Chainpoint

A scalable protocol for recording data in the blockchain and generating blockchain receipts

Authors: Wayne Vaughan, Shawn Wilkinson, Jason Bukowski

July 22, 2015
v1.0

Abstract

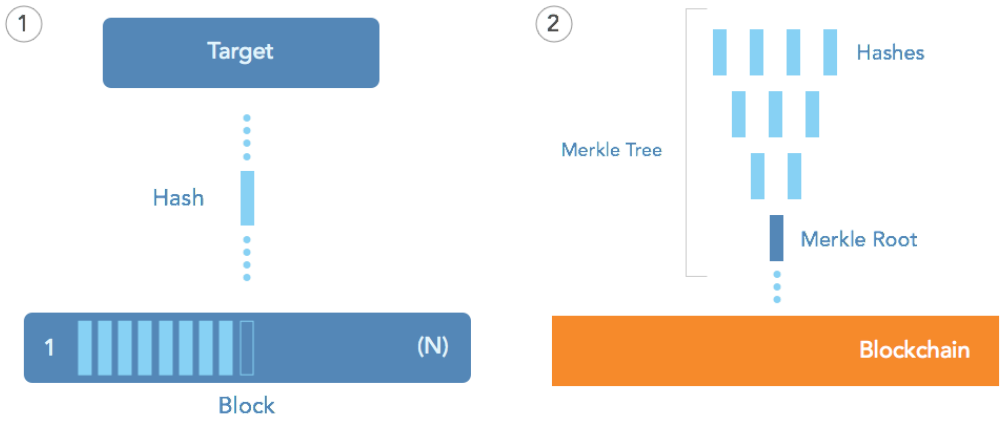
A standard for maximizing the scalability of recording data in the blockchain and generating blockchain receipts. Each receipt contains all the information needed to verify the data without relying on a trusted third party.

Introduction

The use of the Bitcoin blockchain [1] to timestamp and verify data in an immutable public ledger was pioneered by Manuel Aráoz with the creation of Proof of Existence [2]. This system, and others like it, notarize data in the blockchain by publishing a hash of the data in a Bitcoin transaction. By comparing the hash published in the blockchain with the hash of some data, it's possible to verify that the data existed at a specific time. At the time of writing, Bitcoin can handle seven transactions per second and each transaction costs approximately $0.03 USD [3]. These limitations make it impractical and cost prohibitive to record large volumes of data in the Bitcoin blockchain. What is needed is a scalable method to record data in the blockchain and a standard protocol that allows systems to read and verify the data.

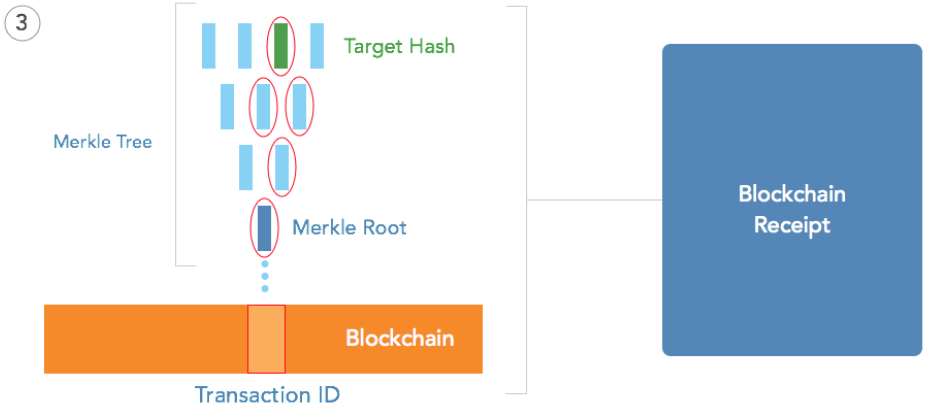
Recording Data in the Blockchain

To record data in the blockchain, we start by using a standard hashing function such as SHA-256 to generate a unique hash of the target data. Multiple hashes are assembled into a block, which is simply a list of hashes. Periodically, these blocks are used to generate a cryptographic primitive known as a Merkle Tree [4], and the Merkle Root is published in blockchain via a transaction. By collating multiple hashes into a Merkle Tree and publishing the Merkle Root, we can record large volumes of data in the blockchain using a single transaction.



Creating Blockchain Receipts

In the real world, a receipt provides proof of a transaction. A ***blockchain receipt*** provides proof that some data existed at a specific time. It contains all the information needed to prove an individual hash was part of the Merkle Tree whose root was published in a transaction in the Blockchain. By tracing a path from the Merkle root to the target hash, we can generate a Merkle Proof that proves any one of the elements is in the Merkle tree, without having to know the entire tree. These elements can be used to create a blockchain receipt that contains, at minimum, the Target Hash, Merkle Proof, Merkle Root, and Bitcoin Transaction ID.



Reading Blockchain Receipts

A standardized Chainpoint receipt format allows any system to verify a receipt by checking a Bitcoin transaction and using math to verify the data.

Chainpoint 1.0 Blockchain Receipt Standard:

Chainpoint receipts are written using JSON, the default hashing algorithm is SHA-256, and hashes are stored in the Bitcoin blockchain.

Header	
chainpoint_version	version of the Chainpoint standard
hash_type	hashing algorithm used to encrypt target data (sha-256)
merkle_root	root of the Merkle Tree that is published in the blockchain
tx_id	transaction id where the Merkle Root is stored
timestamp	non-authoritative Unix timestamp of the target
Target	
target_hash	hash of the target that is being recorded in the blockchain
target_proof	Merkle proof used to prove target_hash is part of Merkle tree
URI (optional)	path to the target
Extra	
custom (optional)	user defined metadata

JSON example of a Chainpoint receipt:


```
{
   "header": {
      "chainpoint_version": "1.0",
      "hash_type": "SHA-256",
      "merkle_root": "6a9a3c86d47f1fe12648c86368ecd9723ff12e3fc34f6ae219d4d9d3e0d60667",
      "tx_id": "012fdc0eb5ebae181e1197b4e9307731473118b0634d3ede749a562e9d11809e",
      "timestamp": 1436172703
   },
   "target": {
      "target_hash": "2f7f9092b2d6c5c17cfe2bcf33fc38a41f2e4d4485b198c2b1074bba067e7168",
      "target_URI": "http://blockchainreciept.com/target_name",
      "target_proof": [
         {
            "left": "e1566f09e0deea437826514431be6e4bdb4fe10aa54d75aecf0b4cdc1bc4320c",
            "parent": "0fdd6b6895e15115c262f6acb9a6ae0c73248568b740454ab21591f8a533dd7f",
            "right": "2f7f9092b2d6c5c17cfe2bcf33fc38a41f2e4d4485b198c2b1074bba067e7168"
         },
         {
            "left": "0fdd6b6895e15115c262f6acb9a6ae0c73248568b740454ab21591f8a533dd7f",
            "parent": "6a9a3c86d47f1fe12648c86368ecd9723ff12e3fc34f6ae219d4d9d3e0d60667",
            "right": "3b7546ed79e3e5a7907381b093c5a182cbf364c5dd0443dfa956c8cca271cc33"
         }
      ]
   },
   "extra": [
      { "custom_key_1": "value_1"},
      { "custom_key_2": "value_2"}
   ]
}
```

Chainpoint 1.0 Block Standard:

A standardized Chainpoint block is a collection of data items which are summarized and inserted into the Bitcoin blockchain. Some systems may archive blocks to dynamically generate blockchain receipts.

Block	
chainpoint_version	version of the Chainpoint standard
block_num	number of block stored in local database
closed	boolean indicating if additional items can be added to the block
leaves	hashes of all the data items in the block
merkle_root	cryptographic summary of all the data items in the block
tx_id	transaction id where the Merkle Root is stored

JSON example of a Chainpoint block:


```
{
   "block_num": 1,
   "closed": true,
   "leaves": [
      "e1566f09e0deea437826514431be6e4bdb4fe10aa54d75aecf0b4cdc1bc4320c",
      "2f7f9092b2d6c5c17cfe2bcf33fc38a41f2e4d4485b198c2b1074bba067e7168",
      "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855"
   ],
   "merkle_root": "6a9a3c86d47f1fe12648c86368ecd9723ff12e3fc34f6ae219d4d9d3e0d60667",
   "tx_id": "012fdc0eb5ebae181e1197b4e9307731473118b0634d3ede749a562e9d11809e"
}
```


Verifying Blockchain Receipts

Verifying a Chainpoint receipt requires access to the Bitcoin blockchain to retrieve merkle_root from OP_RETURN.

1. (Optional) Download and hash data at target_URI
2. Verify that target_hash is in target_proof
3. Parse target_proof to make sure the result is merkle_root
4. Check tx_id in the blockchain and make sure merkle_root is in the OP_RETURN field

Storing Blockchain Receipts

Chainpoint receipts can be stored in a centralized database or a decentralized system such as Storj or Maidsafe.

Other Blockchains

While the initial Chainpoint specification is designed to support the Bitcoin blockchain, the protocol is blockchain agnostic. The Merkle root for a blockchain receipt could be stored in Ethereum, Factom, or any other blockchain.

Conclusion

We have outlined a scalable protocol for recording data in the blockchain and generating blockchain receipts. An open source implementation of the Chainpoint protocol is available at http://github.com/chainpoint.

References

[1] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, (2009). https://bitcoin.org/bitcoin.pdf.

[2] M. Araoz. What is Proof of existence?, (2014). http://www.proofofexistence.com/about.

[3] “Transaction Fees.” Bitcoin Wiki, (2015). https://en.bitcoin.it/wiki/Transaction_fees

4] R.C. Merkle. Protocols for public key cryptosystems, (April 1980). IIn Proc. 1980 Symposium on Security and Privacy, IEEE Computer Society, pages 122-133.