A Survey of Three Atomic Swaps with Monero

F. Barbàra[1]

me@fadibarbara.it

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

insert intro

1.1 Transactions

Before formally introduce atomic swaps, we give a definition of *transaction*. By doing that we are able to make a precise comparison between different atomic swaps methods. Our definition of transaction is based on the notation introduced by Avarikioti *et al.* [**?**] and expanded by Nadahalli *et al.* [**?**]. Their method is structured for atomic swaps between blockchains that use a UTXO model. We expand it further to take into account the cases in which atomic swaps are performed between blockchains where at least one of them uses an Account model.

Our definition of Transaction will use the concept of *transaction predicate*, or "predicate" for short.

Definition 1 (Predicate) *A predicate is one entry among the following list items, of a* $\wedge/\vee$ (AND/OR) *logical combination of two or more predicates. The following predicates are accepted:*

- σ_a*: a signature verifiable with public key* A
- $H_s := h(s)$*: a hash digest obtained from preimage* s *using hash function* h
- Δ_k*: a timelock of* k *blocks needed to elapse for the transaction to be spendable*
- `0xabc...ef`*: an address*

We can now give the definition of transaction

Definition 2 (Transaction) *A transaction* T *is couple* $< model|mapping >$ *such that*

$$T =< \mathfrak{m}, [o_j, o_k, \dots] \mapsto [o_i^1, o_i^1, \dots] >$$

where:

- $\mathfrak{m} \in \{\mathfrak{a}, \mathfrak{u}\}$, *with* $\mathfrak{a}$ *(respectively* $\mathfrak{u}$*) representing a transaction done on a blockchain following the account (resp. UTXO) model*
- $o. =< x.|P. >$ *where* $x.$ *is the value (amount) of the transaction and* $P.$ *is a predicate as defined in Definition 1*

We now show that Definition 2 is able to represent transaction both in UTXO and in Account model blockchains.

The following transaction T_j:

$$T_j =< \mathfrak{u}, [\underbrace{(2|\sigma_a)}_{T_l}, (3|\sigma_b, \Delta_{10})] \mapsto [(1|\sigma_c), (4|\sigma_d)] >$$

is a transaction in the UTXO model whose first input $(2|\sigma_a)$ comes from transaction T_l, has an amount of 2 coins and can be redeemed only using public key A; the second input $(3|\sigma_b, \Delta_{10})$ has an amount of 3 coins and can be redeemed only using public key B after at least 10 from when it has been created, for a total of 5 coins. We may omit the transaction from where an input comes from if it is not of interest. Moreover, the first output $o_u^1 = (1|\sigma_c)$ can be redeemed with public key C and similar reasoning goes for $o_u^2 = (4|\sigma_d)$, for a total of 5 coins redeemed.

On the other hand the following transaction T_k:

$$T_k =< \mathfrak{a}, [(2|\texttt{0xba6d...de})] \mapsto [(2|\texttt{0x32be...9a})] >$$

is a transaction in the account model which spends 2 coins from address `0xba6d...de` to address `0x32be...9a`.

Transaction validity is intuitively obvious, but we give a formal definition to account to some differences between the two models.

Definition 3 (Valid Transaction) *A transaction is valid when ....*

is it necessary??

Note also that a transaction in the account model can be also a combination of multiple transactions.

1.2 Atomic Swaps

2 Building blocks

Three building blocks make atomic swaps from (respectively to Monero to (resp. from) other blockchain possible: a Discrete-Logarithm Equality Proof (DLEP), the use of Adaptor Signatures, and the use of view keys in the Monero blockchain. In the following we analyze these building blocks.

2.1 Discrete-Logarithm Equality Proof

All the presented atomic swaps methods leverage the use of the same private keys on both the Monero and the Bitcoin or Ethereum blockchain. On the one

hand this is an efficient way to deal with the lack of scripts capabilities of Monero and it paves the way to the use of adaptor signatures (Section 2.2). On the other hand since Monero and Bitcoin (or Monero and Ethereum) do *not* share the same curve proving equality of private keys on two curves requires some non trivial work.

All three atomic swap methods reference the technical note written by Noether [1] [2] as their base for the DLEP construction.

While the technical note presents all the computational information needed to perform a successful DLEP, the note does not provide any theoretical information. A formalization of the content of the note ca be found in Appendix I of this work. In this section we just highlight the foundational steps.

The Noether DLEP in [2] is a non-interactive zero-knowledge proof of knowledge between two actors: Peggy (acting as the prover) and Victor (acting as the verifier). We assume that Peggy and Victor share the parameters of the two groups $\mathbb{G}$ and $\mathbb{H}$, with points $G, G' \in \mathbb{G}$ and $H, H' \in \mathbb{H}$ generators of the respective froups. We also assume $|G| = p$ and $|H| = q$, with p and q prime numbers and $p < q$. The goal of Peggy is to provide two points $xG' \in \mathbb{G}$ and $xH' \in \mathbb{H}$, and a proof that xG' and xH' share the same descrete logarithm $x < p$[2]. The proof is built in a three steps process.

In the first step peggy splits x into a bit sequence, and creates as manny "blinders" as the number of bits of x, for each group. Formally:

1. $x = \sum_{i=0}^{n-1} 2^i b_i$
2. $r_0, \ldots, r_{n-2} \xleftarrow{\$} \mathbb{Z}_p, \quad s_0, \ldots, s_{n-2} \xleftarrow{\$} \mathbb{Z}_q$
3. $r_{n-1} = -(2^{n-1})^{-1} \sum_{i=0}^{n-2} 2^i r_i, \quad s_{n-1} = -(2^{n-1})^{-1} \sum_{i=0}^{n-2} 2^i s_i$

In the second step, Peggy creates Pedersen Commitments for each (b_i, r_i) and (b_i, s_i), for a total of $2n$ commitments:

$$C_i^{\mathbb{G}} = b_i G' + r_i G, \quad C_i^{\mathbb{H}} = b_i H' + r_i H, \quad \forall i = 0, \ldots, n-1$$

In the third step, Peggy creates $2n$ different Schnorr ring-signatures of the commitments: two signatures for each message $(C_i^{\mathbb{G}}, C_i^{\mathbb{H}})$, the first signed with r_i and the second with s_i. The signing procedure derives from the work of Abe, Okhubo and Suzuki [1], and since a complete treatment and a comparison between the work done in [2] and [1] is provided in Appendix I, we omit the details in this section. We just say that the proof π is so composed:

insert reference

$$\pi = (xG', xH', \{\sigma_i^{\mathbb{G}}\}_{i=0}^{n-1}, \{\sigma_i^{\mathbb{H}}\}_{i=0}^{n-1}) \tag{1}$$

where $\{\sigma_i^{\mathbb{G}}\}_{i=0}^{n-1}$ are the n signatures of the commmitments using r_i.

[1] As written on the note the credit for the first presentation of the DLEP goes to Andrew Poelstra

[2] The definition of *same* requires some more formal definition, which can be found in Appendix **Iinsert reference**.

2.2 Adaptor Signatures

2.3 View Keys

References

1. Abe, M., Ohkubo, M., Suzuki, K.: 1-out-of-n signatures from a variety of keys. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **87-A**(1), 131–140 (2004), https://link.springer.com/content/pdf/10.1007/3-540-36178-2_26.pdf
2. Noether, S.: Mrl-0010: Dirscrete logarithm equality across groups. Tech. rep., Monero Research Lab (2018)

Appendix I

The goal of this post is the expansion of the MRL-0010 technical note by Sarang Noether in the Monero Research Lab [2] (the "report" from now on) into a more complete work. We will see why pieces of the report fit together.

The goal is to prove that two point $A \in \mathbb{G}$ and $B \in \mathbb{H}$ share the same (up to equivalence) discrete logarithm, i.e. if $A = aG$ where $G \in \mathbb{G}$ is the generator of $\mathbb{G}$ and if $B = bH$ where $H \in \mathbb{H}$ is the generator of $\mathbb{H}$ then $a = b$ (up to equivalence).

give better definition AND insert use cases

We proced in the following manner. Since this is actually a zero-knowledge proof of knowledge, we first explain how to *prove* the equality and then how to *verify* the proof. finally we see how to derive proofs of correctness and security.

As in Section 2.1, we assume Peggy (the prover) wants to prove to Victor (the verifier) the equality. We assume Peggy and Victor to have knowledge of the parameters of both curves. In particular they know of the generators $G, G' \in \mathbb{G}$ and $H, H' \in \mathbb{H}$ and of two statistically independent hash functions:

are they?

$$H_{\mathbb{G}} : \{0,1\}^* \to \mathbb{Z}_p, \qquad p = |G|, |G| := \min\{a \in \mathbb{N} | aG = 0\} \tag{2}$$

$$H_{\mathbb{H}} : \{0,1\}^* \to \mathbb{Z}_q, \qquad q = |H|, \tag{3}$$

Using the aforementioned notation, our goal is to give Peggy a way to contruct a proof for Victor showing the discrete logarithm equality of two points $xG' \in \mathbb{G}$ and $xH' \in \mathbb{H}$, withour revealing x. Also, without loss of generality, assume $p < q$.

2.4 Proof Construction

The proof construction is a three steps process. The bulk of the proof is built in the third steps, which as mentioned can be seen as deriving from the work on ring-signatures in Abe *et al.* [1]. The rest of this section is so composed. We first explain Steps 1 and 2, then we explain how a ring signature is composed under [1] and finally we use it to explain how to complete the proof π in the last part of this section.

Step 1 The first intuition is to see the discrete logarithm x as the sequence of bits in its bit representation. In other words, $x = b_{n-1}, \ldots, b_0$, or

$$x = \sum_{i=0}^{n-1} b_i 2^i \tag{4}$$

Note that since $b_i \in \{0, 1\}$ then b_i can be easily mapped and used as element of either $\mathbb{Z}_p$ or $\mathbb{Z}_q$.

Step 2 Goal of this step is to create two sets of Pedersen Commitments, one set for xG' and one for xH'. In each set, each commitment is a commitment for one bit. Next paragraph explain the procedure in details.

Step 2.1 Create two sets of $n-1$ blinders

$$r_{n-2}, \ldots, r_0 \xleftarrow{\$} \mathbb{Z}_p, \quad s_{n-2}, \ldots, s_0 \xleftarrow{\$} \mathbb{Z}_q \tag{5}$$

Step 2.2 Create the n-th blinder as linear combianation of the previous ones (the reason will be clear in the next paragraph):

$$\begin{aligned} r_{n-1} &= -(2^{n-1})^{-1} \sum_{i=0}^{n-2} r_i 2^i \in \mathbb{Z}_p \\ s_{n-1} &= -(2^{n-1})^{-1} \sum_{i=0}^{n-2} s_i 2^i \in \mathbb{Z}_q \end{aligned} \tag{6}$$

Note that

$$\begin{aligned} \sum_{i=0}^{n-1} r_i 2^i &= \sum_{i=0}^{n-2} r_i 2^i + r_{n-1} 2^{n-1} \\ &= \sum_{i=0}^{n-2} r_i 2^i + (-(2^{n-1})^{-1} \sum_{i=0}^{n-2} r_i 2^i) 2^{n-1} \\ &= \sum_{i=0}^{n-2} r_i 2^i - (2^{n-1})^{-1} (2^{n-1}) (\sum_{i=0}^{n-2} r_i 2^i) \\ &= 0 \in \mathbb{Z}_p \end{aligned} \tag{7}$$

and that the same works for $s_{n-1}, \ldots, s_0$.

Step 2.3 For each $i \in 0, \ldots, n-1$ Peggy creates two Pedersen Commitments:

$$\begin{aligned} C_i^{\mathbb{G}} &:= b_i G' + r_i G \in \mathbb{G} \\ C_i^{\mathbb{H}} &:= b_i G' + s_i G \in \mathbb{H} \end{aligned} \tag{8}$$

Note that $\forall i \in 0, \ldots, n-1$

$$\begin{aligned} \sum_{i=0}^{n-1} 2^i C_i^{\mathbb{H}} &= \sum_{i=0}^{n-1} 2^i b_i G' + \sum_{i=0}^{n-1} 2^i r_i G \\ &= (\sum_{i=0}^{n-1} 2^i b_i) G' + (\sum_{i=0}^{n-1} 2^i r_i) G \\ &= xG' + 0G = xG' \end{aligned} \tag{9}$$

where xG' follows from Equation 4 and $0G$ is obtained from equality in Equation 7. Obviously the equality in Equation 9 holds for $C_i^{\mathbb{H}}$ and s_i too.

The sets ${C_i^{\mathbb{G}}}_{i=0}^{n-1}$ and ${C_i^{\mathbb{H}}}_{i=0}^{n-1}$ are the Pedersen Commitments we mentioned at the beginning of this section.

Step 3

Signing in Abe et al. [1]

Signing in Noether [2]

2.5 Proof Verification

Verifying in Abe *et al.* [1]

Verifying in Noether [2]