TU DELFT

MASTER THESIS

Blockchain-based distributed tamper-proof filesystem using threshold encryption

Author:
Angela PLOMP

Supervisor:
Dr. Johan POUWELSE

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Distributed Systems Group
Computer Science

January 11, 2018

Chapter 1

Introduction

1.1 Sensitive medical data

1.1.1 Digitalized medical records

Medical health records, once stored on paper cards in the doctor's office, have moved towards digital files that can be shared between health care providers such as GP's, hospitals and specialized clinics. These records may contain extremely sensitive data: most people would not want others to know if they suffer from stigmatized illnesses like sexually transmittable diseases or mental disorders. In a more practical way, information about someone's medical history may for example have a negative effect on their chances of being hired for a job.

1.1.2 Data ownership

A central question with regard to this type of information is: Who owns the data? Many patients feel that they do not control access to their data, but would like to be able to access the data themselves, look at the history of data access and give or deny access permissions to healthcare providers (World Economic Forum, 2012). The data is about them, so they feel they should have ultimate control over it. In a particularly bad case, patients that doubts the confidentiality of their records may not make completely honest disclosures, holding back potentially crucial information. On the other hand, the data has been collected and stored by the healthcare providers. They invest time and money into this process. Data ownership should not be seen as a binary either/or choice. Moreover, the burden of coming up with policies and implementation of these policies lies on the health care provider (Kostkova et al., 2016).

1.2 Problem statement

The goal of this master thesis project, is to research the possibilities of expanding patients' power over and knowledge about their medical records. This power consists of two parts:

1. Accountability on access;
2. Validation of EMR entries from both the health care provider and the patient.

In addition to this, the more traditional requirements for an EMR still stand. For example, the files should be kept secret for unauthorized people through strong encryption.

1.2.1 Accountability on access

Accountability on access means that a patient can verify who has accessed accessed a file, and when. There should be no way for someone to access the file without leaving a trace. When a patient questions the legitimacy of an access event, the person who looked into the file can be asked for an explanation.

1.2.2 Validation of EMR entries

Validation of EMR entries means that an entry becomes official only when both the patient and the health care provider have agreed to the entry. This is similar to a person sending a registered letter and the recipient signing for delivery. The patient cannot claim not to know the content of the entry.

1.3 Research question

Taking the aforementioned considerations into account, the research question for this thesis project is as follows:
R: *"How can an Electronic Medical Record (EMR) system be designed, that guarantees accountability on access and validation on entry addition?'*

This question can be split into two subquestions:
R1: *"How can accountability on access be guaranteed in an EMR?"*
R2: *"How can entries be validated by a patient as well as a healthcare provider in an EMR?"*

In Chapter 2: Previous work, the existing literature on these topics is explored. A possible solution is proposed in Chapter 3.

Chapter 2

Previous work

2.1 Blockchain

Considering that we are looking for a system that ensures that access to it is being logged in a tamper-proof way, a technology that comes to mind is blockchain. Blockchain emerged in 2008 with the implementation of the first cryptocurrency, Bitcoin. Essentially, blockchain is a peer-to-peer distributed ledger, which can only be updated via consensus (Nakamoto, 2008). It runs as a layer on top of TCP/IP. Blockchains can be public, private or semi-private. Anyone can participate in a public (or permissionless) blockchain: all participants hold a copy of the ledger but none of the participants actually own the ledger. This ensures the decentralized nature of the blockchain. A private blockchain is open only to an organization or consortium. Semi-private blockchains are a combination of a public and private part (Bashir, 2017). A block minimally consists of:

1. The hash of the previous block;
2. A nonce (number used only once);
3. A bundle of transactions.

The first block in a blockchain is called the genesis block. This is hardcoded at the time the blockchain was started. To add a block to the blockchain, all nodes must agree on a single version of truth. There are roughly two categories of consensus mechanisms (Bashir, 2017): Proof- and leader-based or Byzantine fault tolerance-based Bitcoin uses the proof-of work consensus mechanism to prove that enough computational resources have been spent in order to propose an addition to the blockchain. Nodes can compete with each other to be selected in proportion to their computing capacity. For Bitcoin, the proof-of-work requirement is as follows: H (N $||$ P_{hash} $||$ Tx $||$ Tx $||$ . . . Tx) < Target. N represents a nonce, P_{hash} is the hash value of the previous block and Tx are the transactions in the proposed block. The hash value of these concatenated fields should be smaller than the set Target for difficulty. This problem cannot be solved with a smart algorithm: it must be brute forced. A major quality of this system is the effectiveness against Sybil attacks as a result of the high costs of creating pseudonymous identities (Vukolić, 2015). A drawback is that it is (obviously) computationally intensive, and therefore uses much energy, which is an unnecessary strain on the environment. The proof-of-stake algorithm uses the stake that a user has in the system, for example invested time, to trust that the benefits of performing malicious activities would not outweigh the benefits of staying in the system as a trusted member (Bentov et al., 2014).
Deposit-based consensus requires putting in a deposit before proposing a block to be added to the blockchain. In case the block is rejected by others, the user loses its deposit (Solat, 2017). Reputation-based mechanisms let members elect a leader

node, based on the reputation it has built on the network. When a transaction is added to a block, it should be clear who has performed this transaction. Particularly in the medical use case, any access to the EMR should be linked to an identity. A digital signature confirms the identity, under the condition that such a signature can be verified but cannot be forged. Digital signatures can be issued using different algorithms. Bitcoin uses the Elliptic Curve Digital Signature Algorithm (ECDSA). Adding a block to the blockchain is done through the following consensus algorithm (Nakamoto, 2008): new transactions are broadcast to all nodes; each node collects transactions into a block; in each round, a random node (selected by the proof-of-work) gets to broadcast its block; other nodes accept the block if and only if all transactions in it are valid; nodes express their acceptance of the block by including its hash in the next block they create. As a rule of thumb, a block is 'permanently' added if it has been in in the blockchain for six rounds. The probability of another version of the blockchain, not containing this particular block, becoming longer and thus the official blockchain, is negligible. Because every block contains a hash pointer to the previous block, one can access the previous information, but also verify that it has not changed. Tampering is evident because the hash of the changed information would change, too. A binary tree with hash pointers is called a Merkle tree. Advantages of Merkle trees are: a Merkle tree can hold many items, but one just needs to remember the root hash one can verify membership of the tree in just O(log n) time and space (Szydlo, 2014) Athough data can be stored in a blockchain directly, a blockchain is not suitable to store large amounts of data. This is why many blockchain-based systems use a distributed has table (DHT).

2.2 Blockchain-based EMR systems

This research would definitely not be the first to incorporate blockchain into a EMR system. A white paper from Ekblaw et al. (2016) identifies interoperability challenges between healthcare provider systems as a major barrier towards effective data sharing. They designed a public key cryptography-based blockchain structure that could be applied to create append-only, immutable, timestamped EMRs. The block content consists of information about data ownership and viewership permissions. Zyskind & Nathan (2015) proposed a model called OpenPDS for an information system in which a mechanism for returning computations on the data is included: return answers instead of data itself. This paper is probably the closest related to the proposed research. The contribution of this paper is twofold: Combination of blockchain and off-blockchain storage to construct a personal data management platform focused on privacy; Perform trusted computing on blockchain-handled data. The proposed systems treats users as the owners of their data and provides them with data transparency and fine-grained access control. A rough sketch of the functionality of the system is as follows: A users installs the application on a smartphone. Data collected on the phone is encrypted using a shared encryption key and sent to the blockchain. The blockchain routes it to an off-blockchain key-value store using a DHT, only retaining a SHA-256 hash pointer. Anyone wanting to access the data can send a request to the blockchain, which in turn verifies the digital signature of the requester as well as the listed permissions for this user. Assuming that users manage their keys in a secure manner, the system provides security and privacy. An adversary cannot really learn interesting information from the blockchain itself, because it only stores hash pointers. Even if it would control a large amount of

nodes, the raw data is still encrypted using a key that none of the nodes possess. Adversaries are prevented from posing as a user because of the digitally-signed transactions and the decentralized nature of blockchain. In 2016, Xiao Yue presented a fairly similar system called the Healthcare Data Gateway app. It is a combination of a traditional database and a gateway. Personal electronic medical data is managed by a blockchain. All data requests are evaluated and in case of a positive permission, secure multiparty computation (sMPC) is used to process patient data without risking patient privacy. Enigma is a computation platform proposed by Zyskind et al. (2015). Their paper states that blockchain can neither handle privacy nor heavy computations. Enigma can be connected to an existing blockchain. The goal of the platform is to facilitate developers to build privacy-by-design, decentralized applications without using a trusted third party. Just like most blockchain-based systems, it uses a DHT that stores references to the data. sMPC is used by splitting date between nodes and performing computation on these nodes without transferring any information from one node to another. Each node has a piece of seemingly random data, that is useless on its own. In general, sMPC systems are based on secret sharing. This is a category of threshold cryptosystems, in which a secret s is divided into n parts, and at least t shares are required to reconstruct s. Such a system is written as a (t, n) threshold system. Shamir's secret sharing scheme is a famous example of a secret sharing scheme, which uses polynomial interpolation. The Enigma platform provides an API which facilitates the uses of a sharing scheme based on Shamir's scheme. In total, there are three decentralized databases in the system: the public ledger, the DHT and the sMPC database. Nodes are compensated for their computational resources via computation fees.

2.3 Digital signatures

As paperwork has been replaced by digital entries, digital signatures have taken over the role of traditional signatures. A digital signature provides proof of the integrity of the authorship, because anyone can verify that the signature is based on the author's public key. On the other hand, only the person who creates the message should be able to generate a valid signature. In general, the steps to create a digital signature are as follows:

1. The signature algorithm is a function of the signer's private key k_{pr}. Hence, only one person can sign a message x, assuming that the private keys are kept secret.

2. The message x is an input to the signature algorithm as well, to make sure that the signature is related to the message and cannot be re-used.

3. A digital signature algorithm is run with the right inputs, which yields signature s. Then, s is appended to x and the pair (x, s) can be sent.

Digital signatures can be created using a range of different algorithms, based on for example prime factorization (RSA-based signatures) or the discrete logarithm problem (ElGamal-based signatures) or on the elliptic curve discrete logarithm problem.

2.3.1 Elliptic Curve Digital Signature Algorithm

Elliptic curves have some advantages over RSA and discrete logarithm-based schemes. One of these advantages is that a small key length provides the same security as

other schemes, but with a shorter processing time. The Elliptic Curve Digital Signature Algorithm (ECDSA) is defined over prime fields as well as over Galois fields. Here, the procedures for the more popular version over prime fields are given.

1. For key generation, an elliptic curve E is chosen with modulus p, coefficients a and b and a point A which generates a cyclic group of prime order q. Choose a random integer d such that $0 < d < q$. Compute the new point $B = dA$.
   $k_{pub} = (p, a, b, q, A, B)$
   $k_{pr} = (d)$

2. In order to generate a signature, an integer such that $0 < k_E < q$ is chosen as an ephemeral key. Compute $R = k_E A$. Let $r = x_R$ (the x-coordinate of point R) and compute the signature $s \equiv (h(x) + d \cdot r)k_E^{-1} \bmod q$.

The main analytical attack against ECDSA, assuming that the parameters are chosen correctly, is trying to solve the elliptic curve discrete logarithm problem. Considering that this is an NP-complete problem, it is extremely unrealistic to solve this in time.

2.3.2 Elliptic curve threshold signatures

Similarly to the threshold encryption schemes discussed before, threshold cryptography can be applied to digital signatures. A scheme to achieve this was first presented in 1992 by Desmedt & Frankel. This method was based on the RSA signature scheme. Since then, many papers have been published presenting threshold signature schemes. For this project, the focus will be on

Chapter 3

Proposed solution

3.1 Blockchain choices

Private blockchain. Proof of stake?

3.2 Digital signature algorithm

Threshold ECDSA. Benefits (key length etc).

Appendix A

Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:

`\hypersetup{urlcolor=red}`, or
`\hypersetup{citecolor=green}`, or
`\hypersetup{allcolor=blue}`.

If you want to completely hide the links, you can use:

`\hypersetup{allcolors=.}`, or even better:
`\hypersetup{hidelinks}`.

If you want to have obvious links in the PDF but not the printed text, use:

`\hypersetup{colorlinks=false}`.