Generalized Bulletproofs for Opening Vector Commitments

Brandon Goodell*

January 28, 2026

Contents

1 Introduction

This report reviews changes made to the Generalized Bulletproofs construction and corresponding security proofs, and is a continuation of the previous work in [4] by a distinct author.

We describe a generalization of Bulletproofs which supports proving the openings of some Pedersen scalar commitments and Pedersen vector commitments satisfy an arithmetic circuit, and prove our generalization secure. The original Bulletproofs protocol in [1] supports only Pedersen scalar commitments. In [2], a generalization of Bulletproofs supporting vector commitments was first described; see also [3]. A problem with the extractor was identified and a proposed fix was presented in [4], but that proposal had problems with the security proofs. In fact, the correction was such that a verifier could not even be convinced an arithmetic circuit is satisfied without external verification.

The protocol here fixes these errors. Proofs are dependent on all witness data, the extractor works without breaking the discrete logarithm problem,

[*] mailto:brandon@cypherstack.com

and, for the desired relation, the protocol closes the correctness-soundness gap from the previous proposal. If the parameter n_c governs the number of Pedersen vector commitments and the parameter m governs the number of Pedersen scalar commitments used in the protocol, then our protocol requires communicating $3n_c + 8$ elements of $\mathbb{G}$ (namely A_I, A_O, S, and $\underline{T}$) and $4n_c + 13$ elements of $\mathbb{F}$.

2 Preliminaries

Let $\mathbb{F}$ be a field and $\mathbb{G}$ an elliptic curve group over $\mathbb{F}$. Let $O \in \mathbb{G}$ denote the group identity. Let $FR(\mathbb{F}^{m\times n})$ be the *full-rank* $m \times n$ matrices with elements of $\mathbb{F}$. We use underlines to denote vectors. We handle all vectors as matrices, writing inner product notation with matrix transposes, so $\underline{c}_L^\top \underline{G} + \underline{c}_R^\top \underline{H} + c'H$ denotes a Pedersen vector commitment to the value vector $\underline{c}_L$ with individualized blinders in $\underline{c}_R$ and a common blinder in c'. Readers may be more familiar with the equivalent inner product notation $\langle \underline{c}_L, \underline{G}\rangle + \langle \underline{c}_R, \underline{H}\rangle + c'_k H$. For any natural number $n \in \mathbb{N}$, let $[n]$ be the set of n elements $[n] = \{0, 1, \ldots, n-1\}$.

Let Q, m, n, n_c be natural numbers and $Q \geq m$. Herein, n is the number of gates in an arithmetic circuit, $m \geq 1$ is a number of Pedersen scalar commitments, and $n_c \geq 1$ is a number of Pedersen vector commitments. Let $G, H \in \mathbb{G}$ be public base points, and let $\underline{G}, \underline{H} \in \mathbb{G}^n$ be vectors of public base points. We assume these public basepoints have no known discrete logarithm relation.

Let $\mathcal{R}$ be the relation with witnesses and statements

$$\mathtt{w} = \left(\underline{a}_L, \underline{a}_R, \underline{a}_O, \left\{(\underline{c}_{k,L}, \underline{c}_{k,R})\right\}_{k=1}^{n_c}, \underline{v}, \underline{\gamma}, \underline{c}'\right)$$
$$\mathtt{s} = \left(\underline{V}, \underline{C}, \underline{c}, W_L, W_R, W_O, \{W_{k,L}\}_{k=1}^{n_c}, W_V\right)$$

where witnesses satisfy $\mathtt{w} \in (\mathbb{F}^n)^{2n_c+3} \times (\mathbb{F}^m)^2 \times \mathbb{F}^{n_c}$, statements satisfy $\mathtt{s} \in \mathbb{G}^m \times \mathbb{G}^{n_c} \times \mathbb{F}^Q \times (\mathbb{F}^{Q\times n})^{n_c+3} \times FR(\mathbb{F}^{Q\times m})$, and $(\mathtt{s}, \mathtt{w}) \in \mathcal{R}$ if and only if all the following are satisfied.

(a) $\underline{a}_O = \underline{a}_L \circ \underline{a}_R$,

(b) $W_V\underline{v} + \underline{c} = W_L\underline{a}_L + W_R\underline{a}_R + W_O\underline{a}_O + \sum_{k=1}^{n_c} W_{k,L}\underline{c}_{k,L}$,

(c) for each $k \in [m]$, $V_k = v_k G + \gamma_k H$, and

(d) for each $k \in [n_c]$, $C_k = \underline{c}_{k,L}^\top \underline{G} + \underline{c}_{k,R}^\top \underline{H} + c'_k H$.

We assume $n_c \geq 1$, but our discussion extends to the case $n_c = 0$ by reducing to Bulletproofs arithmetic circuit satisfiability.

3 Method

The protocol can be made non-interactive with a strong Fiat-Shamir approach by hashing all transcript data to determine challenges x, y, and z as usual. 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 $\underline{\ell}$ and $\underline{r}$ with an execution of the inner-product argument.

3.1 Protocol

The protocol accommodates the Pedersen vector commitments C_k by defining the Bulletproofs-style vectors of polynomials $\underline{\ell}(X)$ and $\underline{r}(X)$ to contain relevant proof data in coefficients which have been carefully constructed to demonstrate the desired relation.

To clarify, let $n' = 2n_c + 2$. The protocol indices are denoted i_{LR}, j_{LR}, i_O, j_O, i_S, j_S, and, for each $k \in [n_c]$, i_k and j_k, and are defined as follows.

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

These protocol indices, displayed in Table 1, are assigned such that $i_O + j_O = i_{LR} + j_{LR} = n'$ and, for each $k \in [n_c]$, $i_k + j_k = n'$. This way, the Bulletproofs-style polynomial $t(X) = \underline{\ell}(X)^\top \underline{r}(X)$ has all proof data stored in the coefficient on the n'-th monomial.

0	1	$\cdots$	$\frac{n'}{2} - 1$	$\frac{n'}{2}$	$\frac{n'}{2} + 1$	$\cdots$	$n' - 1$	n'	$n' + 1$
j_O	i_1	$\cdots$	i_{n_c}	$i_{LR} = j_{LR}$	j_{n_c}	$\cdots$	j_1	i_O	$i_S = j_S$

Table 1: The protocol indices.

Definition 1. The Generalized Bulletproofs protocol is five-move interactive proving system which takes place between a stateful prover-verifier pair. The prover inputs a statement-witness pair $(\mathtt{s}, \mathtt{w})$, the verifier inputs a statement $\mathtt{s}$, and the protocol proceeds as follows.

1. The prover does the following.

   (a) Sample $\alpha, \beta, \rho \xleftarrow{\$} \mathbb{F}$, $\underline{s}_L, \underline{s}_R \xleftarrow{\$} \mathbb{F}^n$.

   (b) Compute the following.

   $$A_I = \underline{a_L}^\top \underline{G} + \underline{a}_R^\top \underline{H} + \alpha H$$

   $$A_O = \underline{a_O}^\top \underline{G} + \beta H.$$

   $$S = \underline{s}_L^\top \underline{G} + \underline{s}_R^\top \underline{H} + \rho H$$

   (c) Output (A_I, A_O, S).

2. The verifier inputs some (A_I, A_O, S) and does the following.

(a) Sample[1] challenges $y, z \xleftarrow{\$} \mathbb{F}^*$.

(b) Compute the following.

$$\begin{aligned}
\widetilde{\underline{y}} &= \left(1, y, y^2, \ldots, y^{n-1}\right) \in \mathbb{F}^n, \\
\widetilde{\underline{y}}^{-1} &= \left(1, y^{-1}, y^{-2}, \ldots, y^{-(n-1)}\right) \in \mathbb{F}^n, \\
\widetilde{\underline{z}} &= \left(z, z^2, \ldots, z^Q\right) \in \mathbb{F}^Q, \text{ and} \\
\delta(y,z) &= (\widetilde{\underline{y}}^{-1} \circ \widetilde{\underline{z}})^\top W_R W_L^\top \widetilde{\underline{z}}
\end{aligned}$$

(c) Output (y, z).

3. The prover inputs its previous state and the challenge pair (y, z) and does the following.

(a) Compute $\widetilde{\underline{y}}$, $\widetilde{\underline{y}}^{-1}$, $\widetilde{\underline{z}}$, and $\delta(y,z)$ as in Step 2b above.

(b) Compute/set the following vectors.

$$\begin{aligned}
\underline{\ell}_{i_{LR}} &= \underline{a}_L + W_R^\top (\widetilde{\underline{y}}^{-1} \circ \widetilde{\underline{z}}) & \underline{r}_{j_{LR}} &= \widetilde{\underline{y}} \circ \underline{a}_R + W_L^\top \widetilde{\underline{z}} \\
\underline{\ell}_{i_O} &= \underline{a}_O & \underline{r}_{i_O} &= \underline{0} \\
\underline{\ell}_{j_O} &= \underline{0} & \underline{r}_{j_O} &= W_O^\top \widetilde{\underline{z}} - \widetilde{\underline{y}} \\
\underline{\ell}_{i_S} &= \underline{s}_L & \underline{r}_{j_S} &= \widetilde{\underline{y}} \circ \underline{s}_R
\end{aligned}$$

Also compute/set the following vectors for each $k \in [n_c]$.

$$\begin{aligned}
\underline{\ell}_{i_k} &= \underline{0} & \underline{r}_{i_k} &= W_{k,L}^\top \widetilde{\underline{z}} \\
\underline{\ell}_{j_k} &= \underline{c}_{k,L} & \underline{r}_{j_k} &= \widetilde{\underline{y}} \circ \underline{c}_{k,R}
\end{aligned}$$

We display the construction of $\underline{\ell}$ and $\underline{r}$ in Table 2.

(c) Set the following vectors of polynomials.

$$\underline{\ell}(X) = \sum_{i=0}^{n'+1} \underline{\ell}_i X^i$$

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

(d) Compute

$$t(X) = \langle \underline{\ell}(X), \underline{r}(X) \rangle = \sum_{i=\frac{n'}{2}}^{2(n'+1)} t_i X^i \in \mathbb{F}[X]$$

[1] It is important that these are sampled from $\mathbb{F}^* = \mathbb{F} \setminus \{0\}$, not all of $\mathbb{F}$.

Index k	Alias	$\underline{\ell}_k$	$\underline{r}_k$
0	j_O	0	$W_O^\top \underline{\widetilde{z}} - \underline{\widetilde{y}}$
1	i_1	0	$W_{1,L}^\top \underline{\widetilde{z}}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\frac{n'}{2} - 1$	i_{n_c}	0	$W_{n_c,L}^\top \underline{\widetilde{z}}$
$\frac{n'}{2}$	i_{LR}	$\underline{a}_L + W_R^\top (\underline{\widetilde{y}}^{-1} \circ \underline{\widetilde{z}})$	$\underline{\widetilde{y}} \circ \underline{a}_R + W_L^\top \underline{\widetilde{z}}$
$\frac{n'}{2} + 1$	j_{n_c}	$\underline{c}_{n_c,L}$	$\underline{c}_{n_c,R}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$
$n' - 1$	j_1	$\underline{c}_{1,L}$	$\underline{c}_{1,R}$
n'	i_O	$\underline{a}_O$	0
$n' + 1$	i_S	$\underline{s}_L$	$\underline{\widetilde{y}} \circ \underline{s}_R$

Table 2: The construction of the $\underline{\ell}$ and $\underline{r}$ polynomials. Note that the first $\frac{n'}{2}$ entries of $\underline{\ell}$ are $\underline{0}$. This way, $t(X)$ has no small coefficients, allowing reduced communication complexity in sending $\underline{T}$ in optimized implementations.

(e) Sample $\tau_i \xleftarrow{\$} \mathbb{F}$ for each $i = \frac{n'}{2}, \frac{n'}{2} + 1, \cdots, 2(n'+1)$ such that $i \neq n'$ and compute the following.

$$T_i = t_i G + \tau_i H$$

(f) Set $\underline{T} = \{T_i\}_{i=\frac{n'}{2}, i \neq n'}^{2(n'+1)}$ and output $\underline{T}$.

4. The verifier inputs its previous state and $\underline{T}$ and does the following.

   (a) Sample[2] $x \xleftarrow{\$} \mathbb{F}^*$.

   (b) Output x.

5. The prover inputs its previous state and x and does the following.

   (a) Evaluate $\underline{\ell} = \underline{\ell}(X = x)$ and $\underline{r} = \underline{r}(X = x)$.

   (b) Compute the following.

$$\tau_x = \sum_{i=1, i \neq n'}^{2(n'+1)} \tau_i x^i + x^{n'} \underline{\widetilde{z}}^\top W_V \underline{\gamma}$$

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

   (c) Set[3] $\pi = (\tau_x, \mu, \underline{\ell}, \underline{r})$ and output π.

The verifier then checks the transcript $\mathcal{T} = (\mathtt{s}; (A_I, A_O, S); (y, z); \underline{T}; x; \pi)$ for rejection using its previous state as follows.

[2] It is important that this x is sampled from $\mathbb{F}^* = \mathbb{F} \setminus \{0\}$, not all of $\mathbb{F}$.

[3] The variable $\widehat{t}$ need not be communicated, as it is re-computed by the verifier anyway.

1. Set $\underline{H}' = \underline{\widetilde{y}}^{-1} \circ \underline{H}$.
2. Compute
$$\widehat{t} = \langle \underline{\ell}, \underline{r} \rangle$$
$$\widetilde{W}_L = \underline{\widetilde{z}}^\top W_L \underline{H}'$$
$$\widetilde{W}_R = (\underline{\widetilde{y}}^{-1} \circ \underline{\widetilde{z}})^\top W_R \underline{G}$$
$$\widetilde{W}_O = \underline{\widetilde{z}}^\top W_O \underline{H}' - \underline{\widetilde{y}}^\top \underline{H}'$$
and, for each $k \in [n_c]$, compute
$$\widetilde{W}_k = Z^\top W_{k,L} \underline{H}'.$$
3. Compute
$$P = \widetilde{W}_O + \sum_{k=1}^{n_c} x^{i_k} \widetilde{W}_k + x^{i_{LR}} (A_I + \widetilde{W}_L + \widetilde{W}_R) + \sum_{k=1}^{n_c} x^{j_k} C_k + x^{i_O} A_O + x^{i_S} S.$$
4. If the following equation does not hold, reject.
$$\widehat{t}G \ + \ \tau_x H \ = \ x^{n'} \left(\left(\delta(y,z) + \underline{\widetilde{z}}^\top \underline{c} \right) G + \underline{\widetilde{z}}^\top W_V \underline{V} \right) \ + \sum_{i=1, i \neq n'}^{2(n'+1)} x^i T_i \quad (1)$$
5. If the following equation does not hold, reject.
$$P = \underline{\ell}^\top \underline{G} + \underline{r}^\top \underline{H}' + \mu H \quad (2)$$
6. Otherwise, accept.

4 Security

We now prove that the Generalized Bulletproofs arithmetic circuit satisfiability proving system has the desired security properties.

Theorem 2. The interactive protocol in Definition 1 has perfect completeness, perfect special honest-verifier zero knowledge, and computational witness-extended emulation.

This theorem is a direct corollary of the following three lemmas.

Lemma 3. The interactive protocol in Definition 1 is perfectly complete.

Proof. Consider a transcript $\mathcal{T}$ generated by an honest prover using a valid statement-witness pair $(\mathtt{s}, \mathtt{w}) \in \mathcal{R}$. For this transcript, we have the following.

$$\underline{\ell}^\top \underline{G} + \underline{r}^\top \underline{H}' + \mu H = \left(\underline{\ell}_{i_{LR}} x^{i_{LR}} + \sum_{k=1}^{n_c} \underline{\ell}_{j_k} x^{j_k} + \underline{\ell}_{i_O} x^{i_O} + \underline{\ell}_{i_S} x^{i_S} \right)^\top \underline{G}$$

$$+ \left(\underline{r}_{j_O} x^{j_O} + \sum_{k \in [n_c]} \left(\underline{r}_{i_k} x^{i_k} + \underline{r}_{j_k} x^{j_k} \right) + \underline{r}_{i_{LR}} x^{i_{LR}} + \underline{r}_{j_S} x^{j_S} \right)^\top \underline{H}' + \mu H$$

With an honest prover, the first term on $\underline{G}$ can be written as follows

$$(\underline{a}_L + W_R^\top (\underline{\widetilde{y}}^{-1} \circ \underline{\widetilde{z}})) x^{i_{LR}} + \sum_k \underline{c}_{k,L} x^{j_k} + \underline{a}_O x^{i_O} + \underline{s}_L x^{i_S}$$

and the second term on $\underline{H}'$ can be written as follows.

$$W_O^\top \underline{\widetilde{z}} - \underline{\widetilde{y}} + \sum_k (W_{k,L} x^{i_k} + \underline{c}_{k,R} x^{j_k}) + (\underline{\widetilde{y}} \circ \underline{a}_R + W_L^\top \underline{\widetilde{z}}) x^{i_{LR}} + \underline{\widetilde{y}} \circ \underline{s}_R x^{i_S}$$

Thus, we have the following.

$$\begin{aligned}
\underline{\ell}^\top \underline{G} + \underline{r}^\top \underline{H}' + \mu H =& (\underline{s}_L^\top \underline{G} + (\underline{\widetilde{y}} \circ \underline{s}_R)^\top \underline{H}') x^{i_S} + \underline{a}_O^\top \underline{G} x^{i_O} \\
&+ (\underline{a}_L^\top \underline{G} + (\underline{\widetilde{y}} \circ \underline{a}_R)^\top \underline{H}') x^{i_{LR}} \\
&+ ((W_R^\top (\underline{\widetilde{y}}^{-1} \circ \underline{\widetilde{z}}))^\top \underline{G} + (W_L^\top \underline{\widetilde{z}})^\top \underline{H}') x^{i_{LR}} \\
&+ \sum_k (\underline{c}_{k,L}^\top \underline{G} + (\underline{\widetilde{y}} \circ \underline{c}_{k,R})^\top \underline{H}') x^{j_k} \\
&+ \sum_k \underline{\widetilde{z}}^\top W_{k,L} \underline{H}' x^{i_k} + (W_O^\top \underline{\widetilde{z}} - \underline{\widetilde{y}})^\top \underline{H}' + \mu H
\end{aligned}$$

The right-hand side can be re-written as follows.

$$\begin{aligned}
&(S - \rho H) x^{i_S} + (A_O - \beta H) x^{i_O} + (A_I + \widetilde{W}_R + \widetilde{W}_L - \alpha H) x^{i_{LR}} \\
&+ \sum_k (C_k - c_k' H) x^{j_k} + \sum_k \widetilde{W}_k x^{i_k} + \widetilde{W}_O + \mu H
\end{aligned}$$

For an honest prover, we have

$$\mu = x^{i_S} \rho + x^{i_O} \beta + x^{i_{LR}} \alpha + \sum_k c_k' x^{j_k}$$

and so the right-hand side is readily seen to match the verifier's point P computed in Step 3.

On the other hand, an honest prover has the polynomial $t(X) = \underline{\ell}(X)^\top \underline{r}(X)$, and $\widehat{t} = t(X = x)$. So we have the following.

$$\begin{aligned}
\widehat{t}G + \tau_x H &= \left(\sum_k x^k \sum_{i+j=k} \underline{\ell}_i^\top \underline{r}_j \right) G + \tau_x H \\
&= \left(x^{n'} t_{n'} + \sum_{k \neq n'} x^k t_k \right) G + \tau_x H \\
&= x^{n'} t_{n'} G + \sum_{k \neq n'} x^k (T_k - \tau_k H) + \tau_x H \\
&= x^{n'} t_{n'} G + \sum_{k \neq n'} x^k T_k + (\tau_x - \sum_{k \neq n'} x^k \tau_k) H
\end{aligned}$$

For an honest prover, we have the following

$$\tau_x = \sum_{k \neq n'} \tau_k x^k + x^{n'} \underline{\widetilde{z}}^\top W_V \underline{\gamma}$$

yielding

$$\widehat{t}G + \tau_x H = x^{n'} (t_{n'} G + \underline{\widetilde{z}}^\top W_V \underline{\gamma} H) + \sum_{k \neq n'} x^k T_k$$

Lastly, note that $t_{n'} = \sum_{i+j=n'} \underline{\ell}_i^\top \underline{r}_j$ so we obtain the following.

$$\begin{aligned}
t_{n'} = &\sum_{i+j=n'} \underline{\ell}_i^\top \underline{r}_j \\
=& \underline{\ell}_{i_O}^\top \underline{r}_{j_O} + \underline{\ell}_{i_{LR}}^\top \underline{r}_{j_{LR}} + \sum_{k \in [n_c]} \underline{\ell}_{j_k}^\top \underline{r}_{i_k} \\
=& \underline{a}_O^\top (W_O^\top \underline{\widetilde{z}} - \underline{\widetilde{y}}) + (\underline{a}_L + W_R^\top (\underline{\widetilde{y}}^{-1} \circ \underline{\widetilde{z}}))^\top (\underline{\widetilde{y}} \circ \underline{a}_R + W_L^\top \underline{\widetilde{z}}) \\
&+ \sum_{k \in [n_c]} \underline{c}_{k,L}^\top (W_{k,L}^\top \underline{\widetilde{z}}) \\
=& \underline{\widetilde{z}}^\top \left(W_L \underline{a}_L + W_R \underline{a}_R + W_O \underline{a}_O + \sum_k W_{k,L} \underline{c}_{k,L} \right) \\
&+ \underline{a}_L^\top (\underline{\widetilde{y}} \circ \underline{a}_R) - \underline{a}_O^\top \underline{\widetilde{y}} + \delta(y,z)
\end{aligned}$$

We always have $\underline{a}_L^\top (\underline{\widetilde{y}} \circ \underline{a}_R) = (\underline{a}_L \circ \underline{a}_R)^\top \underline{\widetilde{y}}$, and for an honest prover, $\underline{a}_O = \underline{a}_L \circ \underline{a}_R$. Moreover, the honest prover has that the quantity in parentheses is exactly $W_V \underline{v} + \underline{c}$.

$$t_{n'} = \underline{\widetilde{z}}^\top (W_V \underline{v} + \underline{c}) + \delta(y,z)$$

Thus, the verification equation 2 is satisfied.

$$\widehat{t}G + \tau_x H = x^{n'} (\underline{\widetilde{z}}^\top W_V \underline{V} + (\delta(y,z) + \underline{\widetilde{z}}^\top \underline{c}) G) + \sum_{k \neq n'} x^k T_k$$

□

The following proof appeared nearly word-for-word in [4], closely following Appendix D in the Bulletproofs preprint, with corresponding changes to accommodate the protocol modifications. The changes are nontrivial, so we include the full proof.

Lemma 4. The interactive protocol in Definition 1 has special honest-verifier zero knowledge[4].

Proof. 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 $\mathcal{R}$, and sample verifier challenges $x, y, z \in \mathbb{F}^*$ uniformly at random. The simulator begins by sampling $\underline{\ell}, \underline{r}$ uniformly at random, after which it computes the following.

$$\widehat{t} = \langle \underline{\ell}, \underline{r} \rangle$$

It then samples $\tau_x \xleftarrow{\$} \mathbb{F}$, and samples all but one $T_k \xleftarrow{\$} \mathbb{G}$, say for each $\frac{n'}{2} \leq k \leq 2n' + 1$ such that $k \neq n'$. With these, then defines

$$T_{2n'+2} = -x^{-2n'-2} \Bigg[x^{n'} \left(\delta(y, z) + \langle \underline{\widetilde{z}}, \underline{c} \rangle \right) G + x^{n'} \underline{\widetilde{z}}^\top W_V \underline{V} \\ + \sum_{k=\frac{n'}{2}, k \neq n'}^{2n'+1} x^k T_k - \widehat{t} G - \tau_x H \Bigg]$$

to satisfy verification equation 1. There is nothing special about $T_{2n'+2}$ here, as the simulator works if all but one T_k is randomly sampled and the missing T_k is defined similarly.

Finally, it computes P from Step 3, samples A_I, A_O, μ uniformly at random, and defines $S = -x^{-i_S}(P - \underline{\ell}^\top \underline{G} - \underline{r}^\top \underline{H}' - \mu H)$ to satisfy Equation 2.

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 τ_i are sampled uniformly at random, so the elements T_i are distributed uniformly over all Pedersen commitments; the uniform sampling of nonzero x and z means τ_x is also distributed uniformly at random. Both $\underline{\ell}$ and $\underline{r}$ are constructed using masking offsets $\underline{s}_L, \underline{s}_R$ 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. □

[4] I need to think about this one more, it is not clear to me if this is perfect, statistical, or computational.

To prove computational witness-extended emulation, we construct a computational extractor $\mathcal{E}$ by nesting the forking algorithm applied to $\mathcal{P}$, yielding linear systems of equations whose solutions yield an extracted witness. We describe the extractor from the inside-out; we first describe the inner forking algorithm, $\mathcal{F}_x$, then the outer forking algorithm, $\mathcal{F}_z$, and then the processing step to finish constructing $\mathcal{E}$.

Technically, a full proof of our claim requires also forking on y with a third application of the forking algorithm, say $\mathcal{F}_y$. However, doing so introduces a triply-indexed tree of transcripts, which rapidly becomes unwieldy; further, a pedagogical understanding of our proof approach does not require fully elaborating on the third fork. See our discussion near the end of our proof of the following lemma.

Lemma 5. The protocol in Definition 1 has computational witness-extended emulation.

Proof. Let $\mathcal{P}$ be an interactive algorithm which inputs a random tape ξ, and interacts with the verifier according to Definition 1, leading to an accepted transcript $\mathcal{T} = (\mathtt{s}; (A_I, A_O, S); (y, z); \underline{T}; x; \pi)$. We construct a computational extractor $\mathcal{E}$ as described above. We distinguish between *transcript indices* due to the forking algorithm and *protocol indices* by subscripting transcript indices with parentheses. Since the prover may not know any witness, we distinguish extracted data with a round overmark as follows.

$$\overset{\frown}{\mathtt{w}} = \left(\overset{\frown}{\underline{a}}_L, \overset{\frown}{\underline{a}}_R, \overset{\frown}{\underline{a}}_O, \left\{(\overset{\frown}{\underline{c}}_{k,L}, \overset{\frown}{\underline{c}}_{k,R})\right\}_{k=1}^{n_c}, \overset{\frown}{\underline{v}}, \overset{\frown}{\underline{\gamma}}, \overset{\frown}{\underline{c}}'\right)$$

The inner forking algorithm, $\mathcal{F}_x$ inputs a random tape $\xi_{\mathcal{P}}$ for $\mathcal{P}$, challenges y and z, and a sequence of some N_x distinct challenges $\left\{x_{(k)}\right\}_{k \in [N_x]}$. If the challenges in the sequence $\left\{x_{(k)}\right\}_{k \in [N_x]}$ are not all distinct, output a distinct failure symbol and terminate. Otherwise, $\mathcal{F}_x$ computes $\mathcal{T}_{(k)} \leftarrow \mathcal{P}(\xi_P, y, z, x_{(k)})$ for each $k \in [N_x]$. Lastly, $\mathcal{F}_x$ outputs $\left\{\mathcal{T}_{(k)}\right\}_{k \in [N_x]}$. We denote this as follows.

$$\left\{\mathcal{T}_k\right\}_{k \in [N_x]} \leftarrow \mathcal{F}_x\left(\xi_{\mathcal{P}}, y, z, \left\{x_{(k)}\right\}_{k \in [N_x]}\right)$$

The outer forking algorithm, $\mathcal{F}_z$, inputs a random tape $\xi_{\mathcal{P}}$ for $\mathcal{P}$, a challenge y, a sequence of some N_z distinct challenges $\left\{z_{(j)}\right\}_{j \in [N_z]}$, and a doubly-indexed sequence of $N_z \cdot N_x$ distinct challenges $\left\{x_{(j,k)}\right\}_{(j,k) \in [N_z] \times [N_x]}$. If the challenges in the sequence $\left\{z_{(j)}\right\}_{j \in [N_z]}$ are not all distinct, or if the challenges in the sequence $\left\{x_{(j,k)}\right\}_{(j,k) \in [N_z] \times [N_x]}$ are not all distinct, then output a distinct failure symbol and terminate. Otherwise, $\mathcal{F}_z$ executes $\mathcal{F}_x(\xi_P, y, z_{(j)}, \left\{x_{(j,k)}\right\}_{k \in [N_x]}$ for each $j \in [N_z]$ to obtain $\left\{\mathcal{T}_{(j,k)}\right\}_{k \in [N_x]}$ for each $j \in [N_z]$. Lastly, $\mathcal{F}_z$ outputs $\left\{\mathcal{T}_{(j,k)}\right\}_{(j,k) \in [N_z] \times [N_x]}$. We denote this as follows.

$$\left\{\mathcal{T}_{(j,k)}\right\}_{(j,k) \in [N_z] \times [N_x]} \leftarrow \mathcal{F}_z\left(\xi_{\mathcal{P}}, y, \left\{z_{(j)}\right\}_{j \in [N_z]}, \left\{x_{(j,k)}\right\}_{(j,k) \in [N_z] \times [N_x]}\right)$$

We now describe how the algorithm $\mathcal{E}$ runs $\mathcal{F}_z$ as a sub-routine and computes the witness. $\mathcal{E}$ inputs a random tape $\xi_{\mathcal{E}}$ and a random tape $\xi_{\mathcal{P}}$ for $\mathcal{P}$. $\mathcal{E}$ samples a challenge y, a sequence $\left\{z_j\right\}_{j\in[N_z]}$, and a sequence $\left\{x_{(j,k)}\right\}_{(j,k)\in[N_z]\times[N_x]}$ from $\xi_{\mathcal{E}}$, re-sampling until all $\left\{z_{(j)}\right\}_{(j)}$ are distinct and all $\left\{x_{(j,k)}\right\}_{(j,k)}$ are distinct. Then $\mathcal{E}$ executes $\mathcal{F}_z$ with input $\left(\xi_{\mathcal{P}}, y, \left\{z_{(j)}\right\}_{j\in[N_z]}, \left\{x_{(j,k)}\right\}_{(j,k)\in[N_z]\times[N_x]}\right)$, resulting in transcripts $\left\{\mathcal{T}_{(j,k)}\right\}_{(j,k)\in[N_z]\times[N_x]}$.

The transcripts $\left\{\mathcal{T}_{(j,k)}\right\}_{(j,k)\in[N_z]\times[N_x]}$ can be visualized as a tree with height 2 with nodes indexed by the pairs (j,k). We append indices (j,k) (including parentheses) to transcript variables, say $A_{I,(j,k)}$, $A_{O,(j,k)}$, and so on. Thus, for each $j \in [N_z]$ and each $k \in [N_x]$, we have the following (admittedly unwieldy) transcript notation.

$$\mathcal{T}_{(j,k)} = \left(\mathtt{s}_{(j,k)}; (A_{I,(j,k)}, A_{O,(j,k)}, S_{(j,k)}); (y, z_{(j)}); \underline{T}_{(j,k)}; x_{(j,k)}; \pi_{(j,k)}\right)$$

We also carry the (j,k) index notation into variables computed directly from transcript data. However, all data before any challenge appears in these transcripts must match across all transcripts by construction. In particular, each transcript contains data satisfying the following for every pair $j, j' \in [N_z]$ and every pair $k, k' \in [N_x]$.

$$\begin{aligned} \mathtt{s}_{(j,k)} &= \mathtt{s}_{(j',k')} \\ A_{I,(j,k)} &= A_{I,(j',k')} \\ A_{O,(j,k)} &= A_{O,(j',k')} \\ S_{(j,k)} &= S_{(j',k')} \end{aligned}$$

Since these terms are independent of (j,k), we drop indices from these variables entirely, keeping our original notation $\mathtt{s}$, A_I, A_O and S for all transcripts. Furthermore, for each $j \in [N_z]$ and each pair $(k,k') \in [N_x]^2$, $\underline{T}_{(j,k)} = \underline{T}_{(j,k')}$, so we drop the index k from these entirely, writing $\underline{T}_{(j)}$. We similarly omit the index k from data which is independent of the challenge x. For example, instead of writing $\widetilde{W}_{O,(j,k)}$, we note that, $\widetilde{W}_O$ is not dependent on x, and therefore is independent of the index k. So, we write $\widetilde{W}_{O,(j)}$ for all transcripts sharing the index j. This makes our transcript notation slightly more compact as follows.

$$\mathcal{T}_{(j,k)} = \left(\mathtt{s}; (A_I, A_O, S); (y, z_{(j)}); \underline{T}_{(j)}; x_{(j,k)}; \pi_{(j,k)}\right)$$

Since $\mathbb{G}$ is cyclic, there exists a pair $\underline{\widehat{v}}, \underline{\widehat{\gamma}} \in \mathbb{F}^m$ such that $\underline{V} = \underline{\widehat{v}}G + \underline{\widehat{\gamma}}H$. $\mathcal{E}$ begins the extraction by solving for these to open $\underline{V}$. If $\underline{V} = \underline{\widehat{v}}G + \underline{\widehat{\gamma}}H$, then verification equation 1 holds in the (j,k)-th transcript as follows.

$$\sum_{i=\frac{n'}{2}}^{2(n'+1)} x^i_{(j,k)} T_{i,(j)} = \widehat{t}_{(j,k)}G + \tau_{x,(j,k)}H$$

$$\text{where } T_{n',(j)} = \left(\delta(y, z_{(j)}) + \underline{\widetilde{z}}^{\top}_{(j)}\left(\underline{c} + W_V\underline{\widehat{v}}\right)\right)G + \underline{\widetilde{z}}^{\top}_{(j)} W_V \underline{\widehat{\gamma}} H$$

Therefore, $\mathcal{E}$ can use a fixed j and take linear combinations of these equalities with arbitrary coefficient sequences $\left\{\nu_{(k)}\right\}_{k\in[N_x]}$.

$$\sum_{k=1}^{N_x} \nu_{(k)} \left(\sum_{i=\frac{n'}{2}}^{2(n'+1)} x^i_{(j,k)} T_{i,(j)} \right) = \sum_{k=1}^{N_x} \nu_{(k)} \left(\widehat{t}_{(j,k)} G + \tau_{x,(j,k)} H \right)$$

On the left-hand side, finite sums of finite sums commute. On the right-hand side, we can collect all terms on G together in one term, and all terms on H in another term. So, for each fixed j, we have the following.

$$\sum_{i=0}^{2(n'+1)} T_{i,(j)} \left(\sum_{k=1}^{N_x} \nu_{(k)} x^i_{(j,k)} \right) = \left(\sum_{k=1}^{N_x} \nu_{(k)} \widehat{t}_{(j,k)} \right) G + \left(\sum_{k=1}^{N_x} \nu_{(k)} \tau_{x,(j,k)} \right) H$$

For each fixed j, for each $\frac{n'}{2} \leq i^* \leq 2n'+2$, $\mathcal{E}$ can, with high probability, find a choice of $\left\{\nu_{(k)}\right\}_{k\in[N_x]}$ so that $\sum_k \nu_{(k)} x^i_{(j,k)} = I(i = i^*)$ to isolate each monomial. Indeed, the following system consists of $\frac{3}{2}n'+2$ equations with N_x unknowns.

$$\sum_{k=1}^{N_x} \nu_{i^*,(j,k)} x^i_{(j,k)} = \begin{cases} 1; & i^* = i \\ 0; & i^* \neq i \end{cases}$$

The set $\left\{x_{(j,k)}\right\}_{k\in[N_x]}$ is sampled uniformly from the N_x-subsets of $\mathbb{F}^*$, so whenever $N_x \geq \frac{3}{2}n'+2$, the corresponding Vandermonde matrix is full rank except with negligible probability. If no solution exists for some i^*, $\mathcal{E}$ begins again by sampling a fresh ξ_P, fresh samples $\left\{z_{(j)}\right\}_j$ and $\left\{x_{(j,k)}\right\}_{(j,k)}$, and tries again with a new transcript tree.

Otherwise, the systems have uniquely determined solutions. With solutions $\left\{\nu_{i^*,(j,k)}\right\}_{k=1}^{N_x}$ in hand for each i^*, we have $\sum_{k=1}^{N_x} \nu_{i^*,(j,k)} x^i_{(j,k)} = I(i = i^*)$, so $\sum_i T_{i,(j)} \sum_k \nu_{(i^*,(j,k)} x^i_{(j,k)} = T_{i^*,(j)}$. Different choices of i^* isolate different monomials to open $\underline{T}$.

$$\left(\sum_{k=1}^{N_x} \nu_{i,(j,k)} \widehat{t}_{(j,k)} \right) G + \left(\sum_{k=1}^{N_x} \nu_{i,(j,k)} \tau_{x,(j,k)} \right) H = T_{i,(j)}$$

$\mathcal{E}$ therefore learns $T_{i,(j)} = \widehat{\,t\,}_{i,(j)} G + \widehat{\,\tau\,}_{i,(j)} H$ for each $i \in [0, 2(n'+1)]$ where

$$\widehat{\,t\,}_{i,(j)} = \sum_{k\in[N_x]} \nu_{i,(j,k)} \widehat{t}_{(j,k)} \text{ and}$$
$$\widehat{\,\tau\,}_{i,(j)} = \sum_{k\in[N_x]} \nu_{i,(j,k)} \tau_{x,(j,k)}.$$

Recalling $t_{n'}$ contains the proof data, we obtain the following.

$$\left(\delta(y, z_{(j)}) + \underline{\widetilde{z}}^\top_{(j)} \left(\underline{c} + W_V \underline{v} \right) \right) G + \underline{\widetilde{z}}^\top_{(j)} W_V \underline{\gamma} H = \widehat{\,t\,}_{n',(j)} G + \widehat{\,\tau\,}_{n',(j)} H, \text{ so}$$

$$\left(\widehat{t}_{n',(j)} - \delta(y, z_{(j)}) - \widetilde{\underline{z}}_{(j)}^{\top}\underline{c} - \widetilde{\underline{z}}_{(j)}^{\top} W_V \widehat{\underline{v}}\right) G + \left(\widehat{\tau}_{n',(j)} - \widetilde{\underline{z}}_{(j)}^{\top} W_V \widehat{\underline{\gamma}}\right) H = O$$

These coefficients are equivalence classes of elements from $\mathbb{F}$, but both have integer representatives in the half-open interval $[0, \mathrm{ord}(\mathbb{G}))$. If both coefficients are non-zero, then this equality defines a non-trivial discrete logarithm relationship between G, and H (which can be used to break binding of Pedersen commitments). Otherwise, one coefficient is zero (modulo the group order). In this case, both must be zero. Indeed, if the equality can be written $t^* G + 0H = O$ for some $t^* \in [0, \mathrm{ord}(\mathbb{G}) - 1]$, then $t^* G = O$, which implies that $t^* \equiv 0$ modulo $\mathrm{ord}(\mathbb{G})$, so $t^* = 0$. Similarly, if the equality can be written $0G + \tau^* H = O$ for some $\tau^* \in [0, \mathrm{ord}(\mathbb{G}) - 1]$, then $\tau^* = 0$. Thus, both coefficients are zero and we obtain the following for each $j \in [N_z]$.

$$\widehat{t}_{n',(j)} = \delta(y, z_{(j)}) + \widetilde{\underline{z}}_{(j)}^{\top} \left(\underline{c} + W_V \widehat{\underline{v}}\right)$$

$$\widehat{\tau}_{n',(j)} = \widetilde{\underline{z}}_{(j)}^{\top} W_V \widehat{\underline{\gamma}}$$

Of course, $\mathcal{E}$ has learned the left-hand side of these equations from the transcripts, but does not know $\widehat{\underline{v}}$ or $\widehat{\underline{\gamma}}$ yet. For convenience, pack the left-hand side of these equations into vectors as follows.

$$\begin{aligned}\widehat{\underline{t}}_{n'} &= (\widehat{t}_{n',(j)})_{j\in[N_z]}^{\top} \in \mathbb{F}^{N_z}\\ \widehat{\underline{\tau}}_{n'} &= (\widehat{\tau}_{n',(j)})_{j\in[N_z]}^{\top} \in \mathbb{F}^{N_z}\end{aligned}$$

Also define the following.

$$\begin{aligned}\underline{\delta} &= (\delta(y, z_{(j)}) + \widetilde{\underline{z}}_{(j)}^{\top}\underline{c})_{j\in[N_z]}^{\top} \in \mathbb{F}^{N_z}\\ Z &= (\widetilde{\underline{z}}_{(1)} \mid \widetilde{\underline{z}}_{(2)} \mid \cdots \mid \widetilde{\underline{z}}_{(N_z)}) \in \mathbb{F}^{Q\times N_z}\end{aligned}$$

Then $\widehat{\underline{v}}$ and $\widehat{\underline{\gamma}}$ satisfy the following.

$$\begin{aligned}\widehat{\underline{t}}_{n'} =& \underline{\delta} + Z^{\top}\left(\underline{c} + W_V \widehat{\underline{v}}\right)\\ \widehat{\underline{\tau}}_{n'} =& Z^{\top} W_V \widehat{\underline{\gamma}}\end{aligned}$$

To solve for the witness data $\widehat{\underline{v}}$ and $\widehat{\underline{\gamma}}$, first re-arrange terms.

$$\begin{aligned}Z^{\top} W_V \widehat{\underline{v}} &= \widehat{\underline{t}}_{n'} - \underline{\delta} - Z^{\top}\underline{c}\\ Z^{\top} W_V \widehat{\underline{\gamma}} &= \widehat{\underline{\tau}}_{n'}\end{aligned}$$

Next, define the matrix $M = W_V^{\top} Z Z^{\top} W_V$. Recall W_V has dimensions Q-by-m with rank m. Note Z has dimensions Q-by-N_z and, since these z are sampled independently and uniformly, Z has full rank with high probability. Thus, $Z^{\top} W_V$ is an N_z-by-m matrix. If $N_z = m$, then $Z^{\top} W_V$ has rank m, so $(Z^{\top} W_V)^{\top} Z^{\top} W_V$ has dimensions m-by-m and has rank m. Thus, $(Z^{\top} W_V)^{\top} Z^{\top} W_V = M$ is invertible. Thus, by multiplying both sides of our system of equations by $(Z^\top W_V)^\top = W_V^\top Z$, we get M on the left-hand side.

$$\begin{aligned}W_V^\top Z Z^\top W_V \widehat{\underline{v}} =& W_V^\top Z \left(\widehat{\underline{t}}_{n'} - \underline{\delta} - Z^\top \underline{c}\right)\\ W_V^\top Z Z^\top W_V \widehat{\underline{\gamma}} =& W_V^\top Z \widehat{\underline{\tau}}_{n'}\end{aligned}$$

So $\mathcal{E}$ simply multiplies on the left by the inverse to extract $\widehat{\underline{v}}$ and $\widehat{\underline{\gamma}}$ as promised.

$$\begin{aligned}\widehat{\underline{v}} =& M^{-1} W_V^\top Z \left(\widehat{\underline{t}}_{n'} - \underline{\delta} - Z^\top \underline{c}\right)\\ \widehat{\underline{\gamma}} =& M^{-1} W_V^\top Z \widehat{\underline{\tau}}_{n'}\end{aligned}$$

Next, $\mathcal{E}$ opens the Pedersen vector commitments $\{C_k\}_{k\in[n_c]}$ from verification equation 2 as follows, repeating a similar procedure to isolate coefficients on the monomials of x. We obtain the following for each $j \in [N_z]$ by looking at the monomials on the right-hand side of verification equation 2

$$\begin{aligned}\sum_{k\in[N_x]} \nu_{j_O,(j,k)} P_{(j,k)} =& \widetilde{W}_{O,(j)} - \underline{\widetilde{y}}^\top \underline{H}' & \sum_{k\in[N_x]} \nu_{i_O,(j,k)} P_{(j,k)} =& A_O\\ \sum_{k\in[N_x]} \nu_{i_{LR},(j,k)} P_{(j,k)} =& A_I + \widetilde{W}_{R,(j)} + \widetilde{W}_{L,(j)} & \sum_{k\in[N_x]} \nu_{i_S,(j,k)} P_{(j,k)} =& S\end{aligned}$$

as well as the following for each $k' \in [n_c]$.

$$\sum_{k\in[N_x]} \nu_{i_{k'},(j,k)} P_{(j,k)} = \widetilde{W}_k \qquad \sum_{k\in[N_x]} \nu_{j_{k'},(j,k)} P_{(j,k)} = C_k$$

Since $P_{(j,k)} = \underline{\ell}_{(j,k)}^{\top}\underline{G} + \underline{r}_{(j,k)}^{\top}\underline{H}' + \mu_{(j,k)}H$, define the following.

$$\begin{aligned}
\widehat{\underline{a}}_L =& \sum_k \nu_{i_{LR},(j,k)}\underline{\ell}_{(j,k)} - W_R^{\top}(\underline{\widetilde{y}}_{(j)}^{-1} \circ \underline{\widetilde{z}}_{(j)}) \\
\widehat{\underline{a}}_R =& \underline{\widetilde{y}}^{-1} \circ \left(\sum_k \nu_{i_{LR},(j,k)}\underline{r}_{(j,k)} - W_L^{\top}\underline{\widetilde{z}}_{(j)} \right) \\
\widehat{\alpha} =& \sum_k \nu_{i_{LR},(j,k)}\mu_{(j,k)} \\
\widehat{\underline{b}}_L =& \sum_k \nu_{i_O,(j,k)}\underline{\ell}_{(j,k)} \\
\widehat{\underline{b}}_R =& \underline{\widetilde{y}}^{-1} \circ \sum_k \nu_{i_O,(j,k)}\underline{r}_{(j,k)} \\
\widehat{\beta} =& \sum_k \nu_{i_O,(j,k)}\mu_{(j,k)} \\
\widehat{\underline{s}}_L =& \sum_k \nu_{i_S,(j,k)}\underline{\ell}_{(j,k)} \\
\widehat{\underline{s}}_R =& \underline{\widetilde{y}}^{-1} \circ \sum_k \nu_{i_S,(j,k)}\underline{r}_{(j,k)} \\
\widehat{\rho} =& \sum_k \nu_{i_S,(j,k)}\mu_{(j,k)} \\
\widehat{\underline{c}}_{k',L} =& \sum_{k \in [N_x]} \nu_{j_{k'},(j,k)}\underline{\ell}_{(j,k)} \\
\widehat{\underline{c}}_{k',R} =& \underline{\widetilde{y}}^{-1} \circ \sum_{k \in [N_x]} \nu_{j_{k'},(j,k)}\underline{r}_{(j,k)} \\
\widehat{c}'_{k'} =& \sum_{k \in [N_x]} \nu_{i_{k'},(j,k)}\mu_{(j,k)}
\end{aligned}$$

These extractions satisfy the following by construction for each $k \in [n_c]$ as follows.

$$\begin{aligned}
A_I =& \widehat{\underline{a}}_L^{\top}\underline{G} + \widehat{\underline{a}}_R^{\top}\underline{H} + \widehat{\alpha}H \\
A_O =& \widehat{\underline{b}}_L^{\top}\underline{G} + \widehat{\underline{b}}_R^{\top}\underline{H} + \widehat{\beta}H \\
S =& \widehat{\underline{s}}_L^{\top}\underline{G} + \widehat{\underline{s}}_R^{\top}\underline{H} + \widehat{\rho}H \\
C_k =& \widehat{\underline{c}}_{k,L}^{\top}\underline{G} + \widehat{\underline{c}}_{k,R}^{\top}\underline{H} + \widehat{c}'_k
\end{aligned}$$

That is to say, they open the commitments by construction. Thus, if the extractions differ for any $j \neq j' \in [N_z]$, then we learn a discrete logarithm relation between the generators $\underline{G}$, $\underline{H}$, and H. Otherwise, they match across all choices j; hence, we omit the fork index j from our notation.

Now $\mathcal{E}$ defines $\widehat{\underline{a}}_O = \widehat{\underline{b}}_L$ and has the candidate witness as follows.

$$\widehat{\mathtt{w}} = \left(\widehat{\underline{a}}_L, \widehat{\underline{a}}_R, \widehat{\underline{a}}_O, \left\{(\widehat{\underline{c}}_{k,L}, \widehat{\underline{c}}_{k,R})\right\}_{k=1}^{n_c}, \widehat{\underline{v}}, \widehat{\underline{\gamma}}, \widehat{\underline{c}}'\right)$$

All that remains is to show $\widehat{\mathtt{w}}$ satisfies the following.

$$\widehat{\underline{a}}_O = \widehat{\underline{a}}_L \circ \widehat{\underline{a}}_R \tag{3}$$

$$W_V \widehat{\underline{v}} + \underline{c} = W_L \widehat{\underline{a}}_L + W_R \widehat{\underline{a}}_R + W_O \widehat{\underline{a}}_O + \sum_{k \in [n_c]} W_{k,L} \widehat{\underline{c}}_{k,L} \tag{4}$$

Each valid transcript satisfies the following in order to pass verification.

$$\begin{aligned} P_{(j,k)} =& \widetilde{W}_{O,(j)} + \sum_k x^{i_k} \widetilde{W}_{k,(j)} + x^{i_{LR}}(A_I + \widetilde{W}_{L,(j)} + \widetilde{W}_{R,(j)}) \\ &+ \sum_k x^{j_k} C_k + x^{i_O} A_O + x^{i_S} S \end{aligned}$$

Substitute our extractions into the right-hand side of this equation, and recall $P_{(j,k)} = \underline{\ell}^\top_{(j,k)} \underline{G} + \underline{r}^\top_{(j,k)} \underline{H}' + \mu_{(j,k)} H$ on the left to collect terms generator by generator. For $\widehat{\underline{\ell}}_{(j,k)}$, we obtain the following.

$$\widehat{\underline{\ell}}_{(j,k)} = x^{i_{LR}} \left(\widehat{\underline{a}}_L + W_R^\top (\widetilde{\underline{y}}^{-1} \circ \widetilde{\underline{z}})\right) + x^{i_O} \widehat{\underline{a}}_O + x^{i_S} \widehat{\underline{s}}_L + \sum_k x^{j_k} \widehat{\underline{c}}_{k,L}$$

And for $\widehat{\underline{r}}_{(j,k)}$, we get the following.

$$\begin{aligned} \widehat{\underline{r}}_{(j,k)} =& W_O^\top \widetilde{\underline{z}} + \sum_k x^{i_k} W_{k,L}^\top \widetilde{\underline{z}}_{(j)} \\ &+ \widetilde{\underline{y}} \circ \left(x^{i_{LR}} \left(\widehat{\underline{a}}_R + W_R^\top \widetilde{\underline{z}}_{(j)} \right) + \sum_k x^{j_k} \widehat{\underline{c}}_{k,R} + x^{i_O} \widehat{\underline{b}}_R + x^{i_S} \widehat{\underline{s}}_R \right) \end{aligned}$$

Note $\widehat{\underline{\ell}}^\top_{(j,k)} \widehat{\underline{r}}_{(j,k)}$ is a degree $2(n'+1)$ polynomial whose $(n')^{th}$ coefficient $\widehat{t}_{n'}$ is the sum of terms whose monomial exponents sum to n'. The indices of the coefficients which do so are exactly $i_O + j_O = i_{LR} + j_{LR} = i_k + j_k = n'$.

$$\begin{aligned} \widehat{t}_{n'} =& \widehat{\underline{a}}_O^\top W_O^\top \widetilde{\underline{z}}_{(j)} + \sum_k \widehat{\underline{c}}_{k,L}^\top W_{k,L}^\top \widetilde{\underline{z}}_{(j)} \\ &+ \left(\widehat{\underline{a}}_L + W_R^\top (\widetilde{\underline{y}}^{-1} \circ \widetilde{\underline{z}})\right)^\top \left(\widetilde{\underline{y}} \circ \widehat{\underline{a}}_R + W_L^\top \widetilde{\underline{z}}\right) \\ =& \widetilde{\underline{z}}_{(j)}^\top \left(W_L \widehat{\underline{a}}_L + W_R \widehat{\underline{a}}_R + W_O \widehat{\underline{a}}_O + \sum_{k \in [n_c]} W_{k,L} \widehat{\underline{c}}_{k,L} \right) \\ &+ \delta(y, z_{(j)}) + \left(\widehat{\underline{a}}_O - \widehat{\underline{a}}_L \circ \widehat{\underline{a}}_R\right)^\top \widetilde{\underline{y}} \end{aligned}$$

On the other hand, we saw $\widehat{t}_{n'} = \delta(y, z_{(j)}) + \widetilde{\underline{z}}_{(j)}^{\top}\underline{c} + \widetilde{\underline{z}}_{(j)}^{\top} W_V \widehat{\underline{v}}$. Setting these representations equal and re-arranging terms, we obtain the following.

$$\begin{aligned}&\left(\widehat{\underline{a}}_O - \widehat{\underline{a}}_L \circ \widehat{\underline{a}}_R\right)^{\top} \widetilde{\underline{y}} \\ &+ \widetilde{\underline{z}}_{(j)}^{\top} \left(W_L \widehat{\underline{a}}_L + W_R \widehat{\underline{a}}_R + W_O \widehat{\underline{a}}_O + \sum_{k \in [n_c]} W_{k,L} \widehat{\underline{c}}_{k,L} - W_V \widehat{\underline{v}} - \underline{c} \right) = 0\end{aligned}$$

This holds for each $\widetilde{\underline{z}}_{(j)}$, and this expression describes a polynomial in y.

Just as before, we can fork on y, obtaining an even larger transcript tree, and then find coefficients so that linear combinations of these equations can isolate each monomial in y. The method proceeds similarly to the way we nested the prover $\mathcal{P}$ within $\mathcal{F}_x$, which was then nested within $\mathcal{F}_z$. A full explanation of the method requires adding another layer of nesting, and triply-indexing the transcript tree instead of doubly-indexing.

The method is a straightforward extension of the approach described thus far, and reveals that each coefficient is zero as follows, but a full description from here brings little pedagogical benefit.

$$W_L \widehat{\underline{a}}_L + W_R \widehat{\underline{a}}_R + W_O \widehat{\underline{a}}_O + \sum_{k \in [n_c]} W_{k,L} \widehat{\underline{c}}_{k,L} = W_V \widehat{\underline{v}} + \underline{c}, \text{ and} \tag{5}$$

$$\widehat{\underline{a}}_L \circ \widehat{\underline{a}}_R = \widehat{\underline{a}}_O \tag{6}$$

□

4.1 Problems with previous versions

The protocol described and proven secure in this report differs from previous Generalized Bulletproofs designs. Specifically, Pedersen vector commitments in the original design are with respect to $\underline{G}, H$; in ours, they are with respect to $\underline{G}, \underline{H}, H$. Moreover, the protocol indices described in Table 1 differ from the most recent previous version of this protocol, whose indices appear in Table 3. These changes enable the proof of witness-extended emulation to go through.

In the last iteration of this protocol, the protocol parameter $n' = 2 + 2\lfloor \frac{n_c}{2} \rfloor$ and used indices as described in Table 3, where i indices and j indices overlap, and indices in parentheses were only defined in the case that $n_c \equiv 1$ modulo 2. This definition lead naturally to vectors $\underline{\ell}$ and $\underline{r}$ as in Table 4 when computed by an honest prover.

0	1	$\cdots$	$\frac{n'}{2} - 1$	$\frac{n'}{2}$	$\frac{n'}{2} + 1$	$\cdots$	$n' - 1$	n'	$n' + 1$
i_1	i_2	$\cdots$	$i_{\frac{n'}{2}-1}$	i_{LR}	$i_{\frac{n'}{2}}$	$\cdots$	$(i_{n'-1})$	i_O	i_S
j_O	$(j_{n'-1})$	$\cdots$	$j_{\frac{n'}{2}}$	j_{LR}	$j_{\frac{n'}{2}-1}$	$\cdots$	j_2	j_1	j_S

Table 3: Previous protocol indices which lead to a bad extracted witness.

Index k	$\underline{\ell}_k$	$\underline{r}_k$
0	$\underline{c}_{1,L}$	$W_O^\top \widetilde{\underline{z}} - \widetilde{\underline{y}}$
1	$\underline{c}_{2,L}$	$W_{n'-1,L}^\top \widetilde{\underline{z}}$
$\cdots$	$\cdots$	$\cdots$
$\frac{n'}{2}-1$	$\underline{c}_{\frac{n'}{2},L}$	$W_{\frac{n'}{2}+1,L}^\top \widetilde{\underline{z}}$
$\frac{n'}{2}$	$\underline{a}_L + W_R^\top(\widetilde{\underline{y}}^{-1} \circ \widetilde{\underline{z}})$	$\widetilde{\underline{y}} \circ \underline{a}_R + W_L^\top \widetilde{\underline{z}}$
$\frac{n'}{2}+1$	$\underline{c}_{\frac{n'}{2}+1,L}$	$W_{\frac{n'}{2},L}^\top \widetilde{\underline{z}}$
$\cdots$	$\cdots$	$\cdots$
$n'-1$	$\underline{c}_{n'-1,L}$	$W_{2,L}^\top \widetilde{\underline{z}}$
n'	$\underline{a}_O$	$W_{1,L}^\top \widetilde{\underline{z}}$
$n'+1$	$\underline{s}_L$	$\widetilde{\underline{y}} \circ \underline{s}_R$

Table 4: Previous $\underline{\ell}$ and $\underline{r}$ vectors which lead to a bad extracted witness.

Applying the extractor as described above provides the opening data $\underline{s}_L$, $\underline{s}_R$, α, β, ρ, τ_i for the commitments A_I, A_O, S, T_i and extracts data $\widehat{\underline{a}}_L$, $\widehat{\underline{a}}_R$, $\widehat{\underline{a}}_O$, $\widehat{\underline{c}}_{k,L}$, $\widehat{\underline{c}}_{k,R}$, $\widehat{\underline{v}}$, $\widehat{\underline{\gamma}}$, and $\widehat{\underline{c}}'$. Each of these is a linear combination of transcript data. Moreover, the extracted data satisfy the following thanks to the verification equation implied by this approach.

$$\begin{aligned}
\underline{\ell} =& x^{n'+1}\widehat{\underline{s}}_L + +x^{n'}\widehat{\underline{a}}_O + x^{n'/2}(\underline{a}_L + W_R^\top(\widetilde{\underline{y}}^{-1} \circ \widetilde{\underline{z}})) + \sum_k x^{k-1}\widehat{\underline{c}}_{k,L} \\
\underline{r} =& \widetilde{\underline{y}} \circ \widehat{\underline{c}}_{1,R} + W_O^\top \widetilde{\underline{z}} - \widetilde{\underline{y}} + x(\widetilde{\underline{y}} \circ \widehat{\underline{c}}_{2,R} + W_{n'-1,L}^\top \widetilde{\underline{z}}) + \cdots \\
&+ x^{n'/2-1}(\widetilde{\underline{y}} \circ \widehat{\underline{c}}_{n'/2,R} + W_{n'/2+1,L}^\top \widetilde{\underline{z}}) + x^{n'/2}(\widetilde{\underline{y}} \circ \widehat{\underline{a}}_R + W_L^\top \widetilde{\underline{z}}) \\
&+ x^{n'/2+1}(\widetilde{\underline{y}} \circ \widehat{\underline{c}}_{n'/2+1,R} + W_{n'/2,L}^\top \widetilde{\underline{z}}) \\
&+ \cdots + x^{n'-1}(\widetilde{\underline{y}} \circ \widehat{\underline{c}}_{n'-1,R} + W_{2,L}^\top \widetilde{\underline{z}}) + x^{n'} W_{1,L}^\top \widetilde{\underline{z}} + x^{n'+1}(\widetilde{\underline{y}} \circ \widehat{\underline{s}}_R)
\end{aligned}$$

Note that the extracted data $\widehat{\underline{c}}_{k,R}$ in $\underline{r}$ appear here at indices corresponding to locations in $\underline{\ell}$ where the extracted data $\widehat{\underline{c}}_{n'+1-k,L}$ is stored, introducing cross terms in the computation of the inner product. In particular, the n'-th coefficient on x in the inner product $\underline{\ell}^\top \underline{r}$ is as follows.

$$\begin{aligned}
t_{n'} =& \langle \widetilde{\underline{z}}, W_L\underline{a}_L + W_R\underline{a}_R + W_O\underline{a}_O + \sum_{k \in [n_c]} W_{k,L}\widehat{\underline{c}}_{k,L} \rangle + \delta(\widetilde{\underline{y}}, \widetilde{\underline{z}}) \\
&+ \langle \underline{a}_O - \underline{a}_L \circ \underline{a}_R)^\top \widetilde{\underline{y}} + (\widehat{\underline{c}}_{2,L} \circ \widehat{\underline{c}}_{n'-1,R} + \cdots + \widehat{\underline{c}}_{n'-1,L} \circ \widehat{\underline{c}}_{2,R} + \underline{a}_O \circ \underline{c}_{1,R}, \widetilde{\underline{y}} \rangle
\end{aligned}$$

The cross-terms appear when multiplying the k-th term of $\underline{\ell}$ against the $(n'-k)$-th term of r. These cross-terms appear because each x^k with $1 \le k < n'/2$ contributes a term of the following form

$$(x^k \widehat{\underline{c}}_{k+1,L})^\top (x^{n'-k}(\widetilde{\underline{y}} \circ \widehat{\underline{c}}_{n'-k,R} + W_{k+1,L}^\top \widetilde{\underline{z}}))$$

and each x^k with $n'/2 < k \leq n'-1$ contributes a term of the following form

$$(x^k \underline{\widehat{c}}_{k,L})^\top (x^{n'+1-k}(\underline{\widetilde{y}} \circ \underline{\widehat{c}}_{n'+1-k,R} + W_{k,L}^\top z))$$

On the other hand, in order to pass verification, we must have

$$t_{n'} = \delta(\underline{\widetilde{y}}, \underline{\widetilde{z}}) + \langle \underline{\widetilde{z}}, \underline{\widehat{c}} + W_V \underline{v} \rangle$$

so setting these two representations of $t_{n'}$ equal, we obtain the following.

$$\langle \underline{\widetilde{z}}, W_L\underline{a}_L + W_R\underline{a}_R + W_O\underline{a}_O + \sum_k W_{k,L}\underline{\widehat{c}}_{k,L} - W_V\underline{v} - \underline{\widehat{c}} \rangle$$

$$= \langle \underline{\widetilde{y}}, \underline{a}_O - \underline{a}_L \circ \underline{a}_R + \underline{a}_O \circ \underline{c}_{1,R} + \sum_{k=2}^{n'-1} \underline{\widehat{c}}_{k,L} \circ \underline{\widehat{c}}_{n'+1-k,R} \rangle$$

From here we can make an argument that both the following hold.

$$W_L\underline{a}_L + W_R\underline{a}_R + W_O\underline{a}_O + \sum_k W_{k,L}\underline{\widehat{c}}_{k,L} - W_V\underline{v} - \underline{\widehat{c}} = 0 \text{ and}$$

$$\underline{a}_O - \underline{a}_L \circ \underline{a}_R + \underline{a}_O \circ \underline{c}_{1,R} + \sum_{k=2}^{n'-1} \underline{\widehat{c}}_{k,L} \circ \underline{\widehat{c}}_{n'+1-k,R} = 0$$

However, since $\underline{a}_L \circ \underline{a}_R - \underline{a}_O = \underline{a}_O \circ \underline{c}_{1,R} + \sum_{k=2}^{n'-1} \underline{\widehat{c}}_{k,L} \circ \underline{\widehat{c}}_{n'+1-k,R}$ is nonzero in general, the version of the protocol using the indices of Table 3 and the vectors of Table 4 do not yield the arithmetic circuit correctness constraint $\underline{a}_L \circ \underline{a}_R = \underline{a}_O$.

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`.

[3] Simon Kamp. Generalized bulletproofs. GitHub repository, 2023. `https://github.com/simonkamp/curve-trees/blob/main/bulletproofs/generalized-bulletproofs.md`.

[4] Cypher Stack. Generalized bulletproofs. GitHub repository, 2025. `https://github.com/kayabaNerve/monero-oxide/blob/fcmp%2B%2B/audits/generalized-bulletproofs/Updated%20Security%20Proofs.pdf`.