Univeristé Libre de Bruxelles

Preparatory Work For Master Thesis

An academic approach about Bitcoin

OP_CODE

by

Thomas Suau

Supervisors: **Pr. Olivier Markowitch**
Pr. Etienne Riviere

June, 2024

Abstract

In this Preparatory work of Master Thesis I want to focus on the study of Bitcoin [1] Script and look into recent evolution through segwit and tapscript [2, 3] to accommodate new use cases.

The main goal of this work is to purpose a research problem about messaging Protocols built with Bitcoin Script and look into recent usage in the Bitcoin network. Few years ago the Lightning Network [4] was introduced as a second layer solution to the Bitcoin network, to allow micropayments. Lightning Network is using a combination of specific multi-signatures transaction with `SIGHASH_NOINPUT` and a precise structure for the commitment transaction [section 3 [4]].
More recently, the Taproot [3] upgrade was activated on the Bitcoin network. The Taproot upgrade introduced a lot of new script functionalities and new `OP_CODE` like `CHECKSIGFROMSTACK`. The Taproot upgrade is expected to improve the privacy and scalability of the Bitcoin network. Following this update a new protocol emerges: The Ordinals Protocol [5]. Following this new kind of protocol, and regarding to the new "script templates" introduced by the Taproot upgrade, I want to study the new possibilities of the Bitcoin Script and the new usage of the Bitcoin Script in the Bitcoin network.

The Bitcoin Script is a simple stack-based programming language used to define the conditions under which a transaction output can be spent. The Bitcoin Script is a powerful tool that allows the creation of complex transactions.

Keywords

Bitcoin, Bitcoin Script, Lightning Network, Taproot, Ordinals Protocol

Contents

1 Introduction

1.1 Bitcoin transaction

Before to dive into Bitcoin script usage for Lightning Network (LN) [4] and messaging protocols a quick overview about Bitcoin transaction is necessary.

There is a complete tutorial about Bitcoin transaction by chaincodelabs [6]. This GitHub repository is explaining in details how a Bitcoin transaction is built and signed.

I just want to highlight some important points about Bitcoin transaction. Everything sent on the Bitcoin network is a transaction. Every transactions are sent into hexadecimal format, interpreted by the network as a combination of operations, addresses, values and inert data.
We can highlight that Bitcoin is an UTXO-based model which stands for Unspent Transaction Output. Every transactions are built from such unspent output.

Figure 1: Hexadecimal of the first bitcoin transaction

Build an algorithm to colour transactions should be a first step for the research process about Bitcoin script usage. Indeed, it will allow to better understand the transaction structure and how to manipulate it.
To analyse a transaction at the lowest level we need to decode the hexadecimal format. The first transaction ever made on the Bitcoin network is the one from Satoshi Nakamoto Hal Finney. The hexadecimal format of this transaction is shown in the figure 1.
In this transaction scripts are respectively in green and in orange. The green script is the scriptSig and the orange script is the scriptPubKey. We detail further this two kind of script.

1.2 Bitcoin script

Bitcoin script is a stack-based language. An UTXO has a lock script and transaction has an unlock script.
The most common type of transactions called standard transaction are payment "From

Alice to Bob", we called them Pay to-Public Key Hash (P2PKH). Such transactions contain the scriptSig that unlocks the funds and the scriptPubKey that locks the output.
There are other type of transactions called non-standard transactions [7]. As we want to study protocols which are using Bitcoin Script we would focus on non-standard transactions built with custom scriptPubKey.
Validating a transaction requires executing both script and the Stack at the end must be empty. The Bitcoin script is composed of more than +200 operations called `OP_CODES`. The most used operations are:

- `OP_DUP` : ;
- `OP_HASH160`: ;
- `OP_EQUALVERIFY`: ;
- `OP_CHECKSIG`: ;

A very basic scriptPubKey is the following:

```
OP_DUP OP_HASH160 <PubKeyHash> OP_EQUALVERIFY OP_CHECKSIG
```

This script is used to send funds to a Bitcoin address (P2PKH) [7]. The scriptPubKey is composed of 5 operations. The first operation is `OP_DUP` which duplicates the top stack element. The second operation is `OP_HASH160` which hashes twice the top stack element: first by SHA-256 and then with RIPEMD-160. The third operation is `OP_EQUALVERIFY` which compares the top two stack elements. The fourth operation is `OP_CHECKSIG` which checks the signature.

In order to simplify usage of Bitcoin Script, Pieter Wuille, Andrew Poelstra, and Sanket Kanjalkar at Blockstream Research have developed a new language called Miniscript [8]. Miniscript is a subset of Bitcoin Script. It is a high-level language that allows to write Bitcoin Script in a more readable way. Miniscript is a very powerful tool to write Bitcoin Script.

One `OP_CODE` very used to define arbitrary protocols on Bitcoin is: `OP_RETURN`. Introduced in Bitcoin core as a standard operation in 2014 [9], it allows user to push

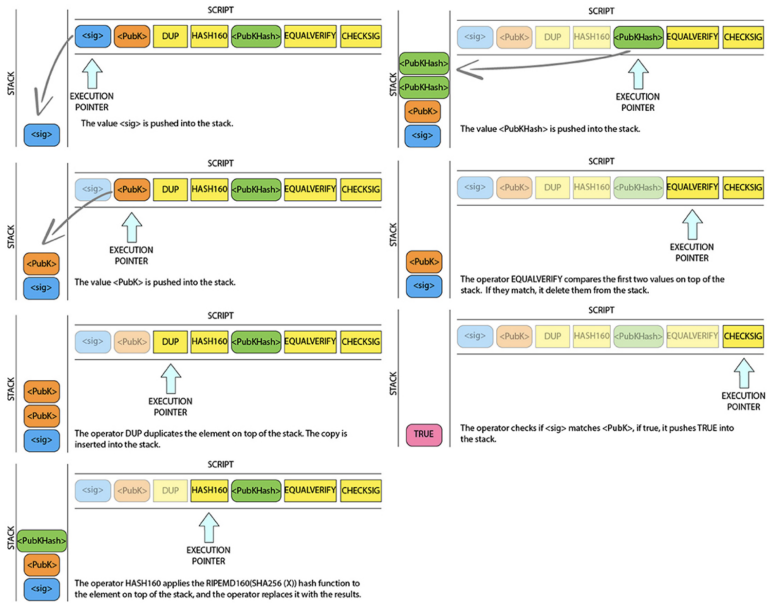

Figure 2: Execution by the Stack [7]

arbitrary data with a length of 83 bytes maximum. Data pushed by user is evaluated to false by the Bitcoin protocol.
In Bartoletti and Pompianu [9] p.7, we have a tab to summarise some usage of `OP_RETURN` in Bitcoin transactions. We can notice that the most used protocol was the `Colu` (CC) protocol. This protocol was used to create colored coins on the Bitcoin network. Colu was apparently a first initiative from Bankbox to allow Central Bank to issue digital currencies, where Colored Coins was a first experimentation on Bitcoin blockchain Colu Open-Sources Protocol to Help Central Banks Issue Digital Currencies.

1.3 State of the art

Blockchain is today a research topic for some Universities and Research Center in the world. But we can notice that a lot of Bitcoin research and Blockchain research articles are more focused on the price than on the technology itself. [Give stats].

The Lightning Network (LN) is a second layer protocol that allows to make instant and low fees transactions. This protocol is more complex and involves a lot of Bitcoin Script. The protocol is based on the concept of payment channels. The payment channels are a way to make off-chain transactions. They are leveraging unsigned Funding Bitcoin transactions (PSBT) and Commitment transactions, with some specific `OP_CODE`.

The LN produced a lot of academic research over the last years. In an attack perspective [10–12], to study network scalability [13] or to study the protocol itself [14].

Since many years, the concept of Vault and permissionned addresses is very discussed into the developper community. It's well know as Bitcoin *Covenants* [15, 16]. The problem can be described as followed: we want to enforced on the protocol side some addresses to make specific actions only. For example a company would want to enable its employee to use the bitcoin (BTC) of the company without allowing them to spend the Bitcoin for themselves. They may want to enforce on the protocol side rules for address to spend the BTC. Other uses cases can be to enforce an address to spend funds only on specific whitelisted address and avoid the loss of BTC by malicious attackers.
The concept of Vault is pretty similar and wants for addresses to play the role of locker for a certain given time. Those approach can be very useful for professional and securities purpose. In another hand, it will create permissionned addresses and some actors are strongly against this upgrade. The exact implementation is still very discussed by Bitcoin Core developers, it's an active research field.
Some new computing power should be available with such upgrades and our work could help the research to better understand improvements and limitations of such implementations.

More recently, with Taprscipt [17] new kind of covenants where purposed as the use of `OP_CAT` [18] or `OP_CHECKTEMPLATEVERIFY` [3]. The Bitcoin Covenants are a very discussed topic by Bitcoin core developper and it should be considered as a research work about Bitcoin script [19].

2 Goals

The goal of the thesis is to provide and reference actual research problems about protocols based on Bitcoin Scripts. One purpose of the Ethereum Virtual Machine (EVM) is to build contracts and deploy the binary code on Ethereum. Those contracts can be executed by the EVM from on-chain calls. Bitcoin miners cannot verify application-specifics checks from given data. Understanding the power and limits of this mechanism could be a first direction for this work. For example, the Professor E. Riviere purposed: can we implement anything we can on Ethereum through these [smart contracts] blobs (binary large objects)?

A more general question could be: is it possible to use Ordinals' enveloppe [5] to push blobs and execute its logic?

BitVM [**?**]: a first

3 Methodology

Bibliography

[1] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, bitcoin.org (2008).

[2] K.-J. Alm. *Generic Signed Message Format.* bip-0322, (2018).

[3] A. T. Pieter Wuille, Jonas Nick. *Taproot: SegWit version 1 spending rules.* bip-0341, (2020).

[4] D. Joseph, Poon ; Thaddeus, *The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments*, (2016).

[5] O. O. Institute. *Ordinals Handbook.* Official Documentation, (2024).

[6] Chaincodelabs. *Bitcoin Transaction Tutorial.* Bitcoin Transaction Tutorial, (2022).

[7] S. Bistarelli, I. Mercanti, and F. Santini. *An Analysis of Non-standard Bitcoin Transactions.* In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 93–96, (2018).

[8] A. P. Pieter Wuille and S. Kanjalkar. *Bitcoin Miniscript.* Miniscript, (2019).

[9] L. P. Massimo Bartoletti, *An Analysis of Bitcoin OP_RETURN Metadata*, Financial Cryptography and Data Security (2017).

[10] A. M. M. Ali Abdullah, *An Invitation Model Protocol (IMP) for the Bitcoin Asymmetric Lightning Network*, Computer Science and Symmetry/Asymmetry: Feature Papers (2023).

[11] R. Y. Antoine Riard, Gleb Naumenko, *Time-Dilation Attacks on the Lightning Network*, Computer Science and Symmetry/Asymmetry: Feature Papers (2021).

[12] E. R. J. M. F. Tschorsch, *Discharged Payment Channels: Quantifying the Lightning Network's Resilience to Topology-Based Attacks*, IEEE Security & Privacy on the Blockchain 2019 (2019).

[13] Y. G. J. T. C. Feng, *A Measurement Study of Bitcoin Lightning Network*, 2020 IEEE International Conference on Blockchain (Blockchain) (2020).

[14] E. R. J. M. F. Tschorsch, *On the impact of attachment strategies for payment channel networks*, 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (2021).

[15] M. Möser, I. Eyal, and E. Gün Sirer. *Bitcoin Covenants.* In *Financial Cryptography and Data Security.* Springer Berlin Heidelberg, (2016).

[16] R. O'Connor and M. Piekarska, *Enhancing Bitcoin Transactions with Covenants*, (2017).

[17] A. T. Pieter Wuille, Jonas Nick. *Validation of Taproot Scripts.* bip-0342, (2020).

[18] E. Heilman and A. Sabouri. *OP_CAT.* BIP-OPCAT.

[19] K. Swambo and S. . a. Hommel, *Bitcoin Covenants: Three Ways to Control the Future*, arXiv (2020).

[20] R. Linus. *BitVM: Compute Anything on Bitcoin.* Whitepaper, (2023).