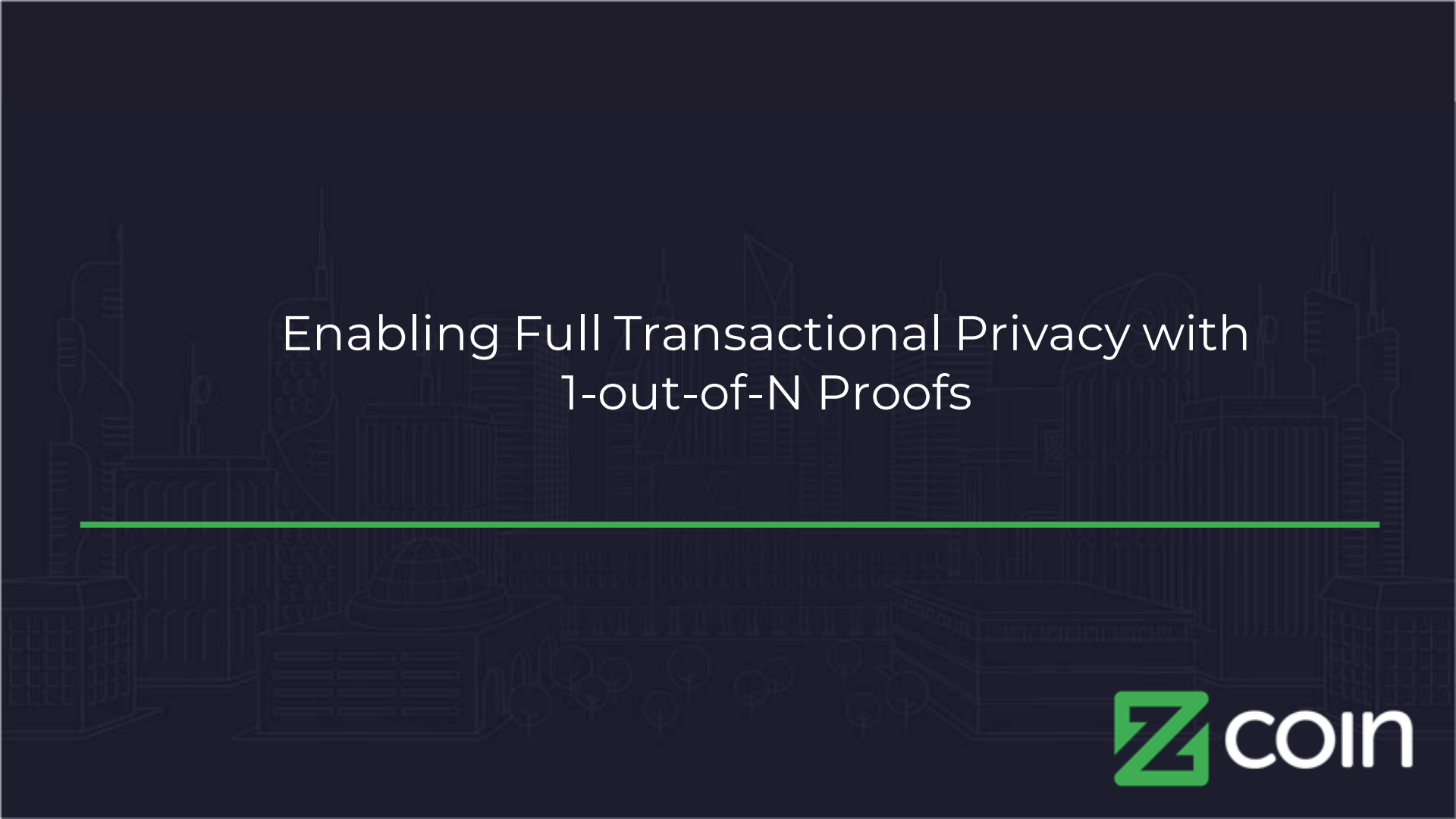
Enabling Full Transactional Privacy with 1-out-of-N Proofs



INTRODUCTION

Cryptocurrency payments to be truly private, transactions have to have three properties:

1. Un-Traceability

   hiding the identities of the sender / the transaction origins

2. Confidentiality,

   hiding the transferred amounts

3. Anonymity

   hiding the recipients identity

CRYPTO BACKGROUND: COMMITMENTS

Double-blinded Homomorphic commitments

$$Comm_{ck}(m; r_1, r_2) = g^m h_1^{r_1} h_2^{r_2}$$

$$Comm_{ck}(m; r_1, r_2) + Comm_{ck}(m'; r_1', r_2') = Comm_{ck}(m + m'; r_1 + r_1', r_2 + r_2')$$

The (Generalized)Pedersen commitment scheme

$$Com_{ck}(m; r) = g^m \cdot h^r$$

$$Com(m_1, m_2, \cdots m_n; r) = h^r g_1^{m_1} g_2^{m_2} \cdots g_n^{m_n}$$

Coins As Double-blinded Commitments

Coins are double blinded commitments:

$$C = Comm(S; V, R) = g^S h_1^V h_2^R$$

- S is unique coin serial number which is revealed during SPEND
- V is the coin hidden value (within a range $[0, 2^{64} - 1)$)
- R is the random blinding factor. It prevents identification of the coin after S is revealed (as V can be easily brute forced)

Transactions Hiding the Values and Origins

We assume each transaction can spend N_{old} coins and output N_{new} coins

1. N_{old} Input Coins:

$$C_{I1} = g^{S_{I1}} h_1^{V_{I1}} h_2^{R_{I1}}, \ldots, C_{IN_{old}} = g^{S_{IN_{old}}} h_1^{V_{IN_{old}}} h_2^{R_{IN_{old}}}$$

2. N_new Output Coins:

$$C_{O1} = g^{S_{O1}} h_1^{V_{O1}} h_2^{R_{O1}}, \ldots, C_{ON_{new}} = g^{S_{ON_{new}}} h_1^{V_{ON_{new}}} h_2^{R_{ON_{new}}}$$

Enabling Full Transactional Privacy

Transaction owner should PROVE that

1. All Input Spends are valid without revealing their origins (Via 1-out-of-N Proofs)

2. Balance is preserved without revealing any input or output coin value

$$V_{I1} + \ldots + V_{IN_{old}} = V_{O1} + \ldots + V_{ON_{new}}$$

3. No Output Coin contains a negative value (Bulletproofs)

4. Output Coins can be spent only by the intended recipients

Proof of Valid Spends via 1-out-of-N Proofs

Initial Set of All Coins is $(C_0, C_1, \ldots, C_N)$

For each input coin

- Prover reveals the coin's serial number S.
- Prover parses the initial set of all commitments $(C_0, C_1, \ldots, C_N)$ and computes $C_i = C_i \cdot Comm(S; 0,0)^{-1}$
- Prover provides a non-interactive 1-oo-N Proof for a double blinded commitment opening to 0 for the new set $(C_0, C_1, \ldots, C_N)$.

$P(gk, crs, (C_0, \ldots, C_{N-1}), l, V, R)$

Compute

$r_A, r_B, r_C, r_D, a_{j,1}, \ldots, a_{j,n-1} \leftarrow_R \mathbb{Z}_q$

for $j \in [0, \cdots, m-1]$

$a_{j,0} = -\sum_{i=1}^{n-1} a_{j,i}$

$B := Com_{ck}(\sigma_{l_0,0}, \ldots, \sigma_{l_{m-1},n-1}; r_B)$

$A := Com_{ck}(a_{0,0}, \ldots, a_{m-1,n-1}; r_A)$

$C := Com_{ck}(\{a_{j,i}(1 - 2\sigma_{l_j,i})\}_{j,i=0}^{m-1,n-1}; r_C)$

$D := Com_{ck}(-a_{0,0}^2, \ldots, -a_{m-1,n-1}^2; r_D)$

$For \quad k \in 0, ..., m-1$

$\rho_k, \tau_k \leftarrow_R \mathbb{Z}_q$

$G_k = \prod_{i=0}^{N-1} C_i^{p_{i,k}}$

computing $p_{i,k}$ as is described above

$Q_k = Comm(0, \rho_k, \tau_k)$

$\forall j \in [0, m-1], i \in [1, n-1]$

$f_{j,i} = \sigma_{l_j i} x + a_{j,i}$

$z_A = r_B \cdot x + r_A$

$z_C = r_C \cdot x + r_D$

$z_V = V \cdot x^m - \sum_{k=0}^{m-1} \rho_k \cdot x^k$

$z_R = R \cdot x^m - \sum_{k=0}^{m-1} \tau_k \cdot x^k$

$\mathcal{P}_2(gk, crs, (c_0, \ldots, c_{N-1}), (\ell, r))$

$r_B, \rho_k \leftarrow \mathbb{Z}_q$

$B := \mathrm{Com}_{ck}(\delta_{\ell_{0,0}}, \ldots, \delta_{\ell_{m-1,n-1}}; r_B)$

$(A, C, D) \leftarrow \mathcal{P}_1(gk, crs, B, (\{\delta_{\ell_j,i}\}_{j,i=0}^{m-1,n-1}, r_B))$

For $k = 0, \ldots, m-1$

$G_k = \prod_{i=0}^{N-1} c_i^{p_{i,k}} \cdot Com(0; \rho_k)$

using $p_{i,k}$ from (1)

$(f_{0,1}, \ldots, f_{m-1,n-1}, z_A, z_C) \leftarrow \mathcal{P}_1(x)$

$z := r x^m - \sum_{k=0}^{m-1} \rho_k x^k$

$P(gk, crs, (C_0, \ldots, C_{N-1}), l, V, R)$

Compute

$r_A, r_B, r_C, r_D, a_{j,1}, \ldots, a_{j,n-1} \leftarrow_R \mathbb{Z}_q$

for $j \in [0, \cdots, m-1]$

$a_{j,0} = -\sum_{i=1}^{n-1} a_{j,i}$

$B := Com_{ck}(\sigma_{l_0,0}, \ldots, \sigma_{l_{m-1},n-1}; r_B)$

$A := Com_{ck}(a_{0,0}, \ldots, a_{m-1,n-1}; r_A)$

$C := Com_{ck}(\{a_{j,i}(1 - 2\sigma_{l_j,i})\}_{j,i=0}^{m-1,n-1}; r_C)$

$D := Com_{ck}(-a_{0,0}^2, \ldots, -a_{m-1,n-1}^2; r_D)$

$For \quad k \in 0, ..., m-1$

$\rho_k, \tau_k \leftarrow_R \mathbb{Z}_q$

$G_k = \prod_{i=0}^{N-1} C_i^{p_{i,k}}$

computing $p_{i,k}$ as is described above

$Q_k = Comm(0, \rho_k, \tau_k)$

$\forall j \in [0, m-1], i \in [1, n-1]$

$f_{j,i} = \sigma_{l_j i} x + a_{j,i}$

$z_A = r_B \cdot x + r_A$

$z_C = r_C \cdot x + r_D$

$z_V = V \cdot x^m - \sum_{k=0}^{m-1} \rho_k \cdot x^k$

$z_R = R \cdot x^m - \sum_{k=0}^{m-1} \tau_k \cdot x^k$

$V(gk, crs, (C_0, \ldots, C_{N-1}))$

Accept if and only if

$A, B, C, D, \{G_K, Q_K\}_{k=0}^{m-1}$ →

← $x \leftarrow \{0,1\}^\lambda$

The values

$A, B, C, D, G_0, Q_0 \ldots, G_{m-1}, Q_{m-1} \in G$

$\{f_{j,i}\}_{j,i=0,1}^{m-1,n-1}, z_A, z_C, z_V, z_R \in \mathbb{Z}_q$

$\forall j : f_{j,0} = x - \sum_{i=1}^{n-1} f_{j,i}$

$B^x A = Com(f_{0,0}, \ldots, \ldots f_{m-1,n-1}; z_A)$

$C^x D = Com(\{f_{j,i}(x - f_{j,i})\}_{j,i=0}^{m-1,n-1}; z_C)$

$f_{0,1}, \ldots f_{m-1,n-1}$

z_A, z_C, z_V, z_R →

$\prod_{i=0}^{N} C_i^{\prod_{j=0}^{m-1} f_{j,i_j}} \cdot \prod_{k=0}^{m-1} (G_k \cdot Q_k)^{-x^k} = Comm(0, z_V, z_R)$

Generating a Balance Proof

Each Transaction published on the blockchain will contain

1. Transaction Output Coins (+ Range Proofs)
2. N_{old} Sigma Proofs (One proof per each Input Coin)

THIS INFORMATION IS ENOUGH TO GENERATE A BALANCE PROOF

Generating a Balance Proof

1. Transaction Output Coins $(C_{O1}, C_{O2}, \ldots, C_{ON_{new}})$

2. The transaction proof transcript contains the following information (taken from the N_old Sigma Proofs)

$(z_{V_1}, \cdots, z_{V_{N_old}})$ and $(z_{R_1}, \cdots, z_{R_{N_old}})$

where $z_{V_t} = V_t \cdot x^N - \sum_{k=0}^{n-1} \rho_k^t x^k$ and $z_{R_t} = R_t \cdot x^N - \sum_{k=0}^{n-1} \tau_k^t x^k$

$G_k = \prod_{i=0}^{N-1} C_i^{p_{i,k}}$

computing $p_{i,k}$ as is c

$Q_k = Comm(0, \rho_k, \tau_k)$

$\forall j \in [0, m-1], i \in [1, n-1]$

$f_{j,i} = \sigma_{l_j i} x + a_{j,i}$

$z_A = r_B \cdot x + r_A$

$z_C = r_C \cdot x + r_D$

$z_V = V \cdot x^m - \sum_{k=0}^{m-1} \rho_k \cdot x^k$

$z_R = R \cdot x^m - \sum_{k=0}^{m-1} \tau_k \cdot x^k$

$\{Comm(0, \rho_0^t, \tau_0^t), \cdots, Comm(0, \rho_{m-1}^t, \tau_{m-1}^t)\}$ for $t \in 1, \ldots, N_{old}$

Generating a Balance Proof

Each Verifier can compute the following values

$$A := (C_{O1} \cdot \ldots \cdot C_{ON_{new}})^{x^n} =$$

$$= g^{(S_{O1}+\ldots+S_{ON_{new}})x^n} h_1^{(V_{O1}+\ldots+V_{ON_{new}})x^n} h_2^{(R_{O1}+\ldots+R_{ON_{new}})x^n}$$

$$B := Comm(0; z_{V1} + \ldots + z_{VN_{old}}, z_{R1} + \ldots + z_{RN_{old}}) \cdot \prod_{t=1}^{N_{old}} \left(\prod_{k=0}^{m-1} Comm(0; \rho_k^t, \tau_k^t)^{x^k} \right)$$

$$= h_1^{(V_{I1}+\ldots+V_{IN_{old}})x^n} h_2^{(R_{I1}+\ldots+R_{IN_{old}})x^n}$$

If the balance is preserved then $\frac{A}{B} = g^X h_2^Y$

Prover provides a proof of representation of the value $\frac{A}{B}$ with respect to the generators g and h_2 (through gen. Schnorr proof of Knowledge?)

Generating Balance Proof

Any feedback or comments on balance proof generation method?

Batch Verification of 1-o-o-N Proofs

The verification of the ***t***-th 1-o-o-N Proof boils down to a big multi-exponentiation

$$\prod_{i=0}^{N} C_i^{t\ \prod_{j=0}^{m-1} f_{j,i_j}^t} \cdot \prod_{k=0}^{m-1} (G_k^t \cdot Q_k^t)^{-x^k} = Comm(0, z_V^t, z_R^t)$$

Batch Verification of 1-o-o-N Proofs

The verification of the ***t***-th 1-o-o-N Proof boils down to a big multi-exponentiation

$$\prod_{i=0}^{N} C_i^{t\ \prod_{j=0}^{m-1} f_{j,i_j}^t} \cdot \prod_{k=0}^{m-1} (G_k^t \cdot Q_k^t)^{-x^k} = Comm(0, z_V^t, z_R^t)$$

Which can be written as

$$\prod_{i=0}^{N} \left(C_i^t\right)^{f_i^t} \cdot D_t = E_t$$

where

$$f_i^t = \prod_{j=0}^{m-1} f_{j,i_j}^t \qquad D_t = \prod_{k=0}^{m-1} (G_k^t \cdot Q_k^t)^{-x^k} \qquad E_t = Comm(0, z_V^t, z_R^t)$$

Batch Verification of 1-o-o-N Proofs

The verification of the ***t***-th 1-o-o-N Proof boils down to a big multi-exponentiation

$$\prod_{i=0}^{N} C_i^{t\ \prod_{j=0}^{m-1} f_{j,i_j}^t} \cdot \prod_{k=0}^{m-1} (G_k^t \cdot Q_k^t)^{-x^k} = Comm(0, z_V^t, z_R^t)$$

Which can be written as

$$\prod_{i=0}^{N} \left(C_i^t\right)^{f_i^t} \cdot D_t = E_t$$

In the Zerocoin setup, the generators C_i^t are transaction specific: $C_i^t = C_i \cdot g^{(-s_t)}$

where

$$f_i^t = \prod_{j=0}^{m-1} f_{j,i_j}^t \qquad D_t = \prod_{k=0}^{m-1} (G_k^t \cdot Q_k^t)^{-x^k} \qquad E_t = Comm(0, z_V^t, z_R^t)$$

Batch Verification of 1-o-o-N Proofs

As $C_i^t = C_i \cdot g^{(-s_t)}$ we have

$$\prod_{i=0}^{N} \left(C_i^t\right)^{f_i^t} \cdot D_t = \prod_{i=0}^{N} \left(\frac{C_i}{g^{s_t}}\right)^{f_i^t} \cdot D_t = E_t$$

Now all N generators C_i are transaction agnostic

Each s_t is explicitly revealed during the SPEND,
The verifier can equivalently check the following equivalency.

$$\prod_{i=0}^{N} \left(C_i^t\right)^{f_i^t} \cdot D_t = E_t \quad \Longrightarrow \quad \prod_{i=0}^{N} C_i^{f_i^t} = \frac{E_t}{D_t} \cdot g^{s_t \cdot \left(\sum_{i=0}^{N} f_i^t\right)}$$

Batch Verification of 1-o-o-N Proofs

For verifying M different spend proofs in batch, the verifier

- Generates M random values $(y_1, \ldots . y_M)$
- Computes

$$\prod_{t=1}^{M}\left(\prod_{i=0}^{N} C_i^{f_i^t}\right)^{y_t} = \prod_{t=1}^{M}\left(\frac{E_t}{D_t} \cdot g^{s_t \cdot (\sum_{i=0}^{N} f_i^t)}\right)^{y_t}$$

Or alternatively

$$\prod_{i=0}^{N} C_i^{\sum_{t=1}^{M} y_t \cdot f_i^t} = g^{\sum_{t=1}^{M}\left(y_t \cdot s_t \cdot \sum_{i=0}^{N} f_i^t\right)} \cdot \prod_{t=1}^{M}\left(\frac{E_t}{D_t}\right)^{y_t}$$

We can save N exponentiation for each extra proof.

Batch Verification Performance

Batch Size	Verification Time	Average cost per verification
5	623	124.6
10	636	63.6
50	1125	22.5
100	1759	17.6
500	6978	14
1000	13719	13.7

Table 2. Batch Verification Timing for the Anonymity Set of 16384

Batch Size	Verification Time	Average cost per verification
5	1090	218
10	1186	118.6
50	1970	39.4
100	2967	29.7
500	11098	22.2
1000	21825	21.8

Table 3. Batch Verification Timing for the Anonymity Set of 32384

Batch Size	Verification Time	Average cost per verification
5	2162	432.5
10	2317	232
50	3691	73.8
100	5342	53.4
500	19660	39.3
1000	38192	38.2

Table 4. Batch Verification Timing for the Anonymity Set of 65536

Batch Size	Verification Time	Average cost per verification
5	9310	1862
10	10024	1000
50	16737	335
100	24995	250

Table 5. Batch Verification Timing for the Anonymity Set of 262144

Batch Verification of 1-o-o-N Proofs

Any feedback or comments on how we can improve these batching methods?

☺

Enabling Direct Anonymous Payments

In order to spend a coin $C = g^S h_1^V h_2^R$, the user should possess all secret values S, V, R.

1. The receiver can generate new public key $g_1 = g^x$

2. The Sender outputs the new coin as $C = (g^x)^S h_1^V h_2^R = g_1^S h_1^V h_2^R$ along with the public key g_1

Only the recipient possessing the secret value x can spend the coin

$$C = (g^x)^S h_1^V h_2^R = g^{x.S} h_1^V h_2^R$$

as only he knows the coin's real serial number $x \cdot S$.

Enabling Direct Anonymous Payments

In the balance proof generation phase, we will end up having more complex representation of A/B.

$$\frac{A}{B} = g_1^{S_{O1}x^n} \cdot \ldots \cdot g_{N_{new}}^{S_{O_{N_{new}}}x^n} h_2^Y \quad \text{(Instead of } \frac{A}{B} = g^X h_2^Y \text{)}$$

For completing the balance proof, the Prover should provide a proof of representation of the value $\frac{A}{B}$ with respect to the generators $(g_1, g_2, g_{N_{new}}, h_2)$ (what is the best way to do this?)

Enabling Direct Anonymous Payments

Any comment of feedback on this?

How we could Improve 1-out-of-N Proofs

- Design of Efficient M-out-of-N Proofs.
- Scaling 1-out-of-N Proofs (?)

Maybe we could work with (you) Markulf on these ideas?

THANK YOU

INTRODUCTION

	No trusted setup	confidentiality	anonymity	performance
Confidential Transactions (CT)	YES	YES	NO	FAST
Bulletproofs	YES	YES	NO	FAST
Zerocoin	NO	NO	YES	SLOW
Zerocash	NO	YES	YES	FASTEST
Sigma	YES	NO	YES	FAST
Lelantus	**YES**	**YES**	**YES**	**FAST**

CRYPTOGRAPHIC BACKGROUND

One-out-of-Many (Σ) Proofs for a Commitment Opening to 0

Jens Groth [2] provided a Σ-protocol for knowledge of one out of N commitments $c_0, \ldots, c_N$ being a commitment to 0, or more precisely a Σ-protocol for the relation

$$R = \{(ck, (c_0; \ldots; c_N); (l, r) \quad | \quad \forall i : c_i \in C_{ck} \wedge l \in \{0, \ldots, N-1\} \wedge r \in Z_p \wedge c_l = Com_{ck}(0, r))\}$$

$$R = \{(ck, (c_0; \ldots; c_N), (l, r) \quad | \quad \forall i : c_i \in C_{ck} \wedge l \in \{0, \ldots, N-1\} \wedge v, r \in Z_q \wedge c_l = Comm_{ck}(0, v, r))\}$$

CRYPTOGRAPHIC BACKGROUND

Bulletproofs

$$R = \{g, h \in G, V, n; \quad v, \gamma \in Z_p \quad | \quad V = g^v h^\gamma \wedge v \in [0, 2^n - 1]\}$$

Bulletproofs can work with V being a double-blinded commitment to the value v using two random values γ1 and γ2

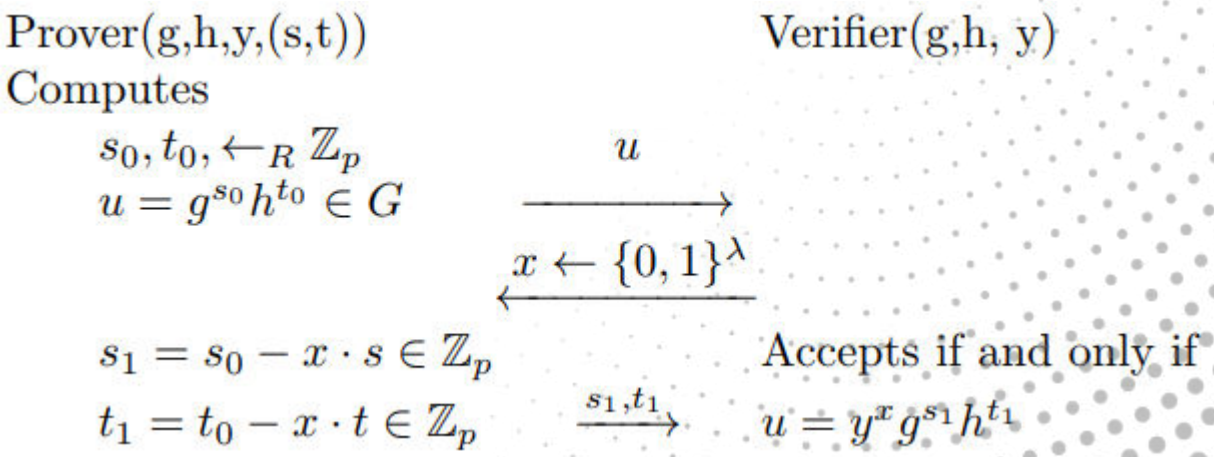
$$R = \{g, h \in G, V, n; \quad v, \gamma_1, \gamma_2 \in Z_p \quad | \quad V = g^v h_1^{\gamma_1} h_2^{\gamma_2} \wedge v \in [0, 2^n - 1]\}$$

CRYPTOGRAPHIC BACKGROUND

Generalized Schnorr proofs

$$R = \{g, h \in G, y \quad ; \quad s, t \in Z_p \quad | \quad y = g^s h^t \quad \}$$

The protocol is depicted in the diagram below.



IDEA EVALUATION

Let us assume the transaction spends N_{old} inputs denoted as $\{c_{i1} = g^{s_1} h^{v_{i1}}, \cdots, c_{iN_{old}} = g^{s_{N_{old}}} h^{v_{iN_{old}}}\}$ and a transparent net value V_{IN} to output N_{new} outputs $\{c_{o1} = (g^{s_{o1}} h^{v_{o1}}, \cdots, c_{oN_{new}} = g^{s_{oN_{new}}} h^{v_{oN_{new}}}\}$ at the transaction fee f. If the transaction balance is preserved, the following equation holds

$$v_{i1} + \cdots v_{iN_{old}} = v_{o1} + \cdots + v_{oN_{new}} + f$$

This is equivalent to having

$$c_{i1} \cdot \ldots \cdot c_{iN_{old}} / (c_{o1} \cdot \ldots \cdot c_{oN_{new}} \cdot h^f) = g^S$$

where g^S is a valid public key corresponding to the private key

$$S = (s_{i1} + \cdots + s_{iN_{old}}) - (s_{o1} + \cdots + s_{oN_{new}})$$

IDEA EVALUATION

$C = g^s h^v$ where v is the coin's hidden value and s is its unique serial number.

$$z_d = vx^n - \sum_{k=0}^{n-1} \rho_k x^k$$

IDEA EVALUATION

After relevant modifications in the original Σ-protocol we can also explicitly reveal the commitments of these blinding factors ρk as Com(0, ρk)

$$(z^1, z^2, \ldots, z^{N_{old}}) \text{ where } z^t = V_t x^n - \sum_{k=0}^{n-1} \rho_k x^k \text{ for } t \in [1, \ldots, N_{old}]$$

$$Com(0, \rho_k^t) \text{ for } k \in [0, \ldots, n-1] \text{ and } t \in [1, \ldots, N_{old}]$$

IDEA EVALUATION

Now one can observe that each verifier can perform the following computations

$$A = (C_{O1} \cdot \ldots \cdot C_{ON_{new}})^{x^n} = g^{(s_{O1}+\cdots+s_{ON_{new}})x^n} h^{(v_{O1}+\cdots+v_{ON_{new}})x^n}$$

x is the challenge parameter generated for the non-interactive one out of many (Σ) protocols.

IDEA EVALUATION

The verifier can compute

$$B = Com(0, z^1 + \ldots + z^{N_{old}}) \cdot \prod_{t=1}^{N_{old}} (\prod_{k=1}^{n} Com(0, \rho_k^t)^{x^k})$$

$$= h^{(v_{i1} + \cdots + v_{iN_{old}})x^n}$$

If the balance equation holds, then the value A/B will be a valid public key of the form

$$\frac{(C_{o1} \cdot \ldots C_{oN_{new}})^{x^n}}{Com(0, z^1 + \ldots + z^{N_{old}}) \cdot \prod_{t=1}^{N_{old}} (\prod_{k=1}^{n} Com(0, \rho_k^t)^{x^k})} = g^{(s_{o1} + \cdots + s_{oN_{new}})x^n}$$

IDEA EVALUATION

Now it becomes evident that, along with the Σ-proofs, the prover has to additionally prove the knowledge of the exponent value

$$S = (s_{o1} + \cdots + s_{oN_{new}})x^n$$

IDEA EVALUATION

Insecured!

CONFIDENTIAL PAYMENT HIDING TRANSACTION VALUES AND ORIGINS

$$C = Comm(S; V, R) = g^S h_1^V h_2^R$$

CONFIDENTIAL PAYMENT HIDING TRANSACTION VALUES AND ORIGINS

Each transaction will be comprised of corresponding **spend descriptions**, **output descriptions** and the **transaction balance proof.**

1. For each input coin
   - Prover proves that he knows an index $l \in [0, ..N]$ and the values S, V, R of the coin C_l , so that $C_l = g^S h_1^V h_2^R$ with the help of Σ-protocol for a double-blinded commitment opening to 0, which in detail description is provided in section 4.1. The process steps are the following
     (a) Prover reveals the serial number S.
     (b) Prover parses the initial set of all commitments $C = (C_0, C_1, .C_{N-1})$ and computes $C_i := C_i \cdot Comm(S, 0, 0)^{-1}$
     (c) Prover provides a non-interactive Σ-proof for a double-blinded commitment opening to 0 for the new set $C_i := C_i \cdot Comm(S, 0, 0)^{-1}$.

CONFIDENTIAL PAYMENT HIDING TRANSACTION VALUES AND ORIGINS

2. For each output coin
   - Prover provides a zero-knowledge range proof, showing that the coin does not hide a negative value. This is done with the help of Bulletproofs for double-blinded commitments described in the section 4.3.
3. Prover provides a zero-knowledge proof that

$$V_{IN} + V_{I1} + \ldots + V_{IN_{old}} = V_{OUT} + V_{O1} + \ldots + V_{ON_{new}} + f$$

CONFIDENTIAL PAYMENT HIDING TRANSACTION VALUES AND ORIGINS

Σ-Protocol for One out of N Double-Blinded Commitments Opening to 0

$$R = \{(ck, (c_0; \ldots; c_N), (l, r) \quad | \quad \forall i : c_i \in C_{ck} \wedge l \in \{0, \ldots, N-1\} \wedge v, r \in Z_q \wedge c_l = Comm_{ck}(0, v, r))\}$$

$P(gk, crs, (C_0, \ldots, C_{N-1}), l, V, R)$

Compute

$r_A, r_B, r_C, r_D, a_{j,1}, \ldots, a_{j,n-1} \leftarrow_R \mathbb{Z}_q$

for $j \in [0, \cdots, m-1]$

$a_{j,0} = -\sum_{i=1}^{n-1} a_{j,i}$

$B := Com_{ck}(\sigma_{l_0,0}, \ldots, \sigma_{l_{m-1},n-1}; r_B)$

$A := Com_{ck}(a_{0,0}, \ldots, a_{m-1,n-1}; r_A)$

$C := Com_{ck}(\{a_{j,i}(1 - 2\sigma_{l_j,i})\}_{j,i=0}^{m-1,n-1}; r_C)$

$D := Com_{ck}(-a_{0,0}^2, \ldots, -a_{m-1,n-1}^2; r_D)$

For $k \in 0, ..., m-1$

$\rho_k, \tau_k \leftarrow_R \mathbb{Z}_q$

$G_k = \prod_{i=0}^{N-1} C_i^{p_{i,k}}$

computing $p_{i,k}$ as is described above

$Q_k = Comm(0, \rho_k, \tau_k)$

$\forall j \in [0, m-1], i \in [1, n-1]$

$f_{j,i} = \sigma_{l_j i} x + a_{j,i}$

$z_A = r_B \cdot x + r_A$

$z_C = r_C \cdot x + r_D$

$z_V = V \cdot x^m - \sum_{k=0}^{m-1} \rho_k \cdot x^k$

$z_R = R \cdot x^m - \sum_{k=0}^{m-1} \tau_k \cdot x^k$

$A, B, C, D, \{G_K, Q_K\}_{k=0}^{m-1}$ →

$\xleftarrow{x \leftarrow \{0,1\}^\lambda}$

$f_{0,1}, \ldots f_{m-1,n-1}$

z_A, z_C, z_V, z_R →

$V(gk, crs, (C_0, \ldots, C_{N-1}))$

Accept if and only if

The values

$A, B, C, D, G_0, Q_0 \ldots, G_{m-1}, Q_{m-1} \in G$

$\{f_{j,i}\}_{j,i=0,1}^{m-1,n-1}, z_A, z_C, z_V, z_R \in \mathbb{Z}_q$

$\forall j : f_{j,0} = x - \sum_{i=1}^{n-1} f_{j,i}$

$B^x A = Com(f_{0,0}, \ldots, \ldots f_{m-1,n-1}; z_A)$

$C^x D = Com(\{f_{j,i}(x - f_{j,i})\}_{j,i=0}^{m-1,n-1}; z_C)$

$\prod_{i=0}^{N} C_i^{\prod_{j=0}^{m-1} f_{j,i_j}} \cdot \prod_{k=0}^{m-1} (G_k \cdot Q_k)^{-x^k} = Comm(0, z_V, z_R)$

CONFIDENTIAL PAYMENT HIDING TRANSACTION VALUES AND ORIGINS

Balance Proof for Transactions with Multiple Spend and Output Transfers

$$(z_{V_1}, \cdots, z_{V_{N_{old}}}) \text{ and } (z_{R_1}, \cdots, z_{R_{N_{old}}}) \text{ where } z_{V_t} = V_t \cdot x^N - \sum_{k=0}^{n-1} \rho_k^t x^k \text{ and } z_{R_t} = R_t \cdot x^N - \sum_{k=0}^{n-1} \tau_k^t x^k$$

$$\{Comm(0, \rho_0^t, \tau_0^t), \cdots, Comm(0, \rho_{m-1}^t, \tau_{m-1}^t)\} \qquad \text{for } t \in 1, .., n$$

1. Takes all output coins, the net output value V_{OUT}, the transaction fee f, and the *Sigma*-proof challenge value x and computes the following element

$$A := (C_{O1} \cdot \ldots \cdot C_{ON_{new}})^{x^n} \cdot h_1^{(V_{OUT}+f)x^n} =$$
$$= g^{(S_{O1}+\ldots+S_{ON_{new}})x^n} h_1^{(V_{OUT}+V_{O1}+\ldots+V_{ON_{new}}+f)x^n} h_2^{(R_{O1}+\ldots+R_{ON_{new}})x^n}$$

2. Second, taking the transaction net input value V_{IN}, the elements $z_{V1}, \ldots, z_{VN_{old}}, z_{R1}, \ldots, z_{RN_{old}}$ and $\{Comm(0, \rho_k^t, \tau_k^t)\}_{k=0}^{m-1}$ from the corresponding Σ-proof transcripts, the verifier computes the element

$$B := h_1^{V_{IN}x^n} \cdot Comm(0; z_{V1} + \ldots + z_{VN_{old}}, z_{R1} + \ldots + z_{RN_{old}}) \cdot \prod_{t=1}^{N_{old}} (\prod_{k=0}^{m-1} Comm(0; \rho_k^t, \tau_k^t)^{x^k})$$
$$= h_1^{(V_{IN}+V_{I1}+\ldots+V_{IN_{old}})x^n} h_2^{(R_{I1}+\ldots+R_{IN_{old}})x^n}$$

3. It becomes evident that, if the balance transaction holds, the h_1 exponents in A and B will cancel each other out and we will have

$$\frac{A}{B} = g^X h_2^Y$$

where

$$X = (S_{O1} + \ldots + S_{ON_{new}})x^n \text{ and } Y = ((R_{O1} + \ldots + R_{ON_{new}}) - (R_{I1} + \ldots + R_{IN_{old}}))x^n$$

Thank you