Distributed Ledger With Secure Data Deletion

Vitalii Demianets (norbloc AB), Astyanax Kanakakis (norbloc AB)

Revision 1.4, November 2016

Abstract

One of the core attributes of distributed ledgers, the immutability of recorded data, presents serious challenges when regulation prohibits permanent retention of information. In those cases, entities storing personal data are legally obliged to delete it at the request of the data owner ("right to be forgotten"). This whitepaper describes a distributed ledger protocol that incorporates a deletion mechanism without compromising the security of the blockchain. Its core thesis lies on securely retaining cryptographic linkage between all components of the blockchain, even after record modification, with the introduction of a new type of transaction and Merkle Tree creation. The paper is targeting general audience involved in record keeping operations. A technical yellow paper will be released in the coming months.

Contents

1. Introduction

Distributed ledgers have been put forward as a pivotal breakthrough in many sectors ranging from transaction settlements, e.g. equities transactions, to record keeping, e.g. KYC (Know-Your-Customer)[1] in financial institutions. Their unique structure allows for the highly secure creation of an immutable ledger, holding one common version of entries across participants, and the retention of a permanent record of actions on its contents.

The advent of digitalisation of records across industries, e.g. healthcare to e-commerce, has at the same time created significant political pressure for greater control of personal data. A case brought forward to a Spanish court in 2010 from a Spanish citizen against a newspaper and Google Inc. acted as a catalyst for the creation of a legal framework surrounding the deletion of personal data at the request of its source.

The "right to be forgotten" or "right to erasure" is the legal allowance for private persons to request all their data to be erased from data depositories, especially once it is not relevant any longer for the purpose it was initially collected. Failure to comply to that regulation caries significant fines, ranging in the millions of euros. As a result, any practical application of a distributed ledger in records that falls within the regulations scope, should allow for deletion of data at the request of its owner.

Our proposal provides a solution for blockchain data deletion on a permissioned distributed ledger with known processing nodes. The resulting protocol is the platform for norbloc's KYC application, an area where the "right to erasure" is crucial.

[1] KYC is the process followed by regulated entities requiring collection of client identity, business mandate and transaction information.

The ability to confirm that the nodes are well behaving and diligently perform the physical deletion of replicated data outside the blockchain is out of the protocol's control. It is assumed that nodes are incentivised to perform any requested replicated physical data deletion by legal means.

2. Protocol design

The protocol design is presented below on a bitcoin blockchain due to its relative simplicity as well as widespread reader familiarity.

The core of security of the bitcoin blockchain is the elegant cryptographical linkage of all major components of the ledger. Specifically,

- Transactions are linked to each other mainly[2] through the Merkle Tree
- The Merkle Tree has its root incorporated to the block header
- The block header includes a reference to the block header that precedes them (Figure 1)

That cryptographically enforced interconnectivity fosters the stability and security of distributed ledgers. At any point a link between any of the components is broken, it leaves those exposed to malicious attacks.

[2] For simplicity purposes, we do not refer to the interdependence of transaction inputs (TxIn) and transaction outputs (TxOut), outputs typically being an asymmetric cryptography challenge that the inputs contain the response to. This is a protocol nuance present in the bitcoin blockchain but not in Hyperledger Fabric or Ethereum

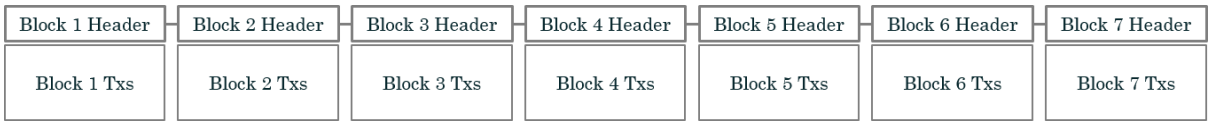

Figure 1.

In the bitcoin blockchain all data is stored in transactions. Hence, data deletion refers to transaction modification that leaves all the inputs and non-zero value outputs intact and only deletes the data contained in the zero-value "un-spendable" outputs (OP_RETURN, nulldata). That kind of transaction outputs cannot be used in subsequent transaction inputs and hence its modification does not alter the chain between transaction inputs and outputs in the blockchain.

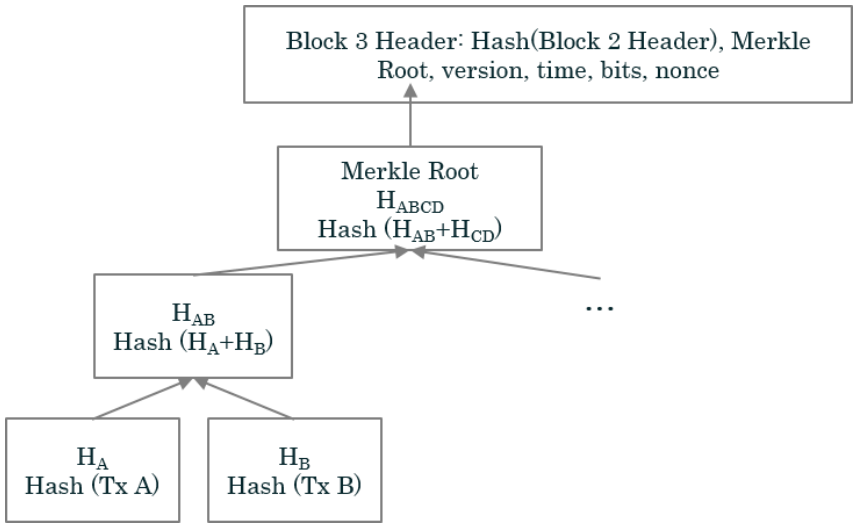

Figure 2.

Transactions are also cryptographically connected to the rest of the blockchain structure mainly[3] through the Merkle Tree (Figure 2). Once a transaction is modified within a block, with all other parts remaining stable, the link between all transactions of the block and its header is broken. The new resulting Merkle Tree root does not match the

[3] For simplicity purposes, we do not refer to the interdependence of transaction inputs (TxIn) and transaction outputs (TxOut), outputs typically being an asymmetric cryptography challenge that the inputs contain the response to. This is a protocol nuance present in the bitcoin blockchain but not in Hyperledger Fabric or Ethereum

one already in the block header, hence providing no connectivity to the rest of the blockchain. If we proceed to change the Merkle tree root in the block's header we will in turn break the chain of headers and thus the security model of the blockchain itself. Therefore, if we only change the contents of a block, the rest of the blockchain components remain stable and secure, especially as the block headers provide the connecting links by including a hash of the previous block header into the header of the next block (Note: as described above, we do not alter the connection between TxIn and TxOut with the modification of zero-value outputs)

Hence, to securely delete/amend data in a blockchain, one also needs to ensure the link between contents of a block and its header is not broken.

Our proposal is to replace the transaction we want to modify (to delete some of the user data in one of its zero-value outputs) with a "special record". That record carries:

- All modified transaction information, specifically the inputs and non-zero outputs
- A hash of the previous transaction, that is now being modified
- A special flag ("this transaction was deliberately modified")

Figure 3.

The conforming blockchain nodes, recognizing the special flag, take the hash of the non-modified transaction when they are calculating the Merkle Tree hashes, thus retaining

the link of the block holding the amended data with the rest of the blockchain. Also, the conforming nodes are obliged to use this "previous txid" when looking up the transaction for verifying the transactions spending money from the outputs of the modified transaction.

This "special record" at that stage is not protected and can be freely modified by anyone with minimal effort. Securing the system from such an attack, we introduce the special type of transaction: "deletion/modification request" (Figure 3).

The “deletion/modification request” transaction includes the following info:

- The full copy of the new "special record" replacing the original transaction
- The height (H) of the block where this transaction resides

When this "deletion/modification request" is processed/mined and receives enough confirmations (e.g. 100 confirmations as required today for spending mined coins) all conforming nodes are obliged to do the following:

- Replace the original transaction in the historical block "H" with the “special record” found in the "deletion/modification request"
- Accept the modified copy as a valid (historical) transaction because of the presence of the confirmed and verified "deletion/modification request"

Thus, we have the same security throughout the entire blockchain as before the modification with one exception: the modified transaction is secured only by blocks after the "deletion/modification request". The data we wanted to be deleted is deleted from the memory storage of all the conforming nodes.

3. Protocol implementation

In the Bitcoin blockchain implementation, the inputs of the transaction contain the signatures which sign almost all the data in the transaction. Our proposed deletion/modification mechanism changes some outputs, thus making the signatures in the inputs invalid. This can be overcome by one of two ways:

- This transaction resides in one of the past blocks, which means it was mined and verified many times (especially considering the requirement for a large number of confirmations before deletion is executed). We can trust all those historical verifications and ignore the now invalid signatures in the inputs. There is even no reason to include the ScriptSig in the modified "special record"

- We modify the protocol and define a special kind of output as "authorized for deletion". Only the data in those outputs can be modified by the "deletion/modification request". The outputs of this kind are not included in the signed data, thus leaving the signatures valid. Nevertheless, there is a separate signature required to retain the integrity of those "authorized for deletion" fields. Only if the normal TxIn signature AND the additional "authorized for deletion" fields' signature are valid, the entire transaction is valid. The additional signature is included in the data normally signed by ScriptSig. The flag in the "special record" informs to ignore the invalidity of the additional signature after the actual deletion is done.

The protocol described above has one limitation; it can only be safely implemented on a permissioned blockchain with known processing nodes. A potential application in an open blockchain has two important security risks:

- As there is no control over the nodes behaviour, they might easily ignore "deletion/modification request" transactions without any consequences

- A malicious party can start constantly issuing a lot of "deletion/modification request" transactions thus removing the data of the third parties stored on the blockchain and constantly rewriting the history and weakening the blockchain security (Note: the time-lag between the deletion transaction being processed and the actual deletion taking place, e.g. after 100 blocks, provides a small safeguard against such an action)

On permissioned blockchains one can restrict the right to issue "deletion/modification request" only to certain trusted parties. Additionally, on permissioned and regulated blockchains the good behaviour of the nodes and control of the actual deletion of the data from the hard drives can be enforced by other (e.g. legal) means.

Finally, in the modified "special record" one can add the txid of the "deletion/modification request" to make the back-reference simpler. Otherwise each node would be forced to create and maintain an index of such back-references (which transaction requested the modification of this one).

4. Technical implementation

In the coming weeks, the norbloc team will be finalizing the technical implementation details for protocol described above. The result will be largely based on Hyperledger Fabric code with the aforementioned additional functionality. We aim to be releasing a technical paper by end of December 2016.

5. Conclusion

Blockchain or distributed ledger technology possesses many innovative features, yet not always congruent to existing laws and regulations in certain jurisdictions. Adherence to data protection laws is crucial for the widespread adoption of that technology as the standard storage and processing platform.

The protocol enhancement outlined above reconciles the regulatory requirements with a solid blockchain protocol. The data deletion can take place in a controlled, secure and transparent fashion, controlled by all permissioned nodes.

Bibliography

- "How does the data protection reform strengthen citizens' rights?" - Věra Jourová Commissioner for Justice, Consumers and Gender Equality
- Factsheet on the "Right to be Forgotten" Ruling (C- 1/12) – European Commission
- Bitcoin: A Peer-to-Peer Electronic Cash System - Satoshi Nakamoto

Proof of creation time

This whitepaper is timestamped with the help of the Bitcoin blockchain by putting the SHA512 hash sum (as calculated by the GNU utility sha512sum) into a transaction and relaying the transaction in the Bitcoin p2p network. The timestamp of the block which includes the transaction is the proof of creating the described method not after the said timestamp