Bitcoin Vault : Peer-to-Peer Anti-Theft Electronic Gold

Eyal Avramovich
github.com/bitcoinvault

Abstract. Satoshi Nakamoto's original vision for Bitcoin was to create a peer-to-peer version of electronic cash. The majority of successful forks of the protocol try to improve on this vision further and provide a more scalable and efficient system for payments. We see Bitcoin's greatest promise not as a medium of exchange but as a store of value - a better form of gold, not cash. We propose a set of modifications to the original protocol aimed at fulfilling this promise by creating the ultimate electronic store of value. By increasing Bitcoin's effective transaction confirmation time of 10 minutes to 24 hours, we are able to tackle Bitcoin's greatest flaw as a form of gold - susceptibility to theft. A system geared less towards paying for coffee and more towards holding one's life savings with complete peace of mind, where every transaction is alerted on-chain for 144 blocks and can be canceled on emergency with a recovery key that was never used prior and hence invulnerable. The bootstrap of this system is not via hard fork but through a fairer mechanism of expedited mining, allowing the system to catch up to Bitcoin in a short period of time. We would like to give all credit to the creator of the Bitcoin Royale whitepaper [1] which created the vision of no-theft electronic gold. As a team, we would like to move forward with that idea and start implementing all the crucial features mentioned below and make it even more advanced with 3rd private key solution that is revealed as an additional feature in paragraph 4.1

1. Introduction

Bitcoin [2] is an innovative decentralized payment system launched in 2009 allowing parties to transact directly without going through a trusted financial institution. The system relies on proof-of-work to maintain a distributed ledger without a trusted operator, that is secure as long as honest nodes control more CPU power than any cooperating group of attacker nodes. Bitcoin was originally described by its creator Satoshi Nakamoto as an "electronic cash system".

Multiple successful modifications of the original protocol have been released over the years in the form of forks of the Bitcoin codebase. These include Litecoin [3] that launched in 2011 to reduce transaction confirmation time and change the proof-of-work algorithm to favor consumer-grade hardware such as GPU; Bitcoin Cash [4] that launched in 2017 to scale the original protocol's transaction throughput by increasing block size; and Bitcoin Gold [5] that also launched in 2017 to render specialized mining equipment obsolete by changing the hashing algorithm.

True to the original vision, the primary focus in these forks and others is to make Bitcoin a better system of *cash*. Limitations of the original protocol such as high transaction fees, 10 minute confirmation times and approximate throughput of only 4 transactions per second hinder Bitcoin's ability to compete with the centralized online payment systems dominant today.

2. Electronic Cash or Electronic Gold

Whereas Bitcoin did not see much success with consumer adoption as electronic *cash*, it has been significantly more successful as a form of electronic *gold*. There is a long standing industry debate whether Bitcoin is superior as a medium of exchange or in fact as a store of value. Gold is not an effective means of payment for day-to-day goods and services. Consumers primarily invest in gold to hedge against inflation and preserve future purchasing power. Unlike national currencies, Bitcoin's fixed monetary policy and limited supply make it particularly attractive in this regard.

History shows that systems can rarely be designed to meet several competing goals at once. Optimizing Bitcoin to become a better medium of exchange diminishes its potential as a store of value. On the same note, by sacrificing further on the properties required for useful electronic cash, we can vastly improve its utility as electronic gold. In this paper, we propose a series of modifications to the original Bitcoin protocol focused on a single goal - creating the ultimate store of value.

If we no longer prioritize competing as an online payment system, we need not focus on transaction fees or transaction throughput. After all, gold is expensive to transport and is normally acquired for long term investment. A property that is particularly relevant to our efforts is transaction confirmation time. Tradeoffs on this front, such as substantially increasing Bitcoin's average 10 minute confirmation time, can yield cardinal advantages. Since we would not expect to freight a shipment of gold across locations in under 10 minutes anyways, this sacrifice seems natural.

3. The Problem of Theft

The key requirement from a store of value is to be nonperishable. Theft is one of the primary risks of loss when dealing with anything of value. Gold performs rather well in this regard. Physical theft of gold is significantly riskier to execute than any virtual attack on an electronic asset. In addition, a thief would have a hard time hiding a sizable cargo of stolen gold from the authorities, especially across borders. Laundering and liquidating stolen gold in large quantities while remaining anonymous is no simple task either.

Unfortunately, Bitcoin fares poorly by comparison. Funds are protected by the protocol with sets of cryptographic private keys. Gaining electronic access to these keys allows an attacker to seize all funds remotely, immediately and irrevocably. Laundering stolen funds is also significantly easier since transactions are pseudonymous, traceability can be disrupted by use of mixers [6] and Sybil identities can be created in bulk. As a result, cryptocurrency theft is on the rise with over $1 billion stolen in 2018 [7]. Community-curated lists of major incidents [8] show that even professional institutions well-versed in the latest security practices are prone to attack.

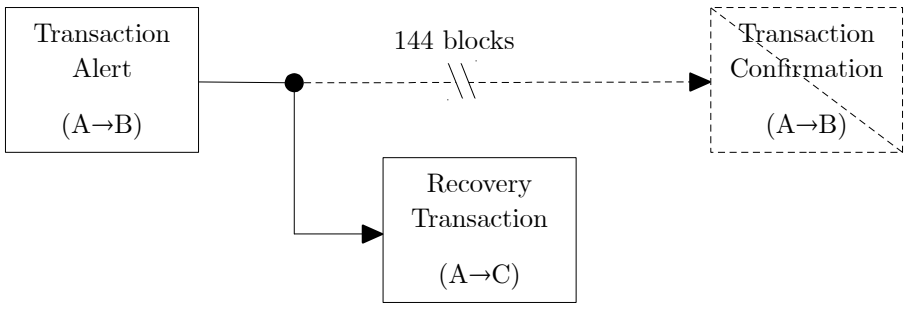
Secure management of private keys by end-users is proving to be one of the major challenges of Bitcoin. Since keys must be regularly used to interact with funds and the signed transactions must be transmitted over the Internet, every key eventually becomes vulnerable. Security-conscious practices like splitting funds between "hot" and "cold" wallets are cumbersome and fail to solve the problem at the root. Incidents show that hot wallets unavoidably hold significant amounts [9, 10] and use of cold wallets only reduces interaction with keys but does not eliminate it completely [11].

4. Anti-Theft Solution

As a means to eliminate theft, we propose to delay confirmation of all transactions in the system by 24 hours. When a miner adds a transaction to a new block, the transaction will no longer be committed immediately. Instead, it will be added on-chain as an *alert* for the duration of 144 blocks (assuming block time of 10 minutes) . If left undisturbed for 144 blocks, the transaction will change from alert state to *confirmed*.

The benefit of on-chain alerts is that coin owners will receive uncensorable advance notifications whenever their coins are moved. The owner will have 24 hours to act upon the alert and will be able to override the transfer if unauthorized. Delayed withdrawals are a proven anti-theft industry standard in custodial wallets and the traditional banking system. Coinbase Vault [12] for example has a wait time of 48 hours. Our proposed solution provides the same protection without a trusted third party.

Emergency override of a transaction in alert state requires a special *recovery transaction* and is carried out immediately without being subject to the standard 24 hour delay. The recovery transaction requires a special *recovery key* that must be registered in advance by the wallet owner using a special *registration transaction*. Once a recovery key has been registered, it cannot be changed.



The private recovery key is intended to be a fresh key that has never been used before the emergency override. This key can be generated offline and should never be connected to the Internet or inputed to any device. The registration process only requires the public part of this key, ensuring that the private key can remain unused. Unlike cold wallet keys that must be used occasionally and are thus vulnerable to theft, the recovery key will be used for the first time only in the emergency override itself, and therefore can be completely theft resistant. Once the recovery key has been used, it is to be considered compromised and all funds should be immediately transferred to a new wallet protected by a new unused recovery key.

Delaying every transaction in the system by 24 hours may be viewed as an extreme measure that will hinder usability. In our mission to create the ultimate store of value, we are gladly willing to trade off a less attractive form of cash with a very attractive form of gold that cannot be stolen. This tradeoff gives a strong foundation of a slow moving and secure base layer, which can still be accelerated in upper layers like the Bitcoin Lightning Network [13].

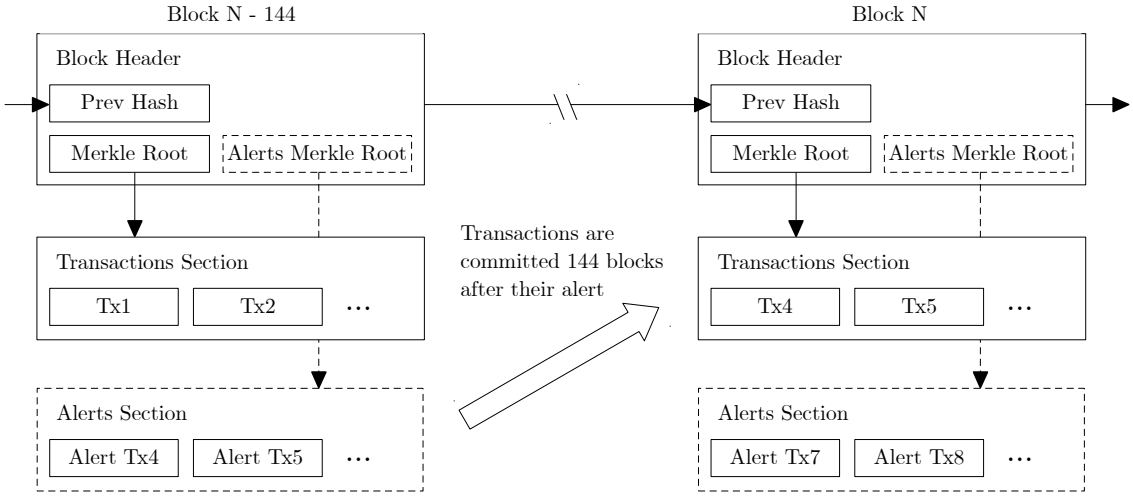
4.1 3rd Private key

We would like to add an additional private key (3rd one) which will decrease the required 144 confirmations to just 1. This solution speeds up the transaction time to around 10 minutes from the required 24 hours. Users who keep their 3rd key in a seperate location from the other 2 will have a greater level of security, detering theft.

5. Blocks

Like the original Bitcoin protocol, every block contains a list of confirmed transactions and holds the Merkle root hash of these transactions in its header. Since regular transactions must wait 24 hours on-chain, they cannot be added immediately to a block's transactions section. Instead, they are added to a new special section that does not exist in the original Bitcoin protocol - the *alerts section*. The Merkle root hash for the alerts section is also stored in the block header.

When a new block is mined, the miner looks back 144 blocks and examines the alert section of block N-144. All alerts for transactions that are still valid become confirmed and populate the transactions section of the new block.



The steps for mining block N are as follows:

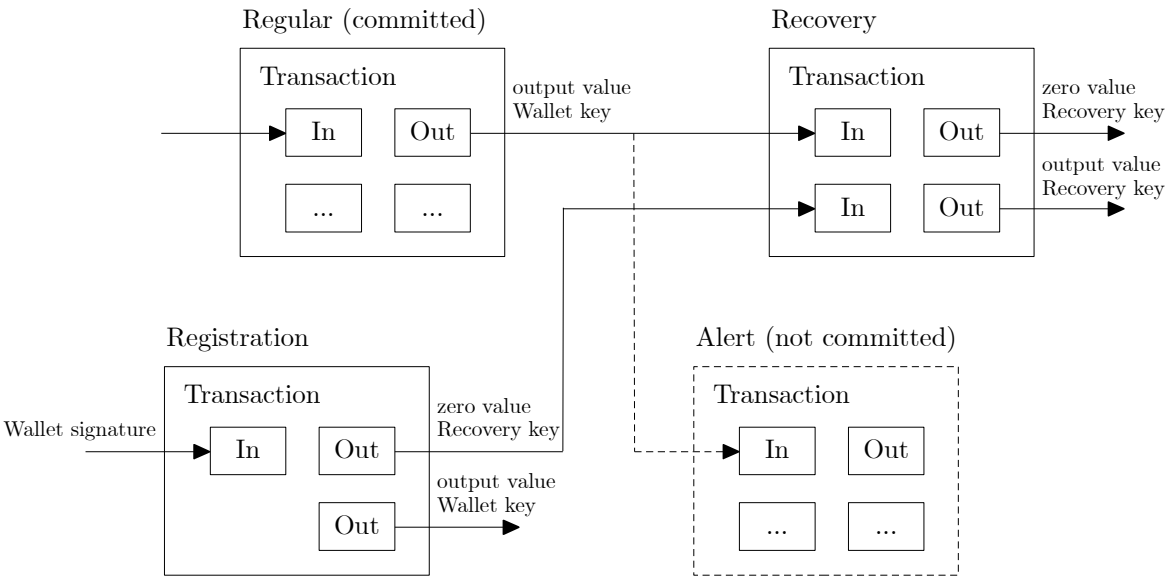
1) Add new regular transactions to block N alerts section (not confirmed).
2) Add new registration transactions to block N transactions section (confirmed).
3) Add new recovery transactions to the block N transactions section (confirmed).
4) Go over block N-144 alerts section and add the valid transactions to block N transactions section (confirmed).

6. Transactions

Coin transfers are performed using regular transactions identical to those in the Bitcoin protocol. Their format does not change as they're moved from the alert section to the transactions section.

A *registration transaction* has a single input which specifies the wallet owner's key. The recovery key is specified in the first output, which must have a value of zero. An additional output is allowed with the same wallet's key. When processed, the registration transaction creates a zero-value UTXO intended to be spent on the first recovery. The wallet's key and the recovery key may be replaced with any general script such as multisig.

A *recovery transaction* has two inputs. The first is the UTXO being recovered, which must belong to the wallet's key. The second is the zero-value UTXO belonging to the recovery key. The first output must specify the recovery key and have a value of zero. Additional outputs are allowed. When processed, the recovery transaction creates a new zero-value UTXO intended to be spent on the next recovery. When a recovery transaction is processed, the recovered UTXO is spent, thus invalidating any existing alerts relying on this UTXO and preventing them from becoming confirmed post their 24 hour delay.



7. Compatibility

Our design attempts to make the protocol as compatible as possible with the standard Bitcoin protocol. The underlying goal is to minimize necessary code changes to existing Bitcoin full nodes, clients and wallets.

The alerts section in blocks is the primary deviation from the original protocol and must be constructed in a backwards compatible way. In order to avoid changing the format of the block header, the Merkle root hash for the alerts section is stored in the input of the standard *coinbase transaction*. Existing Bitcoin clients that are unaware of its existence will ignore the section and all of its alerts. Nevertheless, they will still display transaction confirmation correctly since confirmed transactions are added to blocks by miners in the standard way after 24 hours.

8. Incentive

By large, the system borrows the incentive model of Bitcoin that has proven itself successful over the last decade. The incorporation of alerts requires minor modifications to this model. Miners are incentivized to add new transactions as alerts to a block by the transaction fee paid in these transactions. This means the alerts section must begin with a separate *alerts coinbase transaction* generated by the miner of the block to distribute transaction fees from all transactions found in the alerts section. Naturally, these fees are not distributed immediately but must wait 144 blocks for the future miner creating the block that confirms them and adds them to its transactions section.

The future miner will not receive the fees for confirmed transactions moved from the past alerts section, but will still be given the block reward for mining a new block and all fees from new transactions that are confirmed immediately (like registration transactions and recovery transactions). When the future miner creates the standard coinbase transaction for the block, the outputs of the past alerts coinbase transaction must be added to it to finalize the fee distribution. This mechanism implies that any recovery transaction that is invalidating a past alert must compensate for the transaction fee in this alert.

A potential attack we need to mitigate is theft combined with miner bibery through fees. Consider an attacker that has gained control of a wallet's private key and creates an unauthorized transaction alert that pays significant part of the wallet balance as fee. Since any recovery transaction must compensate miners for the fee, recouping all stolen funds will not be possible. We propose to limit the maximum fee per transaction in the protocol level. This behavior is already common in wallets, Bitcoin Core included, to protect users from accidents.

9. Bootstrap

The Bitcoin protocol introduces new coins to circulation gradually through block rewards. In the early days of the protocol, this behavior creates a very strong inflation in circulating supply, which subsides over time. Since scarcity of supply is paramount for a store of value, if started anew, the system will not meet its desired goal effectively for a long time. Bitcoin, on the other hand, has already reached a mature point in this regard. By the end of 2020 Bitcoin's inflation will be under 1.8%, comparable to 1.6% of gold [14]. We propose to catch up the system to the same stage as Bitcoin to enjoy the same level of maturity. This will also assist the market in assessing the value of this new asset by simplifying the comparison to Bitcoin.

A common method of catching up is relying on the original genesis block and hard forking the Bitcoin ledger on launch. This will import all existing balances from Bitcoin in a 1:1 ratio. We believe this method is not ideal for a store of value, since the majority of supply will be given to individuals who have exerted zero effort in attaining it.

As a fairer alternative, we propose to launch from a fresh genesis block with a short bootstrap period of expedited mining. This period should last at most one year, possibly shortened to meet a symbolic event like the next Bitcoin halving. During this bootstrap period, block rewards for miners will be significantly higher than usual, in order to match Bitcoin's circulating supply at the end of the period. Once the expedited mining period is over, the system will continue with the original protocol's schedule of block rewards in parallel to Bitcoin. An added benefit of this approach is aggressively attracting miners on launch, so by the end of the bootstrap, when the system is ready for use as a store of value, significant hash power will already secure the network.

Expressing this behavior in C++ code:

```
typedef int64_t CAmount; // taken from Bitcoin Core (amount.h)
static const CAmount COIN = 100000000;

static const int BOOTSTRAP_PERIOD = 52500; // in blocks (here 1 year)
static const int BITCOIN_HEAD_START = 12 * 52500; // in blocks (here 12 years)
static const CAmount EXPEDITED_PERIOD_SUBSIDY = ExpeditedPeriodSubsidy();

CAmount GetBlockSubsidy(int nHeight)
{
    if (nHeight < BOOTSTRAP_PERIOD) // expedited mining
        return EXPEDITED_PERIOD_SUBSIDY;
    return BitcoinGetBlockSubsidy(nHeight +
                                  BITCOIN_HEAD_START -
                                  BOOTSTRAP_PERIOD);
}
```

```
CAmount ExpeditedPeriodSubsidy()
{
    CAmount nTotalSubsidy = 0;
    for (int nHeight = 0; nHeight < BITCOIN_HEAD_START; nHeight++)
        nTotalSubsidy += BitcoinGetBlockSubsidy(nHeight);
    return nTotalSubsidy / BOOTSTRAP_PERIOD;
}

// taken from Bitcoin Core (validation.cpp)
CAmount BitcoinGetBlockSubsidy(int nHeight)
{
    int halvings = nHeight / 210000;
    if (halvings >= 64)
        return 0;
    CAmount nSubsidy = 50 * COIN;
    nSubsidy >>= halvings;
    return nSubsidy;
}
```

10. Related Work

We are particularly impressed with the work of Malte Möser et al. on Bitcoin Covenants [15]. Their extension of the Bitcoin script language enables restrictions on the future use of coins, which can be used to implement a variety of security measures. The primary of which is *vault transactions*, which resemble our proposed delayed withdrawals mechanism.

We have opted for a simpler implementation that does not require a recursive array of custom scripts to be implemented by wallets. Our proposal is more than an optional extension, it is a fundamental change to the protocol with a wide mandatory effect on all transactions. Shifting the burden of implementation to miners and leaving the end-user transaction experience identical to the default, enable us to better carry out our mission of creating true electronic gold.

References

[1] Ian Duoteli Fleming, **https://bitcoinroyale.org/bitcoinroyale.pdf**

[2] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", https://www.bitcoin.org/bitcoin.pdf, 2008.

[3] Litecoin Project, "Litecoin, open source P2P digital currency", https://litecoin.org, 2014.

[4] "Bitcoin Cash", https://www.bitcoincash.org, 2018.

[5] bitcoingold.org, "Bitcoin Gold", https://bitcoingold.org, 2018.

[6] J. H. Ziegeldorf, F. Grossmann, M. Henze, N. Inden, and K. Wehrle, "Coinparty: Secure multi-party mixing of bitcoins", In: *CODASPY '15*, 2015.

[7] CipherTrace, "Cryptocurrency Anti-Money Laundering Report, 2018 Q4", https://ciphertrace.com/crypto-aml-report-2018q4, 2019.

[8] "BLOCKCHAIN GRAVEYARD", https://magoo.github.io/Blockchain-Graveyard, 2019.

[9] C. Zhao, "Binance Security Breach Update (May 7 2019)", https://binance.zendesk.com/hc/en-us/articles/360028031711, 2019.

[10] J. Buck, "Coincheck: Stolen $534M ln NEM Were Stored On Low Security Hot Wallet", https://cointelegraph.com/news/coincheck-stolen-534-mln-nem-were-stored-on-low-security-hot-wallet, 2018.

[11] J. Preissler, "Important Notice: Only trade TIO on trade.io", https://medium.com/@trade.io/important-notice-only-trade-tio-on-trade-io-823d59fd7104, 2018.

[12] KM. Cutler, "Coinbase Launches A More Secure Bitcoin Storage Option Called 'Vault'", https://techcrunch.com/2014/07/02/coinbase-vault, 2014.

[13] J. Poon and T. Dryja, "The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments", https://lightning.network/lightning-network-paper.pdf (draft), 2016.

[14] World Gold Council, "How much gold has been mined?", https://www.gold.org/about-gold/gold-supply/gold-mining/how-much-gold, 2018.

[15] M. Möser, I. Eyal, E. Gün Sirer, "Bitcoin Covenants", In: *Financial Cryptography and Data Security, FC 2016, Lecture Notes in Computer Science*, Vol. 9604. Springer, Berlin, Heidelberg.