Product Vision

BBW
Jasper Hu, yjhu, 4356241
Naqib Zarin, nzarin, 4384474
Ashay Somai, asomai, 4366220
Ymte Broekhuizen, yjbroekhuizen, 4246586
Luat Nguyen, tlnguyen, 4467574

May 11, 2017

Abstract

This report presents a way to implement a decentralized way to interchange public keys of contacts for secure communication with a proof-of-identity. This design makes use of the blockchain, and eventually creating an Android app. There are other solutions available, to verify that you are talking to the person you want to talk with, but these have their flaws. We need to improve this to get to our initial goal. We use a simple trust model which uses a single trust factor per contact. There is also the possibility to revoke keys, this is essential to deal with stolen keys. Our product is also scalable, we use peer-to-peer technology, to decentralize our network, and we need to cope with an increasing user base.

Contents

1 Introduction

The subject of our product is to create a blockchain-based web-of-trust. The goal of this subject is to make the public key of users resilient against attacks. With the append-only nature of the blockchain, the resilience is enhanced(Pouwelse, 2017).
We are building a decentralized public key store, in which each user can retrieve what another user's public keys are in a tamper-proof manner(*Web of trust*, 2017). This way two users who do not know each other but are already part of a trusted network of users can know for sure that they are talking to each other, without possible man-in-the-middle attacks. The trusted network is created by two parties who know each other in real life indicating to the network that they know each other by signing each other's public keys. When a third user now goes through this process with the first user, he now also has access to the public keys of the second user, forming a simple network, or web of trust.
When many users use this product, we hope that the resulting web of trust can serve as a useful basis for any service trying to achieve secure communication with a contact while only knowing a generic piece of information such as its name, handle, e-mail, or phone number. The network will be able to resolve this query into a list of public keys with which the user can encrypt its information so that only the contact can read it.
We start with stating our stakeholders. There we name all the people who are involved in this project. Next we will have a look at the current solutions which are available, and how we intend to improve these. After this we will describe or product, more specifically the main subjects which we need for our product. Finally we will conclude with the project specifications. Here we will describe the practical side of our project.

2 Stakeholders and Customers

For our product we have internal and external stakeholders. The internal stakeholders are the entities within the business, and the external stakeholders are entities, which are not in the business itself, who care about or are affected by its performance(*Business Stakeholders: Internal and External.*, 2016). In section 2.1 we will describe the main internal stakeholders and in section 2.2 we will in particular talk about the customers as external stakeholders.

2.1 Stakeholders

This project is done as part of the Bachelor Computer Science at the Delft University of Technology, to be more specific, it is for the course Contextprojext(*Context Project Guidelines*, 2017). So our stakeholders are the people who will assess our project. The following list describes these stakeholders and gives a brief explanation about how they are involved.

1. *dr. ir. J. Pouwelse*, our context coordinator. He is responsible for creating and assessing our project as well as keeping track of our project. Our project is creating a blockchain based web-of-trust, which relates to his own project *Tribler*, and thus he definitely knows what to expect.

2. The Teaching Assistants *S. Hugtenburg* and *M. Gribnau.* The TA's will assist the overall context coordinators *dr. A. Bacchelli* and *prof. dr. A. Hanjalic* as well as *dr. ir. J. Pouwelse*, in assessing the performance of our project. *S. Hugtenburg* is the context TA, and his focus is on if our project meets the context criteria. *M. Gribnau* is the software engineering TA and will focus on the software quality.

3. The developers. We will take no part in assessing our project, but it is our goal to make the project successful and thus passing the course. We will have to do a lot of research about the work, and understand everything thoroughly, which makes us an important stakeholder in this project.

2.2 Customers

Our product meets one big customer need. It helps every customer who deals with transactions which requires trust. For example bankers[1]. They deal with a lot of transactions, which takes place

[1] (Guo & Liang, 2016)

between them and their customers, as well as between their customers only. The bankers benefit from our product, because the chance of attackers succeeding in attacking their system decreases to a low point. They also gain potentially more customers, as customers are ensured that their money is safe with the banks and this is essential in times like this, where hacking is more popular than ever, and banks are a profitable target.
Another potential customer could be polling stations during elections(Pilkington, 2016). Every vote which is made is recorded on the blockchain. Every vote is now better protected and still transparent, preventing possible fraud. This will help in countries where corruption is high.

Basically our potential customers comes down to customers who are dealing with transactions. They will benefit from our product, by creating trust between themselves and their customers, which each comes with their own advantages.

3 Current Solutions

Parts of what we want to build already exist in commonly used products and services: bitcoin for example uses cryptographic functions to store transactions in a public blockchain; the web is encrypted and also uses proof-of-identity. While we can use some existing concepts, these solutions are not directly adaptable for our app. These products also have their own flaws and our goal is to improve this, in our own way, to get rid of these flaws. The flaws are described in *Need for Change*, and the way we want to improve this is described in *Improvements*.

3.1 Need for Change

There are two common ways nowadays to verify that you are talking to the right person: keeping a list of known identities (SSH), or trusting known authorities using TLS (HTTPS). The first solution is not scalable and tedious to maintain(*Authorized keys file in SSH*, 2017), while the second approach relies on a central server, which can be brought down or compromised(*Encryption: How It Works*, 2017). Another existing method is the *web-of-trust*: a network of users that have verified their identity with each other(*Keyanalyze Explanation*, 2017).
While the last approach is decentralized, it is maintained manually; there are tools which give you a path between two trusted nodes, but you still have to contact each node on the way to verify that they also trust a certain node. It also does not keep track of the amount of trust between users: either you trust someone or you don't. Users can manipulate the network by first becoming part of the network, then sending fake public keys of someone else to every contact.
To combat this manual problem public key servers have been brought online; they store name-key pairs for many users so that you can encrypt data using GPG for users that you do not personally know. They are often not secured and do not verify if the data is true, and since they act as authority, they are not an extension to web-of-trust.

3.2 Improvements

Our product will be an android app which facilitates public key exchange with your contacts and easy identity verification using bluetooth. Using a multichain with entanglement every user cannot deny the history of published and revoked contact-key pairs of entangled users.
This facilitates automatic key exchange between you and the contacts in your android phone, which can be further used for a variety of apps, such as secure messaging or calling, secure transactions, ssh, etcetera. Trust management is a challenging subject which has used up many research funds. The main problem of this subject is who to trust and who not. Since this is a large problem and outside the scope of this project, we will work with a trust model only using a single variable - trustworthiness.
Another improvement on existing protocols is our use of a multichain. This is a specialisation on a normal blockchain in which each block in the chain contains two inputs: one of a previous block of someone else, one of our own previous blocks. This way our own chain becomes intertwined with other user's chains and each block inside this multichain is verified and built upon by multiple users. When a stranger now wants to prove to us who he is, he can point out a specific block in this multichain with his keys and we will know enough. Special care has to be taken with lost or

expired keys: a key once valid could be lost or compromised and thus revoked; this information should be woven into the multichain.

4 The Product

The product will be an easy-to-use app that, when combined with the other projects, will let you store encryption keys for your phone contacts along with a trust value. This in turn could be used by other apps for secure communication, spam detection, prioritization, etcetera. The resulting code can also form a basis for other multichain-enabled technologies - free markets, automated exchange, reputation-building. We will have a look at implementation details.

4.1 Trust

We will use simple trust model which uses a single trust factor per contact. It will initially be zero for any stranger, and can be influenced by:

- Bluetooth authentication
  Will greatly boost the trust factor since it means you are meeting and interacting with someone in real life.

- Rated trustworthy by other users.
  A simple model would be to give a small trust bonus for every contact that also trusts this contact; more complicated models could be to also take into account the number of nodes this node is away from you and how much trust this node has received from other nodes.

- Integration with other projects.
  When an user has verified that they have an IBAN account, they could receive higher trust since an adversary would be less likely to expose their identity by using an IBAN account.

- User trust.
  When a user very much trusts another, it could be possible to manually upgrade (or downgrade) the trust of a contact.

The trust could for ease of the user be divided in manageable categories, such as highly reliable, reliable, questionable, shady: these are more user-friendly to view.

4.2 Revoking Keys

It is possible as a user to maintain multiple keys, to maintain anonymity or avoid gangsters. When a user loses a single key or decides that a certain key is not valid anymore he should be able to revoke it; this action is verifiable by following the multichain and checking that a certain user is indeed owner of a key. When users are not careful it is possible that their keys are stolen. This has a disastrous effect on the trust of the network so we should include a mechanism to cope with this; when not all keys are stolen he can simply revoke the stolen keys, otherwise we face a problem.

A most basic solution is to give each key an expiry date after which it becomes invalid. Unused keys will disappear spontaneously while keys still in use can be renewed (automatically if preferred).

Other mechanisms are for example giving a third party the ability to revoke keys of ourself at will when it suspects we are compromised, or allowing other nodes to request a change of key and a fresh bluetooth authentication.

In the first version of our product we only let the proven owner of a key revoke it and do not implement the other solutions since we do not know enough at this time.

4.3 Scalability

As on many distributed systems another question is how to scale our app well. When using a central blockchain like Bitcoin does, a general consensus has to be reached for all peers that includes all information; this can be heavy on single clients(*Scalability*, 2017). It also does not support closed communities or offline transactions since everyone should agree on the common blockchain before knowing if a transaction is valid.

By using a multichain we circumvent this issue; every user only needs the subset of the chains that are relevant to him. Efficiently obtaining and walking this subset is another challenge that we will have to figure out.

The first version of our product will avoid this problem by only keeping track of multichains of our phone contacts and ignoring all other messages. Assuming a user will have less than a few hundred contacts we do not expect the product to run into problems.

Since even on this contact scale we will work with a large amount of blocks and chains, we will implement a simple sqlite database from the beginning.

4.4 Peer-to-Peer

Ideally the entire network is decentralized so we will run into the problem of locating other peers without a server keeping track of all IPs; peers can change their IP or disconnect at any time.

This problem is not new and there are some reasonable solutions, most notably bittorrent. For complete decentralized peer discovery it uses a “distributed sloppy hash table” to find peers that possess a given torrent. This solution is adaptable for us; instead of finding peers corresponding to a torrent we have to search for peer(s?) corresponding to a certain contact or public key.

The algorithm works by generating a random ID for ourself from the same space as the information we are looking for - SHA1 hashes in the case of bittorrent - then storing all peers with a close IDs. When any user wants to know a piece of information with a certain ID, it asks nodes with close IDs for information. (Norberg, 2008)

An easier solution is a simple flooding algorithm that was frequently used in earlier P2P networks; it does not scale well, but for our initial product it will suffice. Of course simple protection has to be built-in as to not forward messages which we have already forwarded; a simple hashmap or slightly more complicated bloom filter could be used. We may also want to restrict the amount of network traffic when a user is not on wifi.

5 Project Specifications

In this chapter we will provide a list of the specifications. The more detailed version of this can be found in the product planning.

5.1 Timeline

The first two weeks of this project the main focus is to write up all the required documents. This includes the Architecture Design, Product Vision and Product Planning. These documents are necessary to specify the structure of our project, and will be formed through discussions and many meetings. Also we will implement the fundamentals of our project, which will create a better understanding of the subject and improve the quality of the documents.

In the middle of the project, our code will be reviewed by the TA’s, and the coordinators.

The following weeks the aim is to further develop our project, into something which will meet the requirements. This involves a lot of further readings, about the subject and optimizing it.

Every friday we have a meeting, along with the TA’s to discuss our progress and to check whether we are on the right track. This is part of weekly sprint, since we are using Scrum(Schwaber, 2015) for this project.

5.2 Budget

For this project we don’t have to spend real money, we are just developing our product for a university project, so we can use the resources of the Delft University of Technology. As we are developing an Android App, we also need a Android device, which is provided by the university. Because we are doing this for a school project, we do not get paid, but we spend our time, which is valuable and thus we are essentially spending something. Each week we have to spend 28 hours on this project per person.

References

Authorized keys file in ssh. (2017, May). https://www.ssh.com/ssh/authorized_keys/.

Business stakeholders: Internal and external. (2016, Jun). Retrieved from https://www.boundless.com/accounting/textbooks/boundless-accounting-textbook/introduction-to-accounting-1/overview-of-key-elements-of-the-business-19/business-stakeholders-internal-and-external-117-6595/

Context project guidelines. (2017). https://docs.google.com/document/d/19J8In7yXjVMQPCNsysOrMUZcK1h6n6OV37eOJP-xMWE/pub.

Encryption: How it works. (2017, May). https://letsencrypt.org/how-it-works/.

Guo, Y., & Liang, C. (2016, December 09). Blockchain application and outlook in the banking industry. *Financial Innovation*, *2*(24). https://jfin-swufe.springeropen.com/articles/10.1186/s40854-016-0034-9.

Keyanalyze explanation. (2017, May). https://web.archive.org/web/20090203235946/http://dtype.org/keyanalyze/explanation.php.

Norberg, A. L. (2008). *Dht protocol.* http://www.bittorrent.org/beps/bep_0005.html.

Pilkington, M. (2016). Blockchain technology: principles and applications. *Research Handbook on Digital Transformations*, 225-253. doi: 10.4337/9781784717766.00019

Pouwelse, J. (2017). *Establish + real-time display of blockchain trust.* https://github.com/Tribler/tribler/issues/2905.

Scalability. (2017, May). https://en.bitcoin.it/wiki/Scalability.

Schwaber, K. (2015). *Agile project management with scrum.* Microsoft.

Web of trust. (2017, May). https://en.wikipedia.org/wiki/Web_of_trust.