The University of Sydney

School of Information Technology

Thesis of Master of IT

Secure Unforkable Blockchain Wallet

Author:
Lin Han

Supervisor:
Dr. Vincent Gramoli

November 8, 2017


SIDERE MENS EADEM MUTATO

Contents

1 Introduction

From the time when Block 0 of Bitcoin blockchain, the Genesis Block, is created at 18:15:05 GMT on January 3rd, 2009, the words "cryptocurrencies" and "Blockchain" become one of the most popular topics in information technology fields. The "decentralized" and "anonymous" nature of cryptocurrencies overcomes the weakness of traditional *trust-based* electronic payments who relies heavily on trusted third-party financial institutions. The *cryptographic proof-of-work* of bitcoin enables reliable transaction between two parties directly[12]. Through almost 10 years development, cryptocurrencies turns out to be a large family and can be accessed via desktop, laptop, or even mobile devices like smart phones.

Though bitcoin gives a practical solution on traditional *double spending* problem in digital currencies, this doesn't mean that it is secure in all aspects. One possible issue is bitcoin blockchain's forkable feature. Actually, other cryptocurrencies allowing forkable chains all suffer from the very same issue. In this sense, unfokable blockchain is proven to overcome this shortcomming[5]. To consider in another aspect, another possible solution to secure transaction is to adopt mechanism like *zero knowledge contingent payment* which are released if and only if some knowledge is disclosed by the payee and to do this in a trustless manner where neither the payer or payee can cheat[16][15]. Whilst there are several theoretical discussion and practices in a variety of contexts, this paper will concentrate on how to develop a secure cryptocurrency wallet on the basis of unforkable blockchain and zero knowledge contingent payments.

In this project, I intend to contribute in two separated directions: one is to develop a mobile client for Red Belly Blockchain - a practice to deploy unforkable blockchain onto modern mobile devices; the other is to explore the way to enable zero knowledge contingent payments within Red Belly Blockchain network. The primary goal of this research is to examine the feasibility of real-life secure application for unforkable blockchain like Red Belly Blockchain.

In the following report, related works will be given first. Essential concepts and problems will be addressed in this part. After the literature review part, detailed report on implementation of secure wallet and zero knowledge contingent payments will be shown. Last but not the least, current outcomes of this project will be discussed. And finally, a conclusion as well as possible improvements will conclude this paper.

2 Literature Review

2.1 Bitcoin

Bitcoin is the first decentralized digital currency as well as a digital payment system. The whole system is peer-to-peer based, whose transactions are between users directly without participation of intermediary. Transactions are verified by network nodes - known as *mining* - and recorded in a public distributed ledger called *blockchain*.

2.1.1 Blockchain

Blockchain is the way how Bitcoin keeps its public ledger within its peer-to-peer network. To some extent, blockchain is a peer-to-peer distributed timestamp server. The ultimate goal of this design is to solve *double-spending* problems and prevent modification of transaction records[12].

Each full node in the Bitcoin network keeps a full copy of the blockchain, in which all blocks validated by this particular is stored. When several nodes within the network independently arrive at identical blockchains, they are considered to be in *consensus*. As its name suggests, a blockchain is a digital chain of blocks, where a timestamp, a nonce, and a Merkle Tree is stored. The blocks are chained cryptographically using hash. In detail, each block contains the hash of its previous block ,finally leading to the Genesis Block. Any modification on blocks in the chain would violates all subsequent hashes, which is vital for consistency of the ledger. Figure 1 shows part of a blockchain.

However, computing a hash is expensive. This truth enables *proof-of-work* in bitcoin network.

2.1.2 Proof of Work

According to blockchains' feature, a huge amount of computation is required in the generation of each block. Meanwhile, there is a *proof-of-work* mechanism to make the distributed timestamp server work and determine representation in majority decision making. Especially, when there are multiple chains (forks), consensus rules will pick up the longest chain, which contains the most proof of work during its generation[12].

In this way, any malicious changes on previous blocks would violate its following blocks. That is to say, hacker with huge computing power can

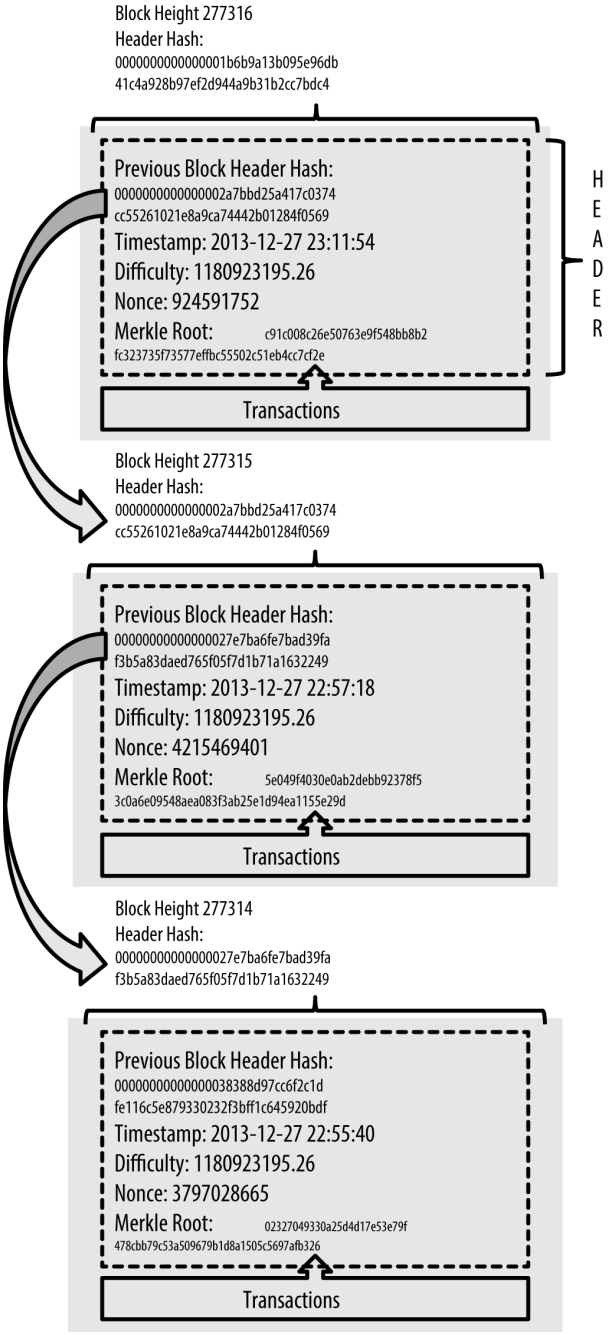

Figure 1: Blockchain

hijack the blockchain if he can generate the longest chain from the block he hacks. In turn, he has to own more than half of the computing power within the whole blockchain network[14].

2.1.3 Contracts

There are distributed contracts in Bitcoin transactions for agreement enforcements, which provides another way to formalize and guarantee agreements rather than traditional court system. Examples include Escrow, Micropayment channels and CoinJoin.

Some of the contracts can be implemented in Bitcoin Script, especially the zero knowledge contingent payments in Bitcoin is achieved with support of bitcoin scripts. However the Red Belly Blockchain doesn't have a robust scripting language like Bitcoin does.

2.2 Ethereum

2.2.1 Previous Work of Bitcoin

Bitcoin provides a protocol allowing weak implementation of *smart contracts*. However, several limitations exists in Bitcoin's scripting language:

1. **Not Turing-Completeness** - Bitcoin scripts lacks loops.
2. **Lack of States** - UTXOs scripts is only for one-off contracts.
3. **Blindness of Blockchain** - Bitcoin scripts cannot access blockchain data.
4. **Blindness of Value** - Bitcoin either consumes the entire UTXO or none of it

2.2.2 Rationale

Ethereum implements a blockchain with Turing-complete scripts, states, value awareness and blockchain awareness, which enables development of smart contracts, and even new protocols[17].

2.3 Balance Attack

As the previous review mentioning, to attack a blockchain, or specifically to rewrite the content of a block, the hacker should have more than half of the computing power of the whole blockchain network which is almost unfeasible in real world. In particular, by delaying the propagation of blocks in Bitcoin system, the hacker can in result delay the growth of the longest branch of the system. In other word, he can then hijack the blockchain even without a large amount of computing power. Ethereums' "Blockchain 2.0" somehow fixes this problem, but there is still other possible method against forked blockchain. One practice is the **Balance Attack** [6][14][13]against *proof-of-work* blockchain systems.

To achieve a balance attack within the blockchain network, the attacker should divide the network into subgroups of similar mining power by cutting off their communications. During this down time, the attacker issues transaction in the *transaction group*, and mine blocks in the *block group* simultaneously. This action only ends when it comes to the point where the tree of the block subgroup outweighs the tree of the transaction group, which is with high possibility. The balance, in result, can leverage the *GHOST* protocol that accounts for sibling or uncle blocks to determine on a chain of blocks. This strategy allows the attacker to mine a branch regardless of the rest of the network so that he can influence the branch determination process while merging[14]. The process is as shown in Figure 2.

2.4 Unforkable Blockchain

2.4.1 Byzantine Consensus Problem

The *Byzantine Consensus Problem* refers to the *Byzantine General's Problem* proposed by Leslie Lamport, Robert Shostak and Marshall Pease in 1982. The problem is complicated by the presence of traitorous generals who may not only cast a vote for a suboptimal strategy, they may do so selectively. All the votes and results are simplified to attack or retreat. The problem is complicated further by the generals being physically separated and having to send their votes via messengers who may fail to deliver votes or may forge false votes[9].

In computer science area, typically computers or participants in network are mapped to generals and links between them are mapped to messengers.

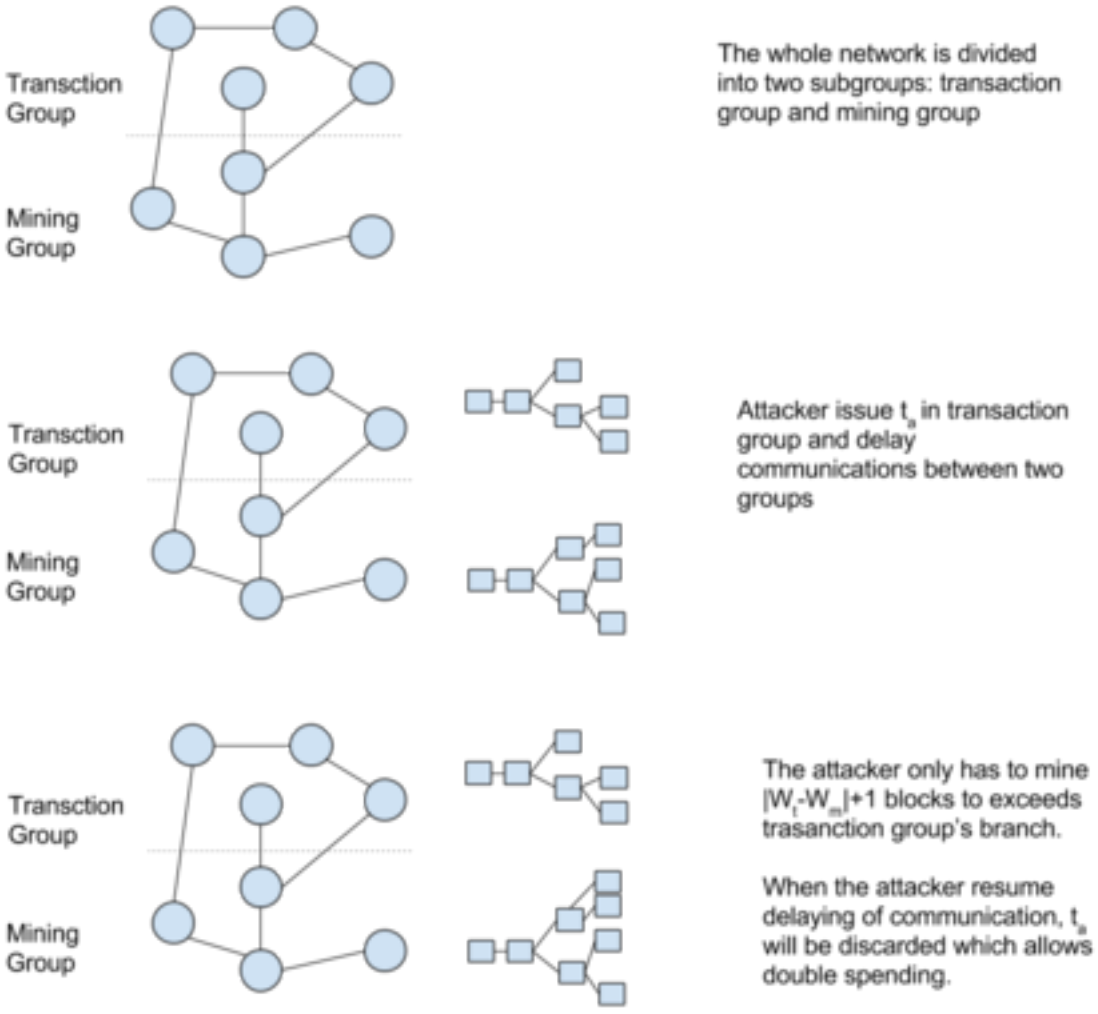

Figure 2: Balance Attack

2.4.2 Traditional Blockchain Byzantine Consensus Problem

The *proof-of-work* of Bitcoin blockchain is the primary solution to *Byzantine Consensus Problem*.

In detail, a model of Bitcoin network can be built upon the classic Byzantine Consensus Problem. The distributed system is the alliance of generals, in which the upper bounds on the delay of communicating and decision-making is unknown.

However, regarding to our previous discussion on *Balance Attack*, attackers can still disrupt the consensus system to beat the Bitcoin Byzantine Consensus[7]. Because there is no guarantee that the decided value is proposed by a valid process.

2.4.3 Democratic Byzantine Consensus

Democratic Byzantine Fault Tolerance is a system specially tailored for consortium blockchains based on *Binary Byzantine Consensus*. In *Binary Byzantine Consensus*[11], each trusted participant issue proposal with either 0 or 1 and decides that the final agreement such that:

1. No pair of trusted participants have different decision.
2. Every trusted participant decides
3. If all correct participant propose the the same value, then no other value can be deiced

In safe Democratic Byzantine Fault Tolerance[5], a mechanism called binary broadcast for binary Byzantine Consensus system is adopted. To conclude, there are four aspects that is strictly followed within DBFT:

1. **Binary Value Obligation** - if t+1 correct BV-broadcats v, then v is eventually added to the set binary values of all correct process.
2. **Binary Value Justification** - if p_i is correct and and v in $binary - values_i$ then v was broadcasted by a correct process.
3. **Binary Value Uniformity** - if v is added to $binary - value_i$ of correct p_i, then eventually v will be in $binary - value_j$ for all correct p_j.
4. **Binary Value Termination** - eventually $binary - value$ of correct p_i is not empty.

2.5 The Red Belly Blockchain

Red Belly Blockchain is new blockchain relying on the Democratic BFT. The *Genesis Block* of this blockchain contains the initial information as well as a list of n participants. All accesses of external nodes requires these participants as the middleware. In this case, a transaction is regarded as committed if $t + 1$ participating nodes agreed so it can be written into next block.

The performance of this new blockchain is quite good. It can achieve more than 400 transactions per second and with great scalability up to 90 server nodes.

2.6 Zero Knowledge Contingent Payment

2.6.1 Bitcoin ZKCP

Zero Knowledge Contingent Payment in Bitcoin is first proposed by Gregory Maxwell in 2011 on Bitcoin Wiki[10]. The basic process can be illustrated by an example[3]:

> Alice is a fan of Sudoku puzzle. However, there is a puzzle that she is trying days but in vain. She gives up and broadcasts a message within a fan group proclaiming that "I will pay whoever solves this puzzle." Bob see the broadcast, solves it, and want to sell the result to Alice. The problem is either Alice or Bob is willing to be the first person to give out what they have.
>
> To solve this dilemma, Alice and Bob goes for a Bitcoin, which allows one to issue a transaction and also specify the conditions to be met in order to claim the transaction. In this case, Alice propose a payment transaction to blockchain that includes encoded Sudoku puzzle and the rules. Whoever solves the puzzle is able to get the fund.
>
> Using the Bitcoin ZKCP protocol, Bob knows a solution s and encrypts the solution using a key k such that $Enc_k(s) = c$ and he computes y such that $SHA256(k) = y$. He then send the key k and c to Alice together with a zero knowledge proof that c is an encryption of s under the key k and that $SHA256(k) = y$. Once Alice has verified the proof, she creates a transaction

to the blockchain that pays Bob n bitcoins, and says that Bob can only claim the coins if he provides the value k' such that $SHA256(k') = y$. Bob then published k and claims the fund. In this way, Alice learns k can decrypt c so that she knows the solution s

Specifically in Blockchain, the ZKCP protocol now has several practices rather than theory. One is ZK-SNARK protocols allowing for the practical implementation of the necessary proofs[8][2][4].

2.6.2 Zero Knowledge Contingent Payment without Scripts

However, all implemented ZKCP protocols now are based on scripting language in traditional forkable blockchain[1]. Besides, there is no convincing data of its performance on modern mobile devices[18].

3 Conclusion

In this literature review, we can see that unforkable blockchain has a more reliable security guarantee compared to traditional forkable blockchain used in mainstream cryptocurrencies. The need to deploy unforkable blockchain and implement ZKCP-enable protocols onto mobile devices is evident through the literatures to fulfill the gaps between the theoretical discussion and real life application. Deploying ZKCP-enbale unforkable blockchains onto mobile devices will benefit both authorities seeking reliable blockchain solutions and users having trust problems with third-parties.

References

[1] Wacław Banasik, Stefan Dziembowski, and Daniel Malinowski. *Efficient Zero-Knowledge Contingent Payments in Cryptocurrencies Without Scripts*, pages 261–280. Springer International Publishing, Cham, 2016.

[2] Eli Ben-Sasson, Alessandro Chiesa, Matthew Green, Eran Tromer, and Madars Virza. Secure sampling of public parameters for succinct zero knowledge proofs. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 287–304. IEEE, 2015.

[3] Matteo Campanelli, Rosario Gennaro, Steven Goldfeder, and Luca Nizzardo. Zero-knowledge contingent payments revisited: Attacks and payments for services.

[4] Ioannis Chatzigiannakis, Apostolos Pyrgelis, Paul G. Spirakis, and Yannis C. Stamatiou. Elliptic curve based zero knowledge proofs and their applicability on resource constrained devices. 2011.

[5] Tyler Crain, Vincent Gramoli, Mikel Larrea, and Michel Raynal. (leader/randomization/signature)-free byzantine consensus for consortium blockchains. *CoRR*, abs/1702.03068, 2017.

[6] Vincent Gramoli. On the danger of private blockchains. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers (DCCL'16)*, 2016.

[7] Vincent Gramoli. From blockchain consensus back to byzantine consensus. *Future Generation Computer Systems*, 2017.

[8] Yael Tauman Kalai and Ran Raz. Succinct non-interactive zero-knowledge proofs with preprocessing for logsnp. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 355–366. IEEE, 2006.

[9] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

[10] G Maxwell. Zero knowledge contingent payment. 2011. *URl: https://en. bitcoin. it/wiki/Zero_Knowledge_Contingent_Payment (visited on 05/01/2016).*

[11] Achour Mostéfaoui, Hamouma Moumen, and Michel Raynal. Signature-free asynchronous binary byzantine consensus with t¡ n/3, o (n2) messages, and o (1) expected time. *Journal of the ACM (JACM)*, 62(4):31, 2015.

[12] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[13] Christopher Natoli and Vincent Gramoli. The blockchain anomaly. In *Proceedings of the 15th IEEE International Symposium on Network Computing and Applications (NCA'16)*, pages 310–317. IEEE, Oct 2016.

[14] Christopher Natoli and Vincent Gramoli. The balance attack or why forkable blockchains are ill-suited for consortium. In *Proceedings of the 47th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'17)*. IEEE, Jun 2017.

[15] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 459–474. IEEE, 2014.

[16] Bitcoin Wiki. Zero knowledge contingent payment, 2011.

[17] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151, 2014.

[18] Fucai Zhou, Yuxi Li, Qingshi Zhou, Jingwei Miao, and Jian Xu. The electronic cash system based on non-interactive zero-knowledge proofs. *International Journal of Computer Mathematics*, 93(2):239–257, 2016.