TU WIEN Informatics

Adaptor Signature Based Atomic Swaps Between Bitcoin and a Mimblewimble Based Cryptocurrency

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Software Engineering & Internet Computing

by

Jakob Abfalter, BSc
Registration Number 01126889

to the Faculty of Informatics

at the TU Wien

Advisor: Univ. Prof. Dr. Matteo Maffei
Assistance: Dr. Pedro Moreno Sanchez

Vienna, 6th April, 2020

Jakob Abfalter	Matteo Maffei

Erklärung zur Verfassung der Arbeit

Jakob Abfalter, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 6. April 2020

Jakob Abfalter

Acknowledgements

Enter your text here.

Abstract

Enter your text here.

Contents

CHAPTER 1

Introduction

TODO

CHAPTER 2

Motivation & Objectives

TODO

CHAPTER 3

Preliminaries

3.1 Bitcoin

3.1.1 Bitcoin Transaction Protocol

3.1.2 Bitcoin Scaling and Layer Two Solutions

3.2 Privacy-enhancing Cryptocurrencies

3.2.1 Zero Knowledge Proofs

3.2.2 Range Proofs

3.2.3 Mimblewimble

The Mimblewimble protocol was introduced in 2016 by an anonymous entity named Jedusor, Tom Elvis [Jed16]. The author's name, as well as the protocols name, are references to the Harry Potter franchise. [1] In Harry Potter, Mimblewimble is a tongue-typing curse which reflects the goal of the protocol's design,which is improving the user's privacy. Later, Andrew Poelstra took up the ideas from the original writing and published his understanding of the protocol in his paper [Poe16]. The protocol gained increasing interest in the community and was implemented in the Grin [2] and Beam [3] Cryptocurrencies, which both launched in early 2019. In the same year, two papers were published, which successfully defined and proved security properties for Mimblewimble [FOS19, BCL+19]. In this section, we will explain the fundamental properties of the protocols which are relevant for this thesis. The section is ba
Compared to Bitcoin, there are some differences in Mimblewimble: [FOS19]

[1] https://harrypotter.fandom.com/wiki/Tongue-Tying_Curse
[2] https://grin.mw/
[3] https://beam.mw/

- Use of Pedersen commitments instead of plaintext transaction values
- No addresses. Coin ownership is given by the knowledge of the opening of the coins Pedersen commitment.
- Spend outputs are purged from the ledger such that only unspent transaction outputs remain.
- No scripting features.

By utilizing Pedersen commitments in the transactions, we hide the amounts transferred in a transaction, improving the systems user privacy, but also requiring additional range proofs, attesting to the fact that actual amounts transferred are in between a valid range. Not having any addresses enables transaction merging and transaction cut through, which we will explain a bit later. However, this comes with the consequence that building transactions require active interaction between the sender and receiver, which is different than in constructions more similar to Bitcoin, where a sender can transfer funds to any address without requiring active participation by the receiver.
Through transaction merging and cut-through and some further protocol features, which we will see later in this section, we gain the third mentioned property of being able to delete transaction outputs from the Blockchain, which have already been spent before. This ongoing purging in the Blockchain makes it particularly space-efficient as the space required by the ledger only grows in the number of UTXOs, in contrast to Bitcoin, in which space requirement increases with the number of overall mined transactions. Saving space is especially relevant for Cryptocurrencies employing confidential transactions because the size of the range proofs attached to outputs can be significant. Another advantage of this property is that new nodes joining the system do not have to verify the whole history of the Blockchain to validate the current state, making it much easier to join the network.
Another limitation of Mimblewimble- based Cryptocurrencies is that at least the current construction does not allow scripts, such as they are available in Bitcoin or similar systems. Transaction validity is given solely by a single valid signature plus the balancedness of inputs and outputs. This shortcoming makes it challenging to realize concepts such as multi signatures or conditional transactions which are required for Atomic Swap protocols. However, as we will see in 3.3 there are ways we can still construct the necessary transactions by merely relying on cryptographic primitives.

Mimblewimble Transaction

Transaction Struction:

- For two adjacent elliptic curve generators g and h. A coin in Mimblewimble is of the form $C = g^v + g^r, \varpi$, C is a so called Pedersen Commitment to the value v with blinding factor r. ϖ is a range proof attesting to the fact that v is in a valid range.

- As already pointed out, there are now addresses in Mimblewimble. Ownership of a coin is equivalent to the knowledge of its opening, so the blinding factor takes the role of the secret key.
- A transaction consists of $C_{inp} = (C_1, \ldots, C_n)$ input coins and $C_{out} = (C'_1, \ldots, C'_n)$ output coins.

A transaction is considered valid iff $\sum v'_i - \sum v_i = 0$ so the sum of all input values has to be 0. (Not taking transaction fees into account)
From that we can derive the following equation:

$$\sum C_{out} - \sum C_{inp} = \sum g^{v'_i} + g^{r'_i} - \sum g^{v_i} + g^{r_i}$$

So if we assume that a transaction is valid then we are left with the following so called excess value:

$$\mathcal{E} = g^{(\sum r'_i - \sum r_i)}$$

Knowledge of the opening of all coins and the validity of the transaction implies knowledge of $\mathcal{E}$. Directly revealing the opening to $\mathcal{E}$ would leak too much information, an adversary knowing the openings for input coins and all but one output coin, could easily calculate the unknown opening given $\mathcal{E}$. Therefore knowledge of $\mathcal{E}$ instead is proven by providing a valid signature for $\mathcal{E}$ as public key. Coinbase transactions (transactions creating new money as part of a miners reward) additionally include the newly minted money as supply s in the excess equation:

$$\mathcal{E} = g^{(\sum r'_i - \sum r_i)} - g^s$$

Finally a Mimblewimble transaction is of form:

$$tx = (s, C_{inp}, C_{out}, \mathcal{K}) \text{ with } \mathcal{K} = (l(\varpi), l(\mathcal{E}), l(\varsigma))$$

where s is the transaction supply amount, C_{inp} is the list of input coins, C_{out} is the list of output coins and $\mathcal{K}$ is the transaction Kernel. The Kernel consists of $l(\varpi)$ which is a list of all output coin range proofs, $l(\mathcal{E})$ a list of excess values and finally $l(\varsigma)$ a list of signatures.

3.3 Scriptless Scripts

3.4 Adaptor Signatures

3.4.1 Schnorr Signature Construction

3.4.2 ECDSA Signature Construction

CHAPTER 4

Mulitparty Fixed Witness Adaptor Signatures

4.1 General Notation

4.2 Cryptographic Primitives

4.3 Mulitparty Fixed Witness Adaptor Signature Scheme

We define a Generalized Multiparty Adaptor Signature Scheme from the standard construction of multiparty Schnorr signatures which are defined as follows:

```
$GEN()$
1 : $k \leftarrow\$ \mathbb{Z}_q$
2 : $r \leftarrow\$ \mathbb{Z}_q$
3 : return $(k, r)$

$GEN\_PT\_SIG(m, k, r, g^{k'}, g^{r'})$
1 : $e = h((m||g^k + g^{k'}||g^r + g^{r'}))$
2 : $sig\_prt = k + e + r$
3 : return $(sig\_prt, g^k, g^r)$

$VRF\_PT\_SIG(m, k, r, g^{k'}, g^{r'}, sig\_prt)$
1 : $e = h(((m||g^k + g^{k'}||g^r + g^{r'}))$
2 : return $g^{sig\_prt} = g^{k'} + g^{e*r}$

$FIN\_SIG(sig\_prt, sig\_prt', g^k, g^{k'}, g^r, g^{r'})$
1 : return $(sig\_prt + sig\_prt', g^k + g^{k'}, g^r + g^{r'})$
```

In order to have adaptable partial signature we add the following procedures

```
$APT\_PT\_SIG(sig\_prt, x)$
1 :  $sig\_part\_apt = sig\_prt + x$
2 :  **return** $(sig\_part\_apt, g^x)$
```

```
$EXT\_WIT(sig\_final, sig\_prt, sig\_part\_apt')$
1 :  $sig\_prt' = sig\_final - sig\_prt$
2 :  $x = sig\_part\_apt' - sig\_prt'$
3 :  **return** $(x)$
```

```
$APT\_PT\_SIG(sig\_prt, x)$
1 :  $e = h((m||g^k + g^{k'}||g^r + g^{r'}))$
2 :  **return** $g^{sig\_part\_apt'} = g^{k'} + g^{e*r} + g^x$
```

CHAPTER 5

Adaptor Signature Based Atomic Swaps Between Bitcoin and a Mimblewimble Based Cryptocurrency

5.0.1 Construction Bitcoin side

5.0.2 Construction Grin side

5.0.3 Security Definitions

CHAPTER 6

Implementation

CHAPTER 7

Implementation Security and Privacy Evaluation

7.1 Security Evaluation

7.2 Privacy Evaluation

CHAPTER 8

Related and Future Work

8.1 Payment Channel Networks on Grin

8.2 Payment Channel Networks on Monero

8.3 Atomic Swaps With Related Cryptocurrencies

8.4 Tumbler Based Atomic Swaps

CHAPTER 9

Conclusion

List of Figures

List of Tables

List of Algorithms

Bibliography

[BCL$^+$19] Gustavo Betarte, Maximiliano Cristiá, Carlos Luna, Adrián Silveira, and Dante Zanarini. Towards a formally verified implementation of the mimblewimble cryptocurrency protocol. *arXiv preprint arXiv:1907.01688*, 2019.

[FOS19] Georg Fuchsbauer, Michele Orrù, and Yannick Seurin. Aggregate cash systems: a cryptographic investigation of mimblewimble. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 657–689. Springer, 2019.

[Jed16] Tom Elvis Jedusor. Mimblewimble, 2016.

[Poe16] Andrew Poelstra. Mimblewimble, 2016.