One-Time, Zero-Sum Ring Signature

Conner Fromknecht

November 12, 2015

1 Introduction

Is Bitcoin a currency? The jurisdictions of institutions all over the world have culminated in everything but a unanimous decision. In practice, Bitcoin offers the majority of features present in typical fiat currencies such as payments of arbitrary amount and exchanges for another currency. However, Bitcoin fails to meet one of the key properties of a true currency—fungibility.

Not all Bitcoins are created equal. Since the transaction history is public, so too are the balances and payments made by its participants. After any transaction is completed, the receiver is able to view the complete history of the coin. If the coin's history includes transactions that are known to be a part of a scandal, the receiver can choose to reject the payment. This inherently makes some Bitcoin less valuable than others, since accepting tainted coins assumes the risk that the next receiver may not accept them. By definition, this inequality between coins prevents Bitcoin from being fungible.

The goal of this work is to design a more fungible cryptocurrency. In order to do so, we tackle two distinct problems that allow a third party to discriminate against the coins used in a standard Bitcoin transaction. The first is linkability, which allows an individual to trace the history of a transaction and determine the accounts that have held the coin before. This work builds upon much of CryptoNote's [4] anonymous transaction scheme to ensure the destination is obfuscated, preventing third parties from explicitly backwards constructing a path to the sender.

However, this alone is not enough. The transaction amounts are still visible, which allows any two transactions of equal amounts to be possibly linked, and thus jeopardizing unlinkability. Therefore, our solution must also hide transaction amounts. This work begins with a similar approach to the one outlined by Gregory Maxwell's Confidential Transaction scheme [2]. This allows the sender to prove that the value is within a certain range, say $[0, 2^l)$. In order to publicly verify this, the signer must coordinate the blinding factors and amounts of different inputs such that they result in a zero sum. However, when using CryptoNote's ring signatures, the signer has no control over the blinding factors of other inputs. Thus, we propose a new ring signature construction, called a One-Time, Zero-Sum Ring Signature (OZRS) that proves the output amount is equal to exactly one of the committed input values. It also proves that a commitment receives a new blinding factor after each transaction and allows only the recipient to learn the amount enclosed. Furthermore, the size of an OZRS is the same as the One-Time Ring Signatures used in the standard CryptoNote protocol.

By combining the unlinkability and confidentiality properties of this work, the receiver of a transaction is able to verify its amount but also learns nothing about the coin's history. Since this is true for any future recipient, the receiver can accept the payment as is without concern that it may be rejected by another party.

2 Construction

Building on CryptoNote's architecture, this work requires relatively simple changes to the high level protocol. The modifications include adding a single field to a transaction output and replacing the One-Time Ring Signature with an OZRS. Furthermore, every transaction amount is committed using a Pedersen Commitment and accompanied by a range proof using the Borromean scheme described in [3]. For simplicity, we assume that the range proof is done in binary, but the results can be extended to any publicly known encoding.

2.1 Transactions

Here we describe how to construct a single output transaction, where some user is trying to send the value v' to the standard address $(A = aG, B = bG)$. These steps operate in addition to the unmodified CryptoNote protocol, unless otherwise specified here.

When building a single output transaction, the signer also chooses a random number $q \in Z_N$ and adds the *blind seed* $Q = qG$ to the transaction output. The signer then computes $y' = \mathcal{H}_S(qB)$ which is called the *output blinding factor*, where B is taken to be the receiver's public key. Using y', the output commitment $C' = y'G + v'H$ is constructed by first deterministically generating blinding factors $\gamma^{(i)}$ to commit each of the l bits β_i in v'. These blinding factors are computed by

$$\gamma^{(1)} = \mathcal{H}_S(y')$$
$$\gamma^{(i+1)} = \begin{cases} y' - \sum_{j=1}^{i} \gamma^{(j)} & : i = l-1 \\ \mathcal{H}_S(\gamma^{(i)}) & : otherwise. \end{cases}$$

The signer then outputs the final $C' = \sum_{i=1}^{l} c_i$, where $c_i = \gamma^{(i)}G + \beta_i H$ and β_i is either 2^i or 0 depending on the i^{th} bit of v'. Each c_i and $\gamma^{(i)}$ is further used to construct the range proof of C' using the techniques in [3].

A receiver uses the blind seed to compute $y' = \mathcal{H}_S(bQ)$ where b is taken to be the receiver's super secret private key. A receiver with knowledge of y' is also able to recover each of the $\gamma^{(i)}$ blinding factors. Then, he can recover the bits of v' by checking $c_i - \gamma^{(i)}G = 0$ or $d_i - \gamma^{(i)}G = 0$, where the challenge public key $d_i = c_i - 2^i H$. If the first is true, then $\beta_i = 0$; if the second check passes then $\beta_i = 2^i$. Note that the signer of a transaction could pick $Q' \neq qG$, which results in different $\gamma^{(i)}$ with high probability. The receiver can easily detect this event if both c_i and d_i failed the above test. In this case, the receiver should not accept the transaction as payment, since he will be unable to spend it.

2.1.1 Multiple Outputs

Supporting transactions with m outputs only requires a small modification to the single output case. We compute one $C'_i = y'_i G + v'_i H$ for each output amount as described above, providing a range proof for each. We then compute $C' = \sum_{i=1}^{m} C'_i$ and $y' = \sum_{i=1}^{m} y'_i$. Note that each output now has its own blind seed, so they can each be recovered independently. Lastly, instead of creating a single transaction public key $R = rG$, we create one for each output. This allows the outputs of a transaction to be spent independently of each other, much like how they are in Bitcoin.

2.1.2 Transaction Structure

This section describes the format of a transaction incorporating the above changes. The construction of the OZRS is described in the subsequent section.

INPUT	
Key Image:	$I = x_* \mathcal{H}_P(x_*)$
Input Transaction Hashes:	$\{\mathcal{H}_S(T_i)\}_n$
OUTPUT	
Transaction Public Keys:	$\{R_i = r_i G\}_m$
Destination Keys:	$\{P_i = \mathcal{H}_S(r_i A_i)G + B_i\}_m$
Blind Seeds:	$\{Q_i = q_i G\}_m$
Commitments:	$\{C'_i = y'_i G + v'_i H\}_m$
Range Proofs:	$\{\pi_l(C'_i)\}_m$
SIGNATURE	
OZRS:	$\Pi = (e, r_1, \ldots, r_n, s_1, \ldots, s_n)$

2.2 One-Time, Zero-Sum Ring Signature

For any transaction with n inputs and m outputs, let $X = \{X_i = x_i G\}_{i \in [1,n]}$ be the set of input destination keys and $* \in [1, n]$ to be the index of signer's public key X_*. Furthermore, let $C = \{C_i = y_i G + v_i H\}_{i \in [1,n]}$ be the set input commitments, where each C_i commits each X_i to the value v_i. We call each y_i an input blinding factor. After constructing a transaction, the signer also holds the new output blinding factor y' and total output value v' in

$$C' = y'G + v'H = \sum_{i=1}^{m} C'_i = \sum_{i=1}^{m} y'_iG + v'_iH,$$

where each C'_i represents an individual output commitment. Here we present a ring signature formulation constructed as an AOS ring signature [1] that uses a three-way chameleon hash to prove the following properties:

1. The signer knows at least one secret key x_i for a public key X_i
2. The signer knows the secret key x_* corresponding to the preimage $I = x_*\mathcal{H}_P(X_*)$ of X_*.
3. The sum of the output commitments C' holds a value equal to the sender's input C_*.

More formally, the ring signature is a Non-Interactive, Zero-Knowledge Proof of Knowledge on a message M such that all values other than $\{x_i\}$, $\{y_i\}$, $\{v_i\}$, y', v', $\{y'_i\}$, $\{v'_i\}$ and are known to the prover, defined

$$\begin{aligned}\textsc{NiZKPoK}[M](\{x_i\},\{y_i\},\{v_i\},y',v',\{y'_i\},\{v'_i\}) : \{\exists i : \; & C' - C_i = (y' - y_i)G \\ & \wedge X_i = x_iG \\ & \wedge I = x_i\mathcal{H}_P(X_i)\}.\end{aligned}$$

The One-Time, Zero-Sum Ring Signature scheme consists of the four operations (Gen, Sign, Verify, Link).

- $\textsc{Gen}(N, G) \to (a, A)$

  Choose $a \leftarrow Z_N$ at random.
  Compute $A = aG$ and output (a, A).

- $\textsc{Sign}(M, C, C', y', y_*, x_*, X) \to \Pi$

  Compute the signing key's preimage and commitment differences

$$I = x_*\mathcal{H}_P(X_*) \tag{1}$$
$$D_i = C' - C_i = (y' - y_i)G. \tag{2}$$

  Next, we build the non-interactive challenge e. Choose $k_1, k_2 \leftarrow Z_N^2$ at random. Starting at index $*$, compute

$$e_*^{(1)} = \mathcal{H}_S(M \mid\mid k_1G \mid\mid k_2G \mid\mid k_2\mathcal{H}_P(X_*))$$

Continue computing successive $e_i^{(\cdot)}$, wrapping around after $i = n$, until $e_{*-1}^{(2)}$ using the following steps

$$r_i, s_i \leftarrow Z_N^2 \tag{3}$$
$$e_i^{(2)} = \mathcal{H}_S(e_i^{(1)}) \tag{4}$$
$$e_{i+1}^{(1)} = \mathcal{H}_S(M \mid\mid r_i G - e_i^{(1)} D_i \mid\mid s_i G - e_i^{(2)} X_i \mid\mid s_i \mathcal{H}_P(X_i) - e_i^{(2)} I) \tag{5}$$

Lastly, we set r_{*-1}, s_{*-1} so that the hash hits $e_*^{(1)}$ by

$$r_{*-1} = k_1 + e_{*-1}^{(1)}(y' - y_*)$$
$$s_{*-1} = k_2 + e_{*-1}^{(2)} x_*$$

Assign $e = e_1^{(1)}$ and output the final proof $\Pi = (e, r_1, \ldots, r_n, s_1, \ldots, s_n)$.

- Verify$(\Pi, M, C, C', X, I) \rightarrow \{0, 1\}$

  First compute each commit difference D_i using equation (2). Starting with $e = e_1^{(1)}$, compute the forward $e_i^{(\cdot)}$ for $i \in [1, n]$ using relations (4) and (5). The verifier then checks that

  $$e = \mathcal{H}_S(M \mid\mid r_n G - e_n^{(1)} D_n \mid\mid s_n G - e_n^{(2)} X_n \mid\mid s_n \mathcal{H}_P(X_n) - e_n^{(2)} I)$$

  and Link(I) fails. If both of these are met, return 1. Otherwise, return 0.

- Link$(I) \rightarrow \{0, 1\}$

  Let $\mathcal{I}$ be the set of all spent preimages. Return 1 if $I \in \mathcal{I}$, otherwise return 0.

References

[1] M. Abe, M. Ohkubo, and K. Suzuki. 1-out-of-n signatures from a variety of keys. In Advances in Cryptology - ASIACRYPT, 2002.

[2] G. Maxwell. Confidential transactions. `https://people.xiph.org/~greg/confidential_values.txt`, 2015.

[3] G. Maxwell and A. Poelstra. Borromean ring signatures. `http://diyhpl.us/~bryan/papers2/bitcoin/Borromean%20ring%20signatures.pdf`, 2015.

[4] N. van Saberhagen. Cryptonote. `https://cryptonote.org/whitepaper.pdf`, 2013.