Confidentiality Technologies on Blockchains

In-depth technical analysis

Erik Rissanen, RISE

2019-04-25

With support from

VINNOVA Sweden's Innovation Agency

Swedish Energy Agency

FORMAS

Strategic innovation programmes

Introduction

This report is part of the broader work of the project "Traceability – For sustainable metals and minerals". During the project, a market survey was performed, and it became clear that business confidentiality is going to be a crucial feature of any traceability system for metals and minerals. A blockchain system offers some unique challenges regarding privacy when compared to a traditionally architected information system. This survey with an overview of privacy technologies on blockchains is therefore provided.

The Challenge of Confidentiality on Blockchains

A blockchain is a distributed database of transactions. The blockchain is maintained by a set of computing nodes, called "miners" or "validators". These validator nodes work to reach consensus on the validity and order of the transactions on the blockchain. To do this the validators need to see the transactions and any data which is relevant for the validation rules. Even further, If the blockchain is a public blockchain, then the data is publicly available on the Internet and the transactions are visible to anyone with the right software.

Because of this, confidentiality of data can be problematic on a blockchain. In the case of metal or mineral certification information such as amounts and prices and who is doing business with whom could potentially be revealed to outside parties.

However, there are different approaches and technical solutions for these issues. In this section we will briefly point to different options which are available, and we make recommendations for a solution.

Richard Chen has written an overview[1] of different privacy solutions for crypto currencies. Another overview of the problem is provided by Vitalik Buterin[2]. The Corda project has also provided a good overview[3] of the subject.

Two major aspects of confidentiality on the blockchain are the source and destination parties of transactions and the contents of the transactions. For example, in case of a Bitcoin payment the parties are the payer and the recipient of the payment and the content of the transaction is the amount that was paid. In case of mineral and metal certifications, the parties are the different actors in the mineral and metal value chain. The contents of the transactions will be information such as what kinds and amounts of material someone is in possession of.

Confidentiality of Transaction Parties

Blockchains operate through cryptographic signatures to authenticate transactions. A cryptographic key is just a binary number and on its own does not reveal which real world entity controls the private key. This provides a layer of confidentiality. While the content of the transactions might still be seen (unless other controls are introduced, see below), one cannot directly know who is performing the transactions.

[1] https://thecontrol.co/an-overview-of-privacy-in-cryptocurrencies-893dc078d0d7

[2] https://blog.ethereum.org/2016/01/15/privacy-on-the-blockchain/

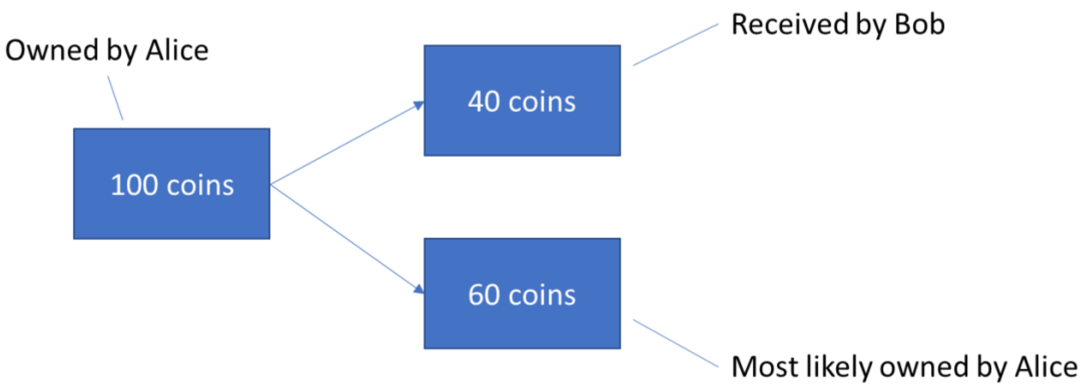
[3] https://www.r3.com/wp-content/uploads/2017/06/survey_confidentiality_privacy_R3.pdf

However, there are limitations to the confidentiality provided by this. Say if Alice makes a payment to purchase something from Bob, then Bob can see which key was used to make the payment, therefore revealing to him that Alice controls that encryption key. If Alice uses the same key multiple times, Bob knows that those other transactions also relate to Alice.

A solution to this is to never use the same key for more than one transaction. There exist various algorithms and tools to generate new keys, **single use keys,** every time a transaction is done.

However, even with single use keys there may be patterns in transactions that reveal information. In a crypto currency as Bitcoin, the currency is stored in the form of unspent transactions. One can think of a transaction like this as a virtual banknote worth the amount of bitcoin that was stored in the transaction. When making a payment in Bitcoin, enough of these transactions need to be spent to cover the amount to be paid. If a smaller amount is to be paid than what is in the unspent transaction used to make the payment, the change needs to be returned to a new transaction controlled by the payer since the inputs are invalidated as soon as they are spent. This leaves patterns in the structure of the transactions.

For instance, if Alice pays Bob 40 coins from a transaction which has 100 coins available Bob can be reasonably sure that the change in the form of the remaining 60 coins was paid back by Alice to herself. By tracing related transactions like this there could be leaks of confidential information. For instance, one might be able to correlate transactions to estimate how much money someone owns.



One approach to hide patterns in transactions is to **mix payments**. If multiple participants mix their intended payments in the same transaction, then it becomes harder to tell who intended to pay whom.

However, mixing of transactions is not always a reliable method to protect confidentiality. There are still patterns available to be seen, such as when they payment amounts differ much in size, then it can become quite clear who paid whom.

There are practical problems in creating reliable and efficient mixing services. The mixing service might steal money, it can take a long time to accumulate enough pending transactions to have enough transactions to mix with each other, and there may be difficulties in providing the payment instructions anonymously to the mixing service and counter party. Finally, the mixing service is an entity who knows the transaction patterns and thus becomes a potential vulnerability.

Some of the transaction patterns can be hidden by **not merging transactions**. As said, these merges are required when someone needs to make a larger payment than what they have available from any single past transaction they own. For instance, as when someone pays 100 coins and merges 30 + 30 + 40 coins to make the payment. This payment correlates the three transactions being spent to belonging to the same person. By tracing back where the transactions in turn came from, it may be possible to trace networks of payments to the same person, revealing payment histories and finding out how much money that person owns.

One could avoid making merges by managing transactions in sizes so that typical payments can be made without merges. By not merging transactions, there will be less patterns to be traced. In the case of metal certificates, it might be easy to avoid merging by standardizing all transactions into standard lot sizes or allocating certificate credits in sizes matching individual customer orders.

Cryptographic alternatives

The above-mentioned approaches for trying to hide transaction patterns all have their limitations, but there are cryptographic techniques to provide stronger confidentiality for transaction parties.

Ring signatures hide the source of a transaction by mixing the transaction signature with a random set of other keys from the blockchain. A **stealth address** is an address that the sender constructs from the key of the recipient in a manner that only the recipient who knows the private key can recognize that they are the recipient of the transaction.

There are several crypto currencies using various techniques and provide various levels of confidentiality. Examples include ZCoin, ZCash and Monero. The protocol which Monero uses for masking the sender and receiver of transactions, CryptoNote, is available as an open source platform and is re-used in several crypto currencies.

Confidentiality of Transaction Content

Providing confidentiality of transaction contents is challenging because the blockchain nodes need to validate the transactions. The validation includes whether the transactions conform to the business rules of the applications. For instance, in case of crypto currency, a transaction must not create new money out of thin air. Similarly, for a metals and minerals certification application, certificates must be created only according to specified processes by accredited certification bodies.

How can this validation be done, without revealing the contents of the transactions?

There exist multiple solutions with different properties in terms of complexity of implementation and the degree of confidentiality protection provided.

Permissioned Blockchains

A permissioned blockchain is a blockchain which is not open for access to anyone. Instead only vetted participants may participate. As such, only approved participants get access to the information and permissioned blockchains offer protection against direct public access. But at least the validators still need the access to the transaction data. This can be a concern if the validators are not fully trusted by everyone.

This might be the case in certification of metals, where a shared and standardized system for tracking certifications would contain a lot of sensitive information. Who is entrusted to operate this system? It is likely going to be a special purpose entity set up by the industry. Would everyone in the industry trust that this entity does not access the data in an unauthorized manner?

Data Encryption

Encrypting of the data on the blockchain is a potential a solution to confidentiality issues. However, it may be necessary that validators get access to the encrypted data in order to validate the transactions. If the validators cannot be trusted with this access, then some other solution will be required.

In other cases, depending on the business requirements of the solution, some data might not need to be validated. In that case the data can be encrypted and revealed only to relevant parties.

Avoiding Sharing of Data

The Corda[4] distributed ledger uses a different approach than traditional blockchains. It maintains a ledger which is not being broadcast to all the participants of the system, but only to the parties involved in any specific transaction. It uses a concept of a special "notary" node, selected by the transaction participants, who keeps track of which transactions have been spent to prevent double spending.

This approach, which mimics traditional bilateral business communication, protects the confidentiality of the information from outside parties.

However, to validate a transaction, all its parent transactions need to be validated to ensure that the transaction originates from a valid source according to the rules of the system. For the metal certification application this means that as the metal flows down the value chain, each participant downstream will see the history of transactions, which is a confidentiality leak.

This problem can be resolved by letting trusted parties re-issue the data on the ledger, which will cut the history out of view. This means that the parties re-issuing certificates must be trusted to not do so incorrectly. However, since a certification system relies on trusted certification bodies to begin with, it may also be a suitable solution for our purposes. In this case the system is not so much a blockchain, rather than a traditional distributed system.

Hashing

Hashing means that the data is not put on the blockchain. Instead a hash value of the data is calculated and the hash value is put on the blockchain. It is not possible to derive the original data from the hash, but when someone is given access to the original data, then they can verify that the data matches the hash, thus proving that the data existed at the time the hash was stored in the blockchain.

[4] https://www.corda.net/

Hashing is a tool that is well suitable for storing evidence of something on the blockchain, like knowledge of some information or dating of a contract. However, since the hashes are not validated in any way by the blockchain validators, there is no enforcement of any business rules.

To enforce business rules, the data must be revealed to trusted parties or to the public at some point.

Trusted Hardware Enclaves

A trusted hardware enclave is a technology in the CPU of a computer. It creates a memory area which is encrypted so it cannot be accessed by the regular software on the computer. The software running in the enclave can receive encrypted data from the outside, which the software in the enclave can decrypt and access only internally, hidden away from the normal operating system and programs on the computer. The CPU can perform computations on this hidden data and certify the result of the computation. Outside programs can know that some computation was done, but the data is protected from view, even by those owning the computer. As an example, Intel offers the Software Guard Extensions, SGX, available on most modern Intel CPUs.

Running business rules validation logic inside an enclave protects the confidentiality of the data, even from the validator nodes themselves, since the data is contained out of view inside the chip. Enigma[5] is one blockchain platform making use of SGX.

The drawbacks of the technology are the dependency on a single vendor for providing the hardware and the security of the hardware itself. Historically there have been various bugs and vulnerabilities in CPU hardware, such as the Spectre and Meltdown attacks which allowed a program to read the memory contents of other programs.

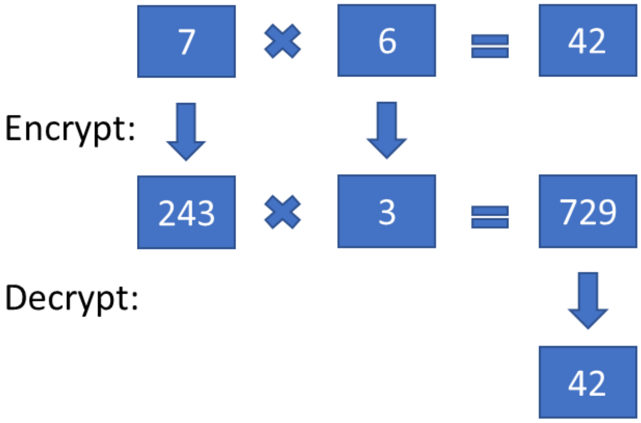
A hardware enclave offers much capability in a practical manner so this might be an option for a metal certification system if confidentiality is required.

Zero Knowledge Proofs

There are cryptographic protocols called “zero knowledge proofs”. These allow one party to prove to another by means of mathematics that they know a solution to a problem without revealing the specific solution to the other party. These protocols can be used to prove the correctness of transactions, such as that the sum of the inputs is greater than the sum of the outputs and that all outputs are non-negative, without revealing the transaction details. More general variants can be used to prove the execution of fixed size computer programs.

There are several alternative technologies and the math is very complex and beyond the scope of this report. However, we can provide some intuition for how it might be possible to perform computations on data which is encrypted. Let us say we want to prove that a transaction is correct according to some business logic. There exist encryption algorithms which preserve mathematical operations such as multiplication and addition on the encrypted values of data. Consider the following figure:

[5] https://enigma.co/



This is a made-up example, but it illustrates the principle. We have two numbers, 7 and 6. Their product is 42. If the numbers 7 and 6 are encrypted, we get some other numbers, say 243 and 3, for example. If we now multiply these two numbers, we get another number, 729. What is interesting here is that if we decrypt this number, we would get the product of the original two numbers, 42.

This example illustrates that it is possible to perform computations on data which has been hidden by encryption. The real algorithms are far more complex than this, but using them, fixed size computer programs without loops can be translated into a series of multiplications and additions, and it is possible to execute these computer programs in a manner analogous to our made up example.

With such algorithms it is possible to implement protocols where the encryption protects the data and it is possible to verify that the transactions follow the rules of the system.

Examples of such technologies are “zk-SNARKS”, “zk-STARKS” and “Bulletproofs”.

In some cases, the resulting amount of data and computation which needs to be processed becomes very big and infeasible to use in practice, but there are also working deployments, such as the privacy oriented crypto currency Monero.

Use of these algorithms currently requires highly specialized technical, cryptological and mathematical skills, but there are projects in the works to bring off-the-shelf tools to a wider industry audience. One such example is the Dero blockchain which aims for generic smart contract programmability secured by bulletproofs. Another example is the company Starkware[6] who is working on a generic platform for transaction confidentiality.

Because of the complexity of these technologies in their current form, they are unlikely to be practically applicable for our needs, but this may change in the future as the technologies and tools get more mature.

[6] https://www.starkware.co/