Bitcoin Addresses. Scaling, Migration and Payment Perspectives

Wai Kok Chan, Ji-Jian Chin, Vik Tor Goh
Faculty of Engineering
Multimedia University
Cyberjaya, Selangor, Malaysia

Abstract

Since Satoshi Nakamoto launched bitcoin in 2009, there have been many improvements in bitcoin addresses required to fulfill newer requirements. These improvements have made the learning curve steeper for a regular user. Since 2015, many advanced features such as Segregated Witness, TapRoot, "Payment Protocols" and many others were introduced into bitcoin. Even advanced bitcoin users find these features hard to comprehend. This paper describes how scaling, migration, and payment protocol are indirectly related to bitcoin addressing. The foundation of many new features in bitcoin lies significantly in its addressing. From bitcoin addressing, transaction scripting can be modeled first. Later on, it is followed with specific features described in each protocol. This paper summarized the main highlight in many bitcoin standards. Readers will be able to gain a quick and clear understanding of advanced bitcoin features.

Keywords: bitcoin address scaling, bitcoin address migration

1. Introduction

In a bitcoin transaction, there are input and output. Input can be a coin generation transaction or a spent transaction. The first miner that computed the nonce that satisfied the current difficulty level (Proof of Work) is the winning miner. The winning miner has the right to produce one bitcoin block and create a coin generation transaction. As of June 2020, a coin generation transaction creates 6.25 bitcoin as a mining reward and is sent to the miner's bitcoin address.

A spent transaction is a transaction where bitcoin users spent the fund in an address belonging to them. Thus, users' public keys and signatures are required before spending is authorized. The output transaction specified the receiver in the form of a script such as P2PK (Pay To Public Key), P2PKH (Pay To Public Key Hash), P2SH (Pay To Script Hash)[1], P2WPKH (Pay To Witness Public Key Hash), and P2WSH (Pay To Witness Script Hash). For each transaction, the total input amount must match the total output amount. In a bitcoin transaction, every input is fully spent even though the user spent a fraction of his/her fund. The user fund's balance is sent to a change output address, which can be the same or different address (preferred for privacy reason). In this way, bitcoin nodes do not need to keep track of the full transaction records after verification. It only needs to keep track of all unspent output for transaction operation.

The foundation of many bitcoin features lies significantly in how bitcoin addressing is designed. Thus, the fundamental aspect of bitcoin addressing should be clearly explained thoroughly. Due to bitcoin's rapid development, few academic resources explain how bitcoin addresses evolution. Users are left with documentation on the bitcoin wiki page [2] to learn more about bitcoin address evolution. Many fragmented articles focus on specific bitcoin addressing topics. However, they do not solely focus on bitcoin address evolution. In one of our previous paper [3], we have described bitcoin addresses in great detail. That paper covered the aspect of bitcoin public key, private keys, wallets, multi-signatures, scripting, segregated witness, and bitcoin fork.

The growing requirements in bitcoin, new bitcoin features becoming increasingly difficult to comprehend for regular readers. It also becomes less clear as to how some bitcoin features are related to its addressing. This paper describes some of the advance addressing features used in bitcoin and how it affects bitcoin operation. The rest of this paper is organized in this manner. Section II describe some method used to improve the bitcoin scalability such as Schnorr Signature, MAST (Merkelized Abstract Syntax Tree) and TapRoot. Section III describes the need for Proof of reserves. In Section IV, some migration issue related to Segregated Witness and Taproot is discussed. Payment Protocol is described in Section V. Finally, the paper concluded in Section VI. In this paper, many references were taken from the bitcoin improvement protocol (BIP)[1]. For simplicity of reference, BIP70 will refer to BitCoin Improvement Protocol 70, "Payment Protocol".

2. Scalability Improvement

2.1. Schnorr Signature

A 3-of-5 multi-signature transaction is 381 bytes consists of three signatures (72 bytes) and five public keys (33 bytes). The need to include most of the signature and public keys could potentially discourage multi-signature

 | DOI: 10.20533/ijisr.2042.4639.2021.0112

usage in bitcoin. In Schnorr signature [4], all signatures in M-of-N multi-signature or a CoinJoin [5] transaction can be aggregated into a single 64 bytes signature (X) 64 and a single public key (P) shown below.

$$P = P_1 + P_2 + ... + P_n$$

$$X = X_1 + X_2 + ... + X_n$$

Thus, this reduces the transaction size, verification time, and privacy. Observers cannot differentiate the transaction from a single signature transaction. On top of that, Schnorr's signature is proven to be secured. Multiple public keys can be aggregated into a single aggregated public key. The challenge is how to produce an aggregated signature from multiple signatures without relieving the actual private key.

In MuSig[4], proposed by Blockstream consists of three rounds. First, all parties send commitments t_i to prevent them from changing their nonce later on. In the second round, all parties reveal their nonce R_i to all parties for verification that $t_i = H(R_i)$. In the third round, all parties calculate the partial signature s_i where $s_i = k_i + H(X, R, m)*a_i*x_i$

$$s = s_1 + s_2 + \ldots + s_n$$

$$= (k_1 + \ldots + k_n) + H(X, R, m)*(a_1*x_1 + \ldots + a_n*x_n)$$

$$= k + H(X, R, m)*x$$

There are many proposals to shortcut the communication process in Schnorr's signature signing. However, Manu et al. [5] have proven that none of the existing two round M-of-N multi-signature schemes is secured. Thus, a secure Schnorr signature implementation requires three rounds of communication. As of January 2021, Schnorr's signature (BIP 340) is still under the draft stage.

2.2 MAST (Merkelized Abstract Syntax Tree)

In a complex payment [7], Alice wants to spend her fund anytime with some default conditions. If the fund is not spent within ten years, she wants her beneficiary, Bob and Carol, to be able to spend her fund. In standard bitcoin scripting, whoever spent the fund will have to provide the full script. Alice will happily spend her fund in most cases, but the script indirectly disclosed the entire contract. On top of that, the whole script will have to be stored in the blockchain, thus wasting block space and transaction fees. Storing the entire contract can potentially limit the size of the smart contract deployed.

OP_If

<Alice's pubkey> OP_CheckSig
OP_Else

"10 years" OP_CSV OP_Drop

2 <Bob's pubkey> <Carol's pubkey> 2 OP_CheckMultiSig

OP_EndIf

Table 1: Merkle Abstract Syntax Tree in Table form

Merkle Root (A)	
(B) <Alice's pubkey> OP_CheckSig	(C) "10 years" OP_CSV OP_Drop 2<Bob's pubkey> <Carol's pubkey> 2 OP_CheckMultiSig

MAST [8] was first described in BIP 114 to solve this problem. However, the proposal was rejected. From table 1, (A) is the Merkle root formed by hashing script (B) and script (C) together. In MAST, if Alice spent her fund, she will provide the script (B), which is computed during run time. Merkle Root (A) and the hash of script (C) are included in the blockchain. Blockchain viewers will not know the detail of (C). If Bob and Carol spent the fund, they would provide the script (C), which is computed during run time. Merkle Root (A) and the hash of script (B) will be included in the blockchain. Blockchain viewers will not know the detail of (B). MAST enhances the smart contract transaction privacy and reduces the transaction size. Thus, a longer smart contract size can be supported. There are three BIP proposals for MAST, which are BIP114, BIP116, and BIP117. Currently, MAST is still not a standard feature in bitcoin. It took two years to get segregated witness implemented in bitcoin. Thus, implementing new features in bitcoin is not a straightforward process.

2.3. TapRoot

In MAST, the selected script for execution is shown in the blockchain. An example is shown in Table 1. If the script (C) is executed, the observer knew that the fund was not spent within ten years and were finally spent by Bob and Carol. To further enhance privacy and scalability, TapRoot is proposed in BIP 341 and BIP 342. The first process is to tweat the keys, as shown below.

$$P_{new} = P_{old} + H(P_{old}, m).\ G$$

 | DOI: 10.20533/ijisr.2042.4639.2021.0112

Where P is the public key, m is the committed message. If the m message is replaced with a Merkle root of MAST, then the public key P is the commitment to a Merkle root. The Merkle root, is the summary of all the scripts in the smart contract. The fund is sent to P_{new}. In order to spend the fund, the user can use either aggregated signature or a threshold signature. In aggregrated signature, each party sign their partial signatures and use all the signed partial signature to construct an aggregated signature. A threshold signature is a subset of aggregated signatures. In threshold signatures, such as 2-of-3 multi-signature, when the number of signing parties reached a threshold, a threshold signature can be produced. In TapRoot protocol, sending and spending fund transactions are indistinguishable from P2PKH or P2SH as they look like single-party transactions. Transaction indistinguishability improves bitcoin privacy. Taproot also reduces the number of bytes in a transaction, thus improve the blockchain scalability. There is some tradeoff in TapRoot. In a multi-signature environment, the number of possible combinations in a Merkle tree grows exponentially, for example, ${}^{N}_{M}C$ possible combinations for M-of-N multi-signature. It is computationally expensive and may be impossible to build a MAST tree with a large ${}^{N}_{M}C$. In this scenario, it may be better to fall back on the traditional bitcoin multi-sig transaction, revealing more private information and taking up more blockchain space. TapRoot requires three rounds of interactive communication between parties to construct the aggregated signature and aggregate public key. In most cases, communication is difficult when the number of parties grows.

3. Proof of Reserves

Specific companies such as bitcoin exchange, custodian services, investment holding, and many others may be holding bitcoin on behalf of their customers. Each customer has the right to know whether these companies are still holding their customer bitcoin. Thus, proof of reserve, as stated in BIP127, is essential. Usually, a bitcoin exchange fund is stored in cold storage protected with multi-signature private keys.

On 24th May 2014, a bitcoin exchange called Bitstamp performed a transaction to merge all their small input into a single output of 184,497 bitcoins [9]. This transaction proved their bitcoin ownership. However, this transaction is dangerous and tends to attract hackers. According to BIP127, an exchange can construct a single transaction consist of all its' Bitcoin UTXOs along with a single invalid input. Thus, this transaction will be rejected by the network upon broadcast. However, it can be used as definitive proof that the exchange is still holding their customer fund. At the same time, there is no fee involved as the transaction is rejected by the network. There are still many unresolved issues, and it is an active area of research. From the customer's perspective, they prefer proof that the exchange is solvent and still have their fund. The exchange cannot reuse one customer's proof for another customer. From the bitcoin exchange perspective, they must ensure their customer privacy is preserved.

For the Kraken Bitcoin exchange, the initial proof of reserve was carried out by Stefan Thomas in 2014 [10]. Only a public statement is published back then. The process was furthered refined, and the user can verify their proof of reserve on the Kraken website. The detailed process is published on the Kraken website [11].

4. Migration issues in segregated Witness and Taproot

4.1. Lack Of Evidence for Court Prosecution

In a segregated witness BIP141, only the hash of the signature is stored in the blockchain for each transaction ID. The signer of this transaction can be requested to provide the signature in court. However, there is some problem with a multi-signature transaction. A 2-of-3 multi-signature transaction assumes a transaction "ABC" can be approved by any two of the following personnel: the finance officer, manager, and CEO. Later on, in a court proceeding, it was found that this transaction is involved in criminal activity. There is no direct evidence to pin-point the responsible parties [12]. The use of a hardware deterministic wallet may further complicate the investigation process. The investigation officer may need to demand the responsible parties to reproduce the signature.

The responsible parties may or may not have technical incompetency to retrieve the related private keys that sign the "ABC" transaction. It may be a tedious process if the hardware wallet is used repeatability for high volume transactions. Even if they have the technical competency, they can deny it.

4.2. Lack of Privacy during Migration

Each of the following transaction scripts, P2PKH, P2WPKH, P2SH, P2WPSH, and P2TR can be uniquely identified. Mixing any of these scripts in a single bitcoin transaction can result in privacy leakage[13]. P2SH was introduced in 2012, and P2WPKH and P2WSH were introduced in 2017. Let assume the following example, assume a user sent his bitcoin from his hardware wallet to a bitcoin exchange, as shown in Table 2.

A bitcoin exchange wallet will most likely be using a multi-signature wallet with addresses that start with "3". A similar case can also happen when a non-segregated witness user sent some funds to a segregated witness user. From these transactions shown in Table 2, an observer can conclude that addresses start with "1111" belong to the same person.

Table 2: P2PKH input with P2SH output

User with 10 BTC sent 1 BTC to a bitcoin exchange	
Input	Output
P2PKH 10 btc *Add : 1111aaaa*	P2SH 1 btc Add: 3333cccc

	P2PKH (8.99 btc return change) *Add: 1111bbbb*
A non-segregated witness User with 10 BTC sent 1 BTC to another segregated user	
P2PKH 10 btc *Add : 1111cccc*	P2WPKH 1 btc Add: bc133aaaa
	P2PKH (8.99 btc return change) *Add: 1111dddd*

Thus, in order to ensure privacy, the input and output transaction types should be the same. During the process of migration, all the possible script combinations are shown in Table 3. Only three combinations are considered acceptable from a privacy point of view, as Nikita Zhavoronkov [13]. Thus, in any migration process, there is a long period of co-existence. Therefore, mix-transactions will exist for an extended period. Only 27% of the bitcoin transaction can be traced when P2SH is introduced. When Segregated Witness is introduced, this figure is increased to 55% [14]. The introduction of Taproot will increase it further. Therefore, it can be concluded that the introduction of Taproot will degrade bitcoin user privacy.

Table 3: P2PKH Input Script

Exchange	User P2PKH	User P2WPKH	User P2TR
P2PKH	Ok	BAD	BAD
P2SH	BAD	BAD	BAD
P2WPKH	BAD	Ok	BAD
P2WSH	BAD	BAD	BAD
P2TR	BAD	BAD	Ok

4.3. Lack of Insentive for Taproot Migration

Table 4: P2PKH Input Script

	Output size (Single signature)	Input Size (Single Signature)
Default (P2PKH)	34	148
Wrapped Segwit (P2SH-P2WPKH)	32	91
Native Segwit (P2WPKH)	31	68
TapRoot (P2TR)	43	58

In Segregated Witness, there is a 50-70% reduction in transaction fees. Thus users are motivated to migrate. In Taproot, the saving is minuscule as compared to Segregated Witness, as shown in Table 4. On top of that, it took Segregated witness three years to reach a 50% adoption rate. Thus, migration to Taproot will be a long process. It will not happen until there are sufficient smart contract transactions that make full use of taproot features. All these upgrades degrade the privacy of bitcoin users.

5. Payment Protocol

Evidence shows that research has been done on payment protocol in BIP70 till BIP75. In a regular bitcoin transaction over the web, buyers will add their shopping item to a list and check out. Upon checking out, the merchant will provide the buyer with a unique bitcoin address to pay. The buyer uses their bitcoin wallet to pay to the provided address. Upon transaction confirmation in the bitcoin p2p network, the merchant will send out the goods to their customer. BIP70 added five new features into a regular bitcoin transaction. Instead of asking the customer to pay to a 34-character bitcoin address, a human-readable secure payment destination such as "www.store.com" is given. This payment can serve as a secure proof of payment, and it is resistant to man-in-the-middle attack whereupon authorization the merchant's bitcoin address cannot be changed. Customers can know immediately whether the merchant has received the fund. In odd cases such as overpayment or partial fulfillment, the customer refund addresses are automatically made available to the merchant.

BIP71, enable payment protocol message to be sent via standard email or HTTP protocol. BIP72 enable a user to click a link in website or email to initiate a bitcoin payment. An "r" parameter is added to the bitcoin payment message. Wallet software that supports BIP72, the "r" parameter can issue a payment message shown below without specifying the bitcoin address.

bitcoin:

?r=https://seller1.com/pay.php?h%3D2a8628fc2faa

Wallet software that does not support BIP72 will not be able to decode the above message. Thus, it is recommended to use a backward compatible message shown below. Non BIP72 wallet will ignore the "r" parameter in the given backward compatible message below.

bitcoin:

1N81wJJL4oPQk6XSeddY8VDNKxysPoDvb7?amount=0.21&r=https://seller1.com/pay.php?h%3D2a8628fc2faa

In some bitcoin payment, there are many options such as

bitcoin: 1N81wJJL4oPQk6XSeddY8VDNKxysPoDvb7
?amount=0.20

&label=Example+Seller1

&message=Order+Book+Pens+%26+Pencil

&r=https://seller1.com/pay.php?h%1234a567

Thus, the QR code to be scanned will be very dense and may be difficult to scan, leading to customer frustration, as

 | DOI: 10.20533/ijisr.2042.4639.2021.0112

shown in Table 5. A more straightforward payment option should be

bitcoin: ?r=https://seller1.com/pay.php?h%1234a567

For compatibility reasons, merchant should provide all available options for their customer to secure the payment.

Table 5: Different QR code for the same payment option

QR code with URL only

QR code with bitcoin payment option and URL

BIP74 was rejected as a standard. Bitcoin users may want to keep a list of known payees to send payment without getting their payees' latest bitcoin address. However, for privacy reasons, each bitcoin address is used only once. In BIP-32, public users can know the latest payee public key by monitoring payees' X-pub chain. In normal child key, the child public key can be derived from the parent public key. Users need to know the index "i" th to derive the child public key. The bitcoin address can be determined from the public key. However, there is no way to know whether the payee has lost access to their parent's private key resulting in funds sent to an inaccessible address. In addition to that, normal child key derivative is less secure. If the child's private key and the parent extended key are compromised, the parent's private key is compromised.

In BIP75, users who wish to make a payment can communicate with their payee using HTTP or email. The wallet software will perform all the communication in the background. The payee will respond with their latest bitcoin address. If there is no response from the payee, sending funds to dead addresses is prevented. For the payee to respond with their latest bitcoin address, they must know the sender/requestor's identity. In BIP75, the sender identity is sent along with the request for the payee's new bitcoin address. The receiving payee can decide the sender's validity and whether to respond to the request. These bitcoin payment protocol standards were completed in Nov 2015; however, market adoption is still in question. As bitcoin price reached more than USD50K per bitcoin in 2021, it is unsure whether it will be adopted as a popular payment protocol due to excessive fees.

6. Conclusion

In this paper, we have described bitcoin's evolution in dealing with scaling requirements, migration to Segregated witness, and features to support payment protocol over the Internet. Most of these requirements are inter-related to the bitcoin address. We hope that readers can gain an easy step-by-step understanding of these issues.

7. References

[1] Github. Bitcoin. https://github.com/bitcoin/bips/blob/master/README.mediawiki (Access Date: 18 June 2020).

[2] En.bitcoin. Wiki Main Page. https://en.bitcoin.it/wiki/Main_Page (Access Date: 16 Juky 2020).

[3] W.K. Chan, J.J. Chin, V.T. Goh. (2020). "Evolution of Bitcoin Addresses from Security Perspectives", International Conference for Internet Technology and Secured Transactions (ICITST-2020), December.

[4] Maxwell, G., Poelstra, A., Seurin, Y., Wuille, P. (2018). Simple Schnorr multi-signatures with applications to bitcoin. Cryptology ePrint Archive, Report 2018/068.

[5] G. Maxwell. (2013). "CoinJoin: Bitcoin privacy for the real world," https://bitcointalk.org/index.php?topic=279249.0 (Access Date: 3 October 2020).

[6] Manu Drijvers, Kasra Edalatnejad, Bryan Ford, Eike Kiltz, Julian Loss, Gregory Neven and Igors Stepanovs. (2019). On the security of two-round multisignatures. In 40th IEEE Symposium on Security and Privacy (SP), May.

[7] Bitcointechtalk. What is a bitcoin merklized abstract syntax tree mast. https://bitcointechtalk.com/what-is-a-bitcoin-merklized-abstract-syntax-tree-mast-33fdf2da5e2f (Access Date: 23 September 2020).

[8] Naik Manali Rubin, Jerry and Nitya Subramanian. (2016). Merkelized Abstract Syntax Trees. http://www.mit.edu/~jlrubin/public/pdfs/858 report.pdf (Access Date: 11 December 2020).

[9] Bitstamp. Bitstamp proof of reserves statement. https://www.bitstamp .net/s/documents/Bitstamp_proof_of_reserves_statement.pdf (Access Da te: 27 January 2021).

[10] Coindesk. Krakens audit proves holds 100 bitcoins reser https://www.coindesk.com/krakens-audit-proves-holds-100-bitcoins-rese rve (Access Date: 6 March 2021).

[11] Kraken. Proof of reserves audit. https://www.kraken.com/proof-of-reserves-audit (Access Date: 21 March 2021).

[12] Jimmy Nguyen. (2017). The Risks of Segregated Witness: Problems under Evidence Laws, The Computer and Internet Lawyer, VOl 34, no 11, November.

[13] Nikita Zhavoronkov, "Negative Impact Of TapRoot On Bitcoin's Privacy. Based On The Experience With SegWit", https://t.co/5LK1QZSUzO ?amp=1. Nov 2020 twitter tweat. (Access Date: 5 April 2020).

[14] Blockchair. Bitcoin. https://blockchair.com/bitcoin/charts/segwit-usage (Access Date: 14 May 2021).

 | DOI: 10.20533/ijisr.2042.4639.2021.0112

Acknowledgement

The authors are grateful for the financial assistance through the Ministry of Education of Malaysia in supporting this work through the Fundamental Research Grant Scheme (FRGS/1/2019/ICT04/MMU/02/5). The 2nd author would like to thank the Information Security Lab at MIMOS Berhad, which hosted his visit during which this paper was written.

 | DOI: 10.20533/ijisr.2042.4639.2021.0112