# ChainSpl.it

## Easily handle forked coins

Non-custodial APIs so wallets and exchanges
don’t need to write code for every new fork and chain split.

## Background

The Bitcoin blockchain contains coins. New coins on the Bitcoin blockchain are created by Bitcoin miners. Users can buy the rights to coins through an exchange service. Users can transfer their rights to Bitcoin coins (or part thereof) via Bitcoin transactions. Digital wallets allow users to execute transactions on the Bitcoin blockchain. Each digital wallet contains a user’s private keys and is therefore a mechanism which allows users to secure their rights, as well as transfer their rights to other users via transactions.

The Bitcoin blockchain computer code is open-source (it can be legally copied and modified by a third party). Some incompatible changes to the Bitcoin software are also known as hard forks i.e. changes which modify the rules used to validate transactions and blocks. The launch of any incompatible modifications to the Bitcoin blockchain software will cause different nodes to no longer agree on transaction history. When this happens Bitcoin itself is not really affected. However, at the time when these changes take effect, the Bitcoin blockchain does split. From that point onwards two blockchains exist. It is important to remember that the blockchain contains every historical transaction and therefore when the blockchain splits each blockchain contains an identical history, right up until the chain split.

Due to this shared transaction history, a user who holds rights to coins on the Bitcoin blockchain, before a blockchain split, automatically holds the identical rights to coins on the other chain, at the time of the blockchain split.

If neither of the chains are sporting a new and unique transaction mark, a problem arises, whereby a transaction on one chain is valid on the other chain. In this case transactions on one chain can be replayed on the other chain. A replay is not automatic, nor is it in any way part of the default operation of either chain. A replay attack can be carried out by a malicious party. The point being, that whilst a signed transaction on one chain can be repeated on the other chain, the coins are essentially stuck together.

Let's investigate this a little further in the following section which introduces Unconfirmed Transaction Outputs (UTXOs).

## Unspent Transaction Outputs (UTXOs)

Your Bitcoin wallet handles spending a little differently than your wallet in the traditional sense. There are some key concepts and terminologies to grasp. Let’s start with the Bitcoin transaction. A transaction transfers your rights to coins on a single blockchain.

All transactions contain at least one input. An input in the transaction, which you are creating, is an output from a previous transaction (except for in the case of mining, but that’s another story). These coin rights which you previously received can be signed by your wallet and sent to another Bitcoin user as part of a new transaction. But until you actually create and execute this transaction you have rights to Bitcoin coins in the form of Unspent Transaction Outputs (UTXOs).

As mentioned above when the Bitcoin blockchain splits, the UTXOs (coin rights) which you have on the original chain also exist on the newly forked chain.

This is where chainspl.it comes in.

Chainspl.it is a service which helps users safely split UTXOs across forks so that users are free to spend their split (uniquelly identifiable) coin rights separately, on each of the two blockchains.

Chainspl.it is non custodial, meaning all of the signing takes place on the user’s end.

For more information please see our "Introducing chainspl.it, Easily handle forked coins" YouTube video [1]

[1] https://www.youtube.com/watch?v=wWkdfnqjpug&feature=youtu.be&a=

Chainspl.it provides five APIs to make life easy for wallets and exchanges.

## Exclusivity  API

This API accepts a small payment and a Bitcoin address. Once the payment is confirmed it sends a batch of 100 private keys. Each private key has a 1,000 satoshi balance that are exclusive (only valid on one chain). By including a small amount of exclusive  Bitcoin in a transaction the entire transaction is only valid on the Bitcoin network (for a discussion of how the exclusive  Bitcoin is created see *How we generate Bitcoin only transactions* below). This splits your coins because the source address is still full on the fork network, but is emptied on the bitcoin network. 1,000 satoshis (about $.06) because smaller amounts will not be relayed by nodes as it is considered “dust.”

The API provides a batch of 1000-satoshi private keys for two reasons. First, if a larger exclusively unique  balance was provided to the exchange or wallet, all transactions sent by that wallet or exchange could be identified because they would include a satoshi from that issued exclusively unique  balance. Second, by providing the private keys directly, we avoid an additional network transaction.

## Ignore For Now API

When queried, this API responds with the ratio of UTXO size to UTXO balance that is currently economical to split. For example if the API returns 10 millibits per byte UTXOs that have a 11 millibits per byte should be split now, but UTXOs that have 9 millibits per byte should remain unsplit until the value of the forked coin increases or the network fees decrease.
This ratio is calculated by evaluating the current market price for the forked coin and the current network fees that will be required to split the coins.

## Delayed Relay API

This API accepts batches of Bitcoin transactions and trickles them out to the network over a 12 hour period. This improves privacy when splitting coins because an attacker can infer ownership, even if UTXOs are not combined, by looking at the timing of transactions. If 50 transactions are split in the same block (using our exclusive  coins) it could provide strong evidence that they were owned by the same person. By spacing transactions this becomes more difficult (especially if there are a large number of people using our exclusive  UTXOs in that period of time).

## Delayed Sell API

This API is similar to the Delay Relay API except instead of trickling out the split transaction, this API trickles out the sell orders. If you sell a large number of forked coins in the same transaction that would indicate the same owner. Those coins can be traced back to their pre-forked Bitcoins and back to the owner. By delaying the sell orders this API makes that much more difficult.

## Split Transaction API

This API accepts a UTXO and a destination address and it returns a transaction that spends the UTXO and exclusively unique UTXO (that it pulls in the background from the Exclusivity API). The client can then sign the transaction with the private key associated with the legacy UTXO without knowing how to generate these transactions or storing the exclusively unique UTXO.

# How we generate bitcoin only transactions

We intend to use a variety of techniques to generate coins that can be used to split transactions. And, where possible we intend to combine techniques, to create “defense-in-depth.” We will also be careful to only use techniques that make sense for a given hardfork.

## New Coinbase Coins

Bitcoin that is generated by mining after a fork are sent to an address that only exists on one of the forks. Any transactions that include these coins will be rejected by the side of the fork that doesn’t contain these brand new addresses with balances.
Pros: Simple and relatively secure.
Cons: Requires that the new coins mature 100 blocks—forces a delay in splitting.

## nLockTime Transactions

By using nLockTime we can generate a transaction that will be immediately valid on one chain and won’t be valid for 100 blocks on the other chain (assuming the chains have diverged more than 100 blocks). This allows us to send a transaction that is fully confirmed on one chain. On the other chain we can spend the balance to zero before nLockTime expires and therefore make that transaction invalid.
Pros: Simple and pretty secure (though a little less than new coinbase coins).
Cons: Need to wait until the chains diverge enough to provide time to invalidate one of the transactions before the nLockTime expires.

## Massive Transaction

If one chain supports larger blocks than the other, a really big transaction (greater than 1MB w/o SegWit) can be created. It would be invalid on the original chain, but would be valid on the new chain.
Pros: Can be executed immediately after the fork. No delay “technically required.”
Cons: Requires participation of a miner to accept the larger than 100 kb transaction. Also involves significant mining expense. The expense could be spread across multiple users, but the cost is still significant.

## High/Low Fees

If one chain has a big enough divergence in fees a transaction could be created that would have a high enough fee on one chain to be confirmed, but low enough to be ignored on the other chain. Once the first chain confirms the transaction a new transaction could be created to invalidate the transaction on the high fee chain.
Pros: Cheap. Should be relatively easy. No delay required.
Cons: May require trial and error. A miner could mine the low fee transaction at a loss to prevent the chain split, which would require you to try again.

## Reasoning from first principles (known truths)

As Aristotle accounts, reasoning from the first basis of which a thing is known. Reasoning from first principles (or known truths), provides greater flexibility for immediate and efficient action. Put simply by American efficiency engineer and business theorist Harrington Emerson, “the man who grasps principles can successfully select his own methods. The man who tries methods, ignoring principles, is sure to have trouble”.

The list directly below this paragraph contains some known truths about the Bitcoin blockchain, hard forks and chain splits. These known truths can be used/referenced during design, development, marketing, interviews, announcements and more.

***It is essential that this information is vetted by experts as soon as possible.***

- Chainsplit is a service which helps users safely split Unspent Transaction Outputs (UTXOs) across forks
- Chainsplit is non custodial, meaning all of the signing takes place on the user’s end
- A transaction valid on both the original chain and the forked chain (after a chain split) is prone to a replay attack
- A replay attack can only be performed when two chains are able to accept identical transactions
- A single transaction can not transfer rights to coins between two chains i.e. the original chain can not directly transfer rights to coins on the forked chain
- An identical (duplicate) transaction can however exist on both chains, and in doing so is mirroring the transferal of coin rights on both chains
- In the event of a chain fork, a replay attack is whereby a public transaction (performed on one chain - i.e. the transferal of coin rights) is performed on the other chain
- If the original Bitcoin chain implemented strong replay protection after a 3rd party initiated a chain split, all original Bitcoin users (all original Bitcoin nodes) would have to update their software
- All transactions contain at least one input and at least one output
- All transactions can contain many inputs and/or many outputs
- A transaction transfers rights to coins on a single blockchain
- The input section of a regular transaction refers to its parent transaction
- A “generation transaction” also known as a “coinbase transaction”, has no parent transaction

Chainsplit is a service which helps users safely split Unspent Transaction Outputs (UTXOs) across forks

Chainsplit is non custodial, meaning all of the signing takes place on the user’s end

A transaction valid on both the original chain and the forked chain (after a chain split) is prone to a replay attack

A replay attack can only be performed when two chains are able to accept identical transactions

A single transaction can not transfer rights to coins between two chains i.e. the original chain can not directly transfer rights to coins on the forked chain

An identical (duplicate) transaction can however exist on both chains, and in doing so is mirroring the transferal of coin rights on both chains

In the event of a chain fork, a replay attack is whereby a public transaction (performed on one chain - i.e. the transferal of coin rights) is performed on the other chain

If the original Bitcoin chain implemented strong replay protection after a 3rd party initiated a chain split, all original Bitcoin users (all original Bitcoin nodes) would have to update their software

All transactions contain at least one input and at least one output

All transactions can contain many inputs and/or many outputs

A transaction transfers rights to coins on a single blockchain

The input section of a regular transaction refers to its parent transaction

A “generation transaction” also known as a “coinbase transaction”, has no parent transaction

- A miner creates the “generation transaction”/”coinbase transaction” as the first transaction in a block
- Miners choose which transactions they include in a block
- By default, miners will include transactions with acceptable transaction fees
- A special field called the “coinbase parameter”/”coinbase field” (not to be confused with the coinbase transaction or the exchange company Coinbase) is used as the sole input for a “generation transaction”/”coinbase transaction”
- After a chain is forked, all subsequent transactions of a “generation transaction” will only be valid on the particular chain which created the “generation transaction”, and will be rejected by the other chain
- “Generation transactions” can only be spent once they receive 100 confirmations
- It's possible to deliberately create a transaction which is greater than 1MB
- A block which is greater than 1MB (excluding SegWit transactions) would-be/is invalid on the original Bitcoin chain
- Segregated Witness is backwards compatible with all previous Bitcoin software
- Segregated Witness is currently active on the Bitcoin Core blockchain (activated on the 24th of August 2017)

## Assumptions

The above first principles are essentially a way for us to avoid reasoning from assumptions. Satoshi’s Bitcoin blockchain was a paradigm shift. With the leading minds here, can we be hopeful for a paradigm shift from ChainSpl.it? It is said that one must fundamentally release themselves from the traditionally taken approach as well as any underlying assumptions in order for a paradigm shift to occur. The list directly below this paragraph contains some assumptions; beliefs that may be based on our past experience and opinions but not necessary true for everyone. It is just as important to define assumptions because a solution which is based (or partly based) on assumptions may a) divide the team,  b) lead to a delay in implementing the solution or c) negate the solution.

- Bitcoin Core does not provide chain split awareness
- There is no built-in/standard feature in Bitcoin Core to discern “Bitcoin only” (original Bitcoin transactions) from transactions on the forked blockchain, after a chain split
- It is not practicable for Bitcoin Core to implement replay protection every time the Bitcoin Core blockchain is split by a third-party
- The onus on implementing replay protection should be on the third-party responsible for splitting the chain
- After a split of the Bitcoin blockchain occurs, the block height of the two (split) chains will diverge more than 100 blocks
- We should not compromise users privacy by default
- The probability of a block being mined in 10 minutes is 63%
- The probability of a block being mined in 30 minutes is 95%
- The probability of a block being mined in 60 minutes is 99.7%