{G} + \beta H.$$

- Samples $\vec{s}_L, \vec{s}_R \in \mathbb{F}^n$ uniformly at random.
- Computes $S = \vec{s}_L\vec{G} + \vec{s}_R\vec{H} + \rho H$.
- Sends A_I, A_O, S to the verifier.

The verifier samples challenges $y, z \in \mathbb{F}^*$ uniformly at random, and sends them to the prover. The prover and verifier both do the following:

- Computes

$$\vec{y^n} = \left(1, y, y^2, \ldots, y^{n-1}\right) \in \mathbb{F}^n$$

and

$$\vec{z}_{[1:]}^{Q+1} = \left(z, z^2, \ldots, z^Q\right) \in \mathbb{F}^Q$$

and

$$\delta(y,z) = \left\langle \vec{y^{-n}} \circ \vec{z}_{[1:]}^{Q+1}\vec{W}_R, \vec{z}_{[1:]}^{Q+1}\vec{W}_L \right\rangle.$$

The prover then does the following:

- Computes the following coefficients (for all $k \in [1, n_c]$):

$$\begin{array}{ll}
\vec{l}_{i_{LR}} = \vec{a}_L + \vec{y^{-n}} \circ \vec{z}_{[1:]}^{Q+1}\vec{W}_R & \vec{r}_{j_{LR}} = \vec{y^n} \circ \vec{a}_R + \vec{z}_{[1:]}^{Q+1}\vec{W}_L \\
\vec{l}_{i_O} = \vec{a}_O & \vec{r}_{j_O} = \vec{z}_{[1:]}^{Q+1}\vec{W}_O - \vec{y^n} \\
\vec{l}_{i_S} = \vec{s}_L & \vec{r}_{j_S} = \vec{y^n} \circ \vec{s}_R \\
\vec{l}_{i_k} = \vec{c}_{k,L} & \vec{r}_{j_k} = \vec{z}_{[1:]}^{Q+1}\vec{W}_{k,L} \\
\vec{l}_{j_k} = \vec{y^{-n}} \circ \vec{z}_{[1:]}^{Q+1}\vec{W}_{k,R} & \vec{r}_{i_k} = \vec{y^n} \circ \vec{c}_{k,R}
\end{array}$$

- Computes polynomials

$$\vec{l}(X) = \sum_{i=0}^{n'+1} l_i X^i$$

and

$$\vec{r}(X) = \sum_{i=0}^{n'+1} r_i X^i$$

in $\mathbb{F}^n[X]$ using the coefficients defined previously.

- Computes

$$t(X) = \left\langle \vec{l}(X), \vec{r}(X) \right\rangle = \sum_{i=0}^{2(n'+1)} t_i X^i \in \mathbb{F}[X].$$

- For $i \in [1, 2(n'+1)], i \neq n'$, samples $\tau_i \in \mathbb{F}$ uniformly at random.
- For $i \in [1, 2(n'+1)], i \neq n'$, computes $T_i = t_i G + \tau_i H$.

- Sends $\{T_i\}_{i=1, i\neq n'}^{2(n'+1)}$ to the verifier.

The verifier samples a challenge $x \in \mathbb{F}^*$ uniformly at random, and sends it to the prover. The prover then does the following:

- Computes $\vec{l} = \vec{l}(x)$ and $\vec{r} = \vec{r}(x)$, and sets $\widehat{t} = \left\langle \vec{l}, \vec{r} \right\rangle$.
- Computes

$$\tau_x = \sum_{i=1, i\neq n'}^{2(n'+1)} \tau_i x^i + x^{n'} \left\langle \vec{z}_{[1:]}^{Q+1} \vec{W}_V, \gamma \right\rangle$$

  and

$$\mu = \alpha x^{i_{LR}} + \beta x^{i_O} + \rho x^{i_S} + \sum_{k=1}^{n_c} c'_k x^{i_k}.$$

- Sends $\tau_x, \mu, \widehat{t}, \vec{l}, \vec{r}$ to the verifier.

The verifier then does the following:

- For $i \in [1, n]$, sets $H'_i = y^{-i+1} H_i$ as the vector $\vec{H}'$.
- Computes

$$W_L = \left(\vec{z}_{[1:]}^{Q+1} \vec{W}_L \right) \vec{H}'$$

  and

$$W_R = \left(\vec{y}^{-n} \circ \vec{z}_{[1:]}^{Q+1} \vec{W}_R \right) \vec{G}$$

  and

$$W_O = \left(\vec{z}_{[1:]}^{Q+1} \vec{W}_O \right) \vec{H}'.$$

- For $k \in [1, n_c]$, computes

$$W_{k,L} = \left(\vec{z}_{[1:]}^{Q+1} \vec{W}_{k,L} \right) \vec{H}'$$

  and

$$W_{k,R} = \left(\vec{y}^{-n} \circ \vec{z}_{[1:]}^{Q+1} \vec{W}_{k,R} \right) \vec{G}.$$

- Asserts that the equation

$$\widehat{t} = \left\langle \vec{l}, \vec{r} \right\rangle \tag{2}$$

  holds.

- Asserts that the equation

$$\widehat{t} G + \tau_x H = x^{n'} \left(\delta(y, z) + \left\langle \vec{z}_{[1:]}^{Q+1}, \vec{c} \right\rangle \right) G + x^{n'} \left(\vec{z}_{[1:]}^{Q+1} \vec{W}_V \right) \vec{V} + \sum_{i=1, i\neq n'}^{2(n'+1)} x^i T_i \tag{3}$$

  holds.

- Sets

$$P = x^{i_{LR}} A_I + x^{i_O} A_O - x^{j_O} \left(\vec{y}^n \vec{H}' \right) + x^{i_{LR}} W_L + x^{i_{LR}} W_R + x^{j_O} W_O + x^{i_S} S + \sum_{k=1}^{n_c} x^{j_k} W_{k,L} + \sum_{k=1}^{n_c} x^{j_k} W_{k,R} + \sum_{k=1}^{n_c} x^{i_k} C_k$$

  and asserts that the equation

$$P = \vec{l}\vec{G} + \vec{r}\vec{H}' + \mu H \tag{4}$$

  holds.

- If the assertions in Equations 2, 3, and 4 all pass, accepts the proof. Otherwise, rejects the proof.

Observe that the modified protocol is purely additive, in the sense that it reduces to the Bulletproofs protocol in the case where $n_c = 0$ and there are no Pedersen vector commitments or associated weighting matrices.

The inner-product argument presented in Protocol 2 of the Bulletproofs preprint applies identically to the Generalized Bulletproofs design. This reduces the communication complexity logarithmically by replacing the prover's transmission of $\vec{l}$ and $\vec{r}$ with an execution of the inner-product argument.

3 Security

We now prove that the Generalized Bulletproofs arithmetic circuit satisfiability proving system has the desired security properties, using a theorem identical to Theorem 4 in the Bulletproofs preprint.

Theorem. *The proof system presented has perfect completeness, perfect special honest-verifier zero knowledge, and computational witness-extended emulation.*

The proof closely follows Appendix D in the Bulletproofs preprint, with corresponding changes to accommodate the protocol modifications. Because the changes are nontrivial, we include the full proof.

Proof. We first show perfect completeness. In the case of an honest prover, it suffices to show that Equations 2, 3, and 4 hold. Equation 2 holds by definition. Equation 4 holds by inspection. Equation 3 holds provided that

$$t_{n'} = \delta(y,z) + \left\langle \vec{z}_{[1:]}^{Q+1}, \vec{W}_L \vec{a_L} + \vec{W}_R \vec{a_R} + \vec{W}_O \vec{a_O} + \sum_{k=1}^{n_c} \left(\vec{W}_{k,L} \vec{c}_{k,L} + \vec{W}_{k,R} \vec{c}_{k,R} \right) \right\rangle$$

since, in this case,

$$t_{n'} = \delta(y,z) + \left\langle \vec{z}_{[1:]}^{Q+1}, \vec{W}_V \vec{v} + \vec{c} \right\rangle$$

for a valid witness by Relation 1.

To show the equality holds, we observe by construction of the polynomials $\vec{l}(X)$ and $\vec{r}(X)$ that $t_{n'}$ is the degree-n' coefficient of their inner product $t(X)$. Because of the construction of polynomial indices, this coefficient is given by

$$t_{n'} = \left\langle \vec{l}_{i_{LR}}, \vec{r}_{j_{LR}} \right\rangle + \left\langle \vec{l}_{i_O}, \vec{r}_{j_O} \right\rangle + \sum_{k=1}^{n_c} \left\langle \vec{l}_{i_k}, \vec{r}_{j_k} \right\rangle + \sum_{k=1}^{n_c} \left\langle \vec{l}_{j_k}, \vec{l}_{i_k} \right\rangle$$

from which the required equality holds algebraically by the definition of these terms, using the fact that $\vec{a}_l \circ \vec{a}_R = \vec{a}_O$ implies that

$$\langle \vec{a}_L, \vec{a}_R \circ \vec{y}^n \rangle - \langle \vec{a}_O, \vec{y}^n \rangle = \sum_{i=0}^{n-1} y^i \left(a_{L,i} a_{R,i} - a_{O,i} \right) = 0$$

for an honest prover.

To show perfect special honest-verifier zero knowledge, we construct a simulator that, given a statement and uniformly-sampled verifier challenges, can produce a valid proof transcript distributed identically to that of a real proof.

Fix an arbitrary statement using Relation 1, and sample verifier challenges $x, y, z \in \mathbb{F}^*$ uniformly at random. The simulator begins by sampling $\vec{l}, \vec{r}$ uniformly at random, after which it sets

$$\widehat{t} = \left\langle \vec{l}, \vec{r} \right\rangle$$

to trivially satisfy Equation 2. It then samples τ_x and $\{T_i\}_{i=2, i \neq n'}^{2(n'+1)}$ uniformly at random, and defines

$$T_1 = -x^{-1} \left[x^{n'} \left(\delta(y,z) + \left\langle \vec{z}_{[1:]}^{Q+1}, \vec{c} \right\rangle \right) G + x^{n'} \left(\vec{z}_{[1:]}^{Q+1} \vec{W}_V \right) \vec{V} + \sum_{i=2, i \neq n'}^{2(n'+1)} x^i T_i - \widehat{t} G - \tau_x H \right]$$

to satisfy Equation 3. Finally, it samples A_I, A_O, μ uniformly at random, and defines

$$S = -x^{-i_S} \left[x^{i_{LR}} A_I + x^{i_O} A_O - x^{j_O} \left(\vec{y}^n \vec{H}' \right) + x^{i_{LR}} W_L + x^{i_{LR}} W_R + x^{j_O} W_O + \sum_{k=1}^{n_c} x^{j_k} W_{k,L} + \sum_{k=1}^{n_c} x^{j_k} W_{k,R} + \sum_{k=1}^{n_c} x^{i_k} C_k - \vec{l} \vec{G} - \vec{r} \vec{H}' - \mu H \right]$$

to satisfy Equation 4.

The resulting simulated transcript is valid by definition, so it remains to show that it is distributed identically to that of a real proof. Because in a real proof α and β are sampled uniformly at random, A_I and A_O are also distributed

uniformly at random as Pedersen vector commitments. Similarly, in a real proof the elements

$$\{\tau_i\}_{i=1,i\neq n'}^{2(n'+1)}$$

are sampled uniformly at random, so the elements

$$\{T_i\}_{i=1,i\neq n'}^{2(n'+1)}$$

are distributed uniformly at random as Pedersen commitments; the uniform sampling of nonzero x and z means τ_x is also distributed uniformly at random. Both $\vec{l}$ and $\vec{r}$ are constructed using masking offsets $\vec{s}_L, \vec{s}_R$ sampled uniformly at random and applied against nonzero x and y, so they are distributed uniformly at random as well. Further, $\widehat{t}$ is defined identically in both a simulated and real proof. Finally, μ is distributed uniformly at random in a real proof given the nonzero random challenge x and masks α, β, ρ.

This shows the protocol has perfect special honest-verifier zero knowledge.

It remains to show that the protocol has computational witness-extended emulation. To do so, we construct an extractor that, given a fixed statement, is able to arbitrarily rewind the transcript and sample distinct challenges to generate a tree of valid transcripts; we must show that this extractor is able to produce a valid witness for the statement. The extractor we define uses n distinct challenges y, $Q+1$ distinct challenges z, and $2(n'+1)+1$ challenges x; this results in a total of $[2(n'+1)+1]\,(Q+1)n$ transcripts.

The extractor first fixes challenges y, z and rewinds to obtain n_c+3 unique x challenges $\{x_i\}_{i=1}^{n_c+3}$. With high probability, we can obtain weighting coefficients $\{\nu_i\}_{i=1}^{n_c+3}$ such that

$$\sum_{i=1}^{n_c+3} \nu_i x_i^{i_{LR}} = 1$$

and

$$\sum_{i=1}^{n_c+3} \nu_i x_i^{\xi} = 0$$

for $\xi \in \{i_O, i_S\} \cup \{i_k\}_{k=1}^{n_c}$. Using the weighting coefficients with Equation 4, we obtain a linear combination where all challenge powers in the sum other than the i_{LR} exponent vanish, and that determines values $\alpha, \vec{a}_L, \vec{a}_R$ such that $A_I = \vec{a}_L\vec{G} + \vec{a}_R\vec{H} + \alpha H$. If these values are not uniquely determined, they yield a nontrivial discrete logarithm relation between the corresponding group generators.

The extractor then uses the same challenges $\{x_i\}_{i=1}^{n_c+3}$ to obtain new weighting coefficients such that all challenge power linear combinations vanish aside from that corresponding to the i_O exponent. This yields values $\beta, \vec{a}_{O,L}, \vec{a}_{O,R}$ such that $A_O = \vec{a}_{O,L}\vec{G} + \vec{a}_{O,R}\vec{H} + \beta H$; if these are not uniquely determined, they also yield a nontrivial generator discrete logarithm relation. The same reasoning applies to obtain openings $S = \vec{s}_L\vec{G} + \vec{s}_R\vec{H} + \rho H$ and, for $k \in [1, n_c]$, $C_k = \vec{c}_{k,L}\vec{G} + \vec{c}_{k,R}\vec{H} + c'_k H$. Note that in each case, the openings are with respect to all generators $\vec{G}, \vec{H}, H$ that appear in Equation 4.

We can then use these openings in Equation 4 for all challenge tuples (x, y, z) to replace the values $A_I, A_O, S, \vec{C}$. We can then express the vectors $\vec{l}, \vec{r}$ as

$$\vec{l} = \vec{a}_L x^{i_{LR}} + \vec{a}_{O,L} x^{i_O} + \left(\vec{y}^{-n} \circ \vec{z}_{[1:]}^{Q+1} \vec{W}_R\right) x^{i_{LR}} + \vec{s}_L x^{i_S} + \sum_{k=1}^{n_c} x^{j_k} \left(\vec{y}^{-n} \circ \vec{z}_{[1:]}^{Q+1} \vec{W}_{k,R}\right) + \sum_{k=1}^{n_c} \vec{c}_{k,L} x^{i_k}$$

and

$$\vec{r} = (\vec{y}^n \circ \vec{a}_R)\, x^{i_{LR}} + (\vec{y}^n \circ \vec{a}_{O,R})\, x^{i_O} - \vec{y}^n x^{j_O} + \left(\vec{z}_{[1:]}^{Q+1} \vec{W}_L\right) x^{i_{LR}} + \left(\vec{z}_{[1:]}^{Q+1} \vec{W}_O\right) x^{j_O} + (\vec{y}^n \circ \vec{s}_R)\, x^{i_S} + \sum_{k=1}^{n_c} \left(\vec{z}_{[1:]}^{Q+1} \vec{W}_{k,L}\right) x^{j_k} + \sum_{k=1}^{n_c} (\vec{y}^n \circ \vec{c}_{k,R})\, x^{i_k}$$

by matching generators. If this does not hold for all such challenges in valid transcripts, we again have a nontrivial discrete logarithm relation between the generators.

We now show that the inner-product coefficient $t_{n'}$ has the form described above, corresponding to that of an honest prover. To do so, the extractor takes $2(n'+1)$ valid transcripts corresponding to fixed challenges y, z and distinct challenges for x. Applying a similar linear combination technique as before to Equation 3 using these challenges, we obtain for each $i \in [1, 2(n'+1)], i \neq n'$ a tuple (τ_i, t_i) such that $T_i = t_i G + \tau_i H$. We also obtain v, γ such that the equation

$$vG + \gamma H = \left(\vec{z}_{[1:]}^{Q+1} \vec{W}_V\right) \vec{V}$$

holds. Now using m distinct challenges for z, we apply the linear combination technique to the above equation to obtain, for each $j \in [1, m]$, a tuple (v_j, γ_j) such that $v_j G + \gamma_j H = V_j$. Observe that this requires the weighting matrix $\vec{W}_V$ to be of full rank m. Matching terms associated to the generator G using the extracted values, we obtain that for any challenge tuple (x, y, z), either the equation

$$\widehat{t} = x^{n'} \left(\delta(y, z) + \left\langle \vec{z}_{[1:]}^{Q+1}, \vec{W}_V \vec{v} + \vec{c} \right\rangle\right) + \sum_{i=1, i \neq n'}^{2(n'+1)} t_i x^i$$

holds, or we obtain a nontrivial discrete logarithm relation between G and H.

Provided the equation holds, consider $2(n'+1)+1$ transcripts with fixed y, z and distinct x. Let

$$t_{n'} = \delta(y, z) + \left\langle \vec{z}_{[1:]}^{Q+1}, \vec{W}_V \vec{v} + \vec{c} \right\rangle$$

and define

$$p(x) = \sum_{i=1}^{2(n'+1)} p_i x^i = \left\langle \vec{l}, \vec{r} \right\rangle$$

and

$$t(x) = \sum_{i=1}^{2(n'+1)} t_i x^i$$

considering both $t(x)$ and $p(x)$ as polynomials in $\mathbb{F}^n[X]$ evaluated at $X = x$. Observe that $t(X) - p(X)$ is a polynomial of degree $2(n_c + 1)$ such that $t(x) - p(x) = 0$ for each of the $2(n' + 1) + 1$ values of x; this means it is the zero polynomial in $\mathbb{F}^n[X]$.

Using the representations of $\vec{l}$ and $\vec{r}$ above, it follows that

$$p_{n'} = \delta(y,z) + \langle \vec{a}_L, \vec{y}^n \circ \vec{a}_R \rangle - \langle \vec{a}_{O,L}, \vec{y}^n \rangle + \left\langle \vec{z}_{[1:]}^{Q+1}, \vec{W}_L \vec{a}_L + \vec{W}_R \vec{a}_R + \vec{W}_O \vec{a}_{O,L} + \sum_{i=1}^{n_c} \left(\vec{W}_{k,L} \vec{c}_{k,L} + \vec{W}_{k,R} \vec{c}_{k,R} \right) \right\rangle$$

using the definitions of polynomial indices that sum to n' across $\vec{l}$ and $\vec{r}$. The polynomial equality between $t(X)$ and $p(X)$ mean that for any y, z we have $p_2 = t_2$.

Now fix y and consider $Q + 1$ distinct challenges z, and consider $p_2(z)$ and $t_2(z)$ as polynomials in $\mathbb{F}^n[Z]$ evaluated at $Z = z$. Since $p_2(z) = t_2(z)$ for all such evaluations, this difference is the zero polynomial. Similarly, using n distinct challenges y, the equalities

$$\vec{a}_L \circ \vec{a}_R - \vec{a}_{O,L} = \vec{0}^n$$

and

$$\vec{W}_L \vec{a}_L + \vec{W}_R \vec{a}_R + \vec{W}_O \vec{a}_{O,L} + \sum_{i=1}^{n_c} \left(\vec{W}_{k,L} \vec{c}_{k,L} + \vec{W}_{k,R} \vec{c}_{k,R} \right) = \vec{W}_V \vec{v} + \vec{c}$$

follow. Setting $\vec{a}_O = \vec{a}_{O,L}$, we have a valid witness tuple

$$(\vec{a}_L, \vec{a}_R, \vec{a}_O, \{(\vec{c}_{k,L}, \vec{c}_{k,R})\}_{k=1}^{n_c}, \vec{v}, \vec{\gamma}, \vec{c}')$$

that satisfies Relation 1.

The extractor is efficient and the number of transcripts is polynomial in the security parameter. Extraction returns either a valid witness or a nontrivial discrete logarithm relation between generators. If the generators are sampled uniformly at random, the extractor returns such a relation with negligible probability. This means we can apply the forking lemma and computational witness-extended emulation holds. □

Because the inner-product argument technique directly applies to the Generalized Bulletproofs design, we can present a theorem identical to Theorem 5 in the Bulletproofs preprint. The proof is identical as well, so we omit it here.

Theorem. *The arithmetic circuit protocol using the improved inner-product argument has perfect completeness, statistical zero knowledge, and computational soudness under the discrete logarithm assumption.*

4 Finding

As noted above, the protocol described and proven secure in this report differs from the existing Generalized Bulletproofs design. Specifically, Pedersen vector commitments in the original design are with respect to $\vec{G}, H$; in our modification, they are with respect to $\vec{G}, \vec{H}, H$. Further, we include additional weighting matrices; for all $k \in [1, n_c]$, the weighting matrix $W_{k,L}$ acts on corresponding vector commitment opening component $\vec{c}_{k,L}$, and the weighting matrix $W_{k,R}$ acts on opening component $\vec{c}_{k,R}$.

The reason for these changes can be seen in the proof of computational witness-extended emulation. The extractor defined in the proof obtains, for $k \in [1, n_c]$, the opening

$$C_k = \vec{c}_{k,L}\vec{G} + \vec{c}_{k,R}\vec{H} + c'_k H$$

to the corresponding Pedersen vector commitment. For this opening to represent a valid witness in the original design, it must be the case that $\vec{c}_{k,R}$ vanishes; however, this does not follow from the structure of the verifier.

By modifying the vector commitment structure to permit $\vec{H}$ components, the extracted witness is valid. This requires a minor change to the prover as well, which now accounts for each $\vec{c}_{k,R}$. This enables us to include the weighting matrix generalization as well.

We caution that any use of this modified protocol must ensure that the relation meets any security requirements.

References

[1] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. Cryptology ePrint Archive, Paper 2017/1066, 2017. `https://eprint.iacr.org/2017/1066`.

[2] Matteo Campanelli, Mathias Hall-Andersen, and Simon Holmgaard Kamp. Curve trees: Practical and transparent zero-knowledge accumulators. Cryptology ePrint Archive, Paper 2022/756, 2022. `https://eprint.iacr.org/2022/756`.