Digital Voting Pass - Research report

Wilko Meijer Daan Middendorp Jonathan Raes Rico Tubbing

May 15, 2017

1 Introduction

Currently, voting in the Netherlands is done in an old fashioned way, by paper and pencil. However, there is a strong call for digital voting systems from municipalities in the Netherlands, according to a survey by the public news organization NOS[1]. Propositions for digital voting are often met with scepticism, because people have major security concerns with digital voting. The goal of this project is to implement a digital alternative for the paper voting pass, this system must make use of blockchain technology. With this digital voting pass voters must at least be able to do the same things as with the paper variant. The digital system should improve the voting process for voters in some way, so voters will have some incentive to switch to the new system. If voters have objections to the system it is highly unlikely that it will be implemented, so gaining the approval of voters for the system is a key aspect. This approval consists of two parts: usability or ease-to-use and trust that the system is secure. Since the lives of people involve many digital solutions, it is likely that in the future a cumbersome paper voting process could invoke more frustration with the elections and in the worst case even cause people to refrain from casting their vote. A digital solution should make voting easier and prevent voting from becoming old-fashioned or outdated. Furthermore since elections have major influence on the lives of people, they should trust that the system won't be defrauded. The scope of the project is limited to replacing the voting pass for two reasons: to be able to finish the project within 10 weeks and to give the possibility for the government to set at least a small step in the direction of a fully digital voting process.

During the first days of the project we determined the rough requirements of the project and from that derived the research subjects that are discussed in this report. More information about these first days can be found in the Project Plan. These first days also resulted in the following research question which this report answers. *How can the paper voting pass in the Dutch voting system be replaced by a secure, trustworthy, and easy-to-use digital system?* Multiple architectural choices are considered and substantial decisions are made, this report elaborates these decisions and concludes the research phase of the project.

2 Current Dutch voting system

In order to better understand the setting of the problem, this section will give a short outline of how the Dutch voting system currently works.

A few weeks before the election day municipalities send voting passes to a voters home address. These arrive at least 14 days before the election. Every Dutch citizen of 18 years and older is eligible to vote for Dutch and European elections, with exception of people whose voting rights have been revoked by a judge. Revoking people's voting rights rarely happens, at the start of 2016 the voting rights of a total of 56 people had been revoked [1].

If a voter is not able to cast his vote himself, he can authorize another voter to vote on his behalf, this is called proxy-voting. There are two ways to do this:

1. For votes who live in the same municipality: by filling in the back side of the voting pass
2. For voters who live in different municipalities: by requesting a proxy-voting pass from the local municipality

[1] http://nos.nl/artikel/2165112-gemeenten-willen-graag-terug-naar-computerstemmen.html

A voter can only cast votes for himself and a maximum of two other people [2].

On election day, a voter brings his voting pass and his identification card to a polling station. Here, this person's identity is checked and if all is in order. The voter exchanges the voting pass for a ballot paper on which he marks his vote. At the end of election day all votes are counted by hand and the results are announced [3].

3 Existing digital solutions

In the past and present digital voting is practiced around the world by different governments. This section discusses some digital voting technologies and list advantages and disadvantages of these techniques.

In the first digital voting technique, voters come to a polling station and use a punch card to cast their vote [4]. The punch card has several ballot positions, where each positions represents a pre-determined candidate. A voter can use a punch device, such as a stylus, to create a hole in the card, this hole will be the vote on a candidate. A counting machine is now able to determine on which candidate, and thus also the party, has been voted. An example of such as system is Votamatic [5]. The advantage of this technique, and actually all other digital voting techniques mentioned, is that there is no need to count the vote by hand. A disadvantage is chad, chad is the chunk that is punched out of the card. If the chad is completely out of the card, the punch card is valid and a counting machine will register this vote. If this is not the case, a vote might not be registered by the counting machine and thus result in an inaccurate result [6]. In the 2000 general election a recount by hand, in Florida, was demanded by Gore (the opponent of Bush), because of chad [6].

Another technique that was used in the Netherlands, is the use of a direct-recording electronic voting system. In this system voters come to a polling station and cast a vote on a special machine [7]. Afterwards, the results of a polling station are put on a USB stick and is transported to one of the twenty locations were the results of all polling stations are counted [8]. These machines are not trusted by some people as discussed in section 4.

In Arizona a pilot with digital voting was run [9]. In this pilot a person could register online, via Election.com, and state that he or she wants to vote online. After registration this person would receive a personal PIN via mail, this PIN is needed to cast this persons vote. Apart from the PIN, the person must answer two personal questions so stealing a PIN does not directly hand out a vote.

In Estonian voters can vote with their ID card. Their ID card contains a PKI (Public Key Infrastructure), this allows a person to sign a document with his or her ID [10]. To sign a document with an ID card, a PIN must be provided. Voters can download special software to cast their votes via the Internet. The software will ask for a persons ID and PIN, when this entered correctly a centralized server will be contacted. This server checks if this person is eligible to vote and if so return the set of candidates. The person can now cast his or her vote and the vote is send to the centralized server, where it will be stored. After this, the server returns a personal QR code. With a verification app and this QR code, a person can verify if his vote is correctly recorded. Security of this system is a controversial topic, in [10] researchers have shown that the software contains several vulnerabilities. An attacker would be able to disable voting in 75 minutes via a denial-of-service attack. Besides this, an attacker would be able to perform a shell-injection which allowed root access to centralized servers.

4 Trust and the old Dutch digital system

Trust is an integral component in many kinds of human interaction, and elections are no exception. Many varying definitions of trust exist, a definition applicable to this project, taken from Mui et al: "[Trust is] a subjective expectation an agent has about another's future behavior based on the history of their encounters." [11].

A voter needs to trust the system that the vote cast by him or her is actually counted and participates in the election, that no extra votes are added, and that the vote has the same value as any other vote cast by anyone else.

4.1 Trust issues

Some people have an instant disapproval when it comes to voting digitally [12], perhaps not without reason. In 1967 some municipalities in The Netherlands already started with using computers to vote [13], since October 2007, voting computers are no longer in use due to lobbying of the group "Wij Vertrouwen Stemcomputers Niet" [14]. Mostly due to a security analysis that showed multiple technical issues with the used machines [15]. Issues addressed by this group included that the Nedap computers were 'non-voter verifiable', meaning that voters had no way of verifying that their vote was actually counted. In the current (paper) system this can be achieved because each voter has the right to watch while his vote is counted [16]. Another problem shown by this group relating to voting computers is that there usually is one central machine involved, that, if in control by a malicious party, could manipulate the outcome of the election [12]. Opponents of the Nedap computer also had problems with the closed source implementation they had and the non-transparent voting process [17].

4.2 Design philosophies

Evidently, by looking at the history of electronic voting, it can be seen that providing any digital solution to voting will be greeted with a lot of scepticism. It is clear that, in order to be trusted, a digital solution to the elections needs to be as open and transparent as possible, while still remaining secure. This transparency can be achieved by using all open source technology, both in software as in hardware and should be enough to silence sceptics and make voters trust the system. Digital voting must be voter verifiable and preferably be a decentralized implementation such as a decentralized blockchain, so that a single compromised machine does not endanger the rest of the network. End-to-end integrity checks will ensure that the correctness of the data is verified and that any anomalies will be detected, so the voters can trust the results of the elections.

5 Proposed solution

This section gives an outline of the proposed solution and states which questions about the implementation and technologies follow from this.

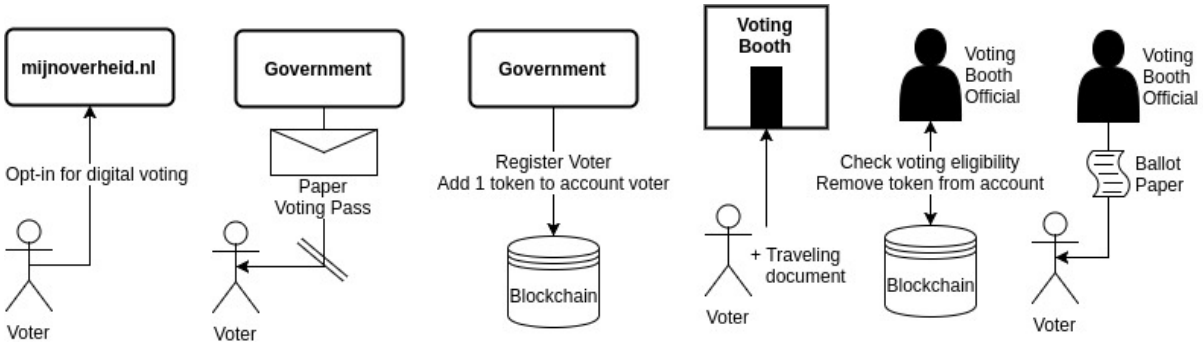

Figure 1: Schematic rendering of the proposed system

The proposal for a solution consists of a blockchain instance that defines a vote token with a centralized owner, in Figure 1 a visualization of the proposed voting process is given. Every person would be able to go to a government website (e.g. "MijnOverheid") and register him or herself as a digital voter. Then, this person would need to choose from a list of their identification documents that contain a contactless chip (this list is already available). This document's public key which is probably known to the government, will then be used to create an account on the blockchain for this person. If this public key is not known by the government, there could be a mechanism implemented where the user must register his travel document first, this can happen remotely. At some point, for example after the digital voting registration period, the government, being the owner of the blockchain, can associate a single vote token to the accounts of registered voters.

At the time of the elections, the person would go to a voting booth and bring only the identification document that was registered. There, a voting booth official will scan his document using a NFC scanner installed on a smartphone or tablet, so the public key that corresponds to the document can be read. This app will check the chain to see if the public key still has an outstanding balance. If so, a transaction is made and signed, using the voter's document, to send the vote token to a government wallet. After this transaction is confirmed it is ensured that it was valid and a passing message would be shown on the official's device. After this, a ballot paper can be given so the person can cast his or her vote.

From this proposal questions arise about the architecture. These questions will be discussed and answered in this paper:

1. Which blockchain technology is best suited?
2. How to keep track of the identity of a voter using traveling documents?
3. How to prevent double-spending of a voting pass?
4. How to allow proxy-voting?
5. How to ensure the devices used in the voting process are secure?

6 Blockchain

Key aspects of blockchain technology are its decentralized network, immutability and the transparency of everything that happens on the blockchain. The decentralized structure and its structure of linked blocks of data makes it extremely hard, if not virtually impossible with the right setups, to alter data. Transparency ensures that everyone can check exactly what happened in the voting process, which makes it very hard to defraud the system without detection [18]. This makes blockchain technology very interesting to use in digital voting implementations. Transparency in the chain creates no problems for ensuring secret ballot in this project, because the transparent transaction only shows that a vote has been cast. More specifically, it only shows that a voting pass has been exchanged for a ballot paper. This physical ballot paper is used to cast the actual vote.

This section discusses which blockchain implementation would be best to use for this project and how to map the identity of the voter to an account on the blockchain so the voting eligibility of a person can be checked.

6.1 Blockchain implementation

There is a variety of choices for the implementation of blockchain technology in the election process. Multiple smart contracts on a chain like Ethereum or Bitcoin using Counterparty are considered [19]. Another option is setting up a blockchain for this special purpose. Such a chain could be created using MultiChain [20] or OpenChain [21]. The MultiChain as implemented in Tribler is also considered [22]. Smart contracts, which have emerged in the last two years, are computerized transaction protocols which execute terms that are written in the contract [23]. These contracts can be validated by every node.

Using smart contracts on a well-known and well-reviewed chain like Ethereum, which already has a large network of miners backing it, makes a monopoly attack (51% attack) unfeasible. Ethereum smart contracts have almost the same functionality as a self-made blockchain would. Custom tokens can be generated, such as in [24], which could represent whether the owner has already claimed his voting pass or not. These can be distributed from a single authority to all vote-eligible people. A problem in this set-up is that verification of the transactions will need to happen inside the smart contract, which is an expensive operation in terms of gas, Ethereum's processing fee. Our rough estimates show that it will be a few euros per verification.

Furthermore, there will have to be some sort of mapping that maps private/public key pairs from the passport to key pairs on the network, this mapping would probably have to be made outside of the Ethereum network. This is necessary because the cryptographic implementations in the passport are different than in the Ethereum or Bitcoin network. Another advantage of the Ethereum network is that it is well-known and trusted, so this choice may receive less scepticism than others.

OpenChain is more like a transaction chain than a blockchain. It has no blocks but chains all transactions together. It can be anchored to the Bitcoin blockchain to benefit from the irreversibility of its Proof of Work [25]. The downside of OpenChain is that it relies on a client-server model where there is only 1 server node that verifies transactions [21]. Even though integrity can still be proven this opens up an attack surface for DDoS attacks. It this validator node is brought down, no transactions can be verified anymore and the network is halted. Since OpenChain uses a client-server model, the Java integration would be as easy as speaking to some endpoints in HTTP.

MultiChain is a toolset to easily setup a personalized blockchain, based on the Bitcoin core, that can be as private or public as needed. It features a permission system where nodes can be allowed or disallowed to do things like mining or even sending transactions. This is necessary to ensure only the government can be a mining node, otherwise a third party could supply so much processing power that it can do a monopoly attack on the network. MultiChain relies on nodes that mine in order to verify transactions [20]. This can be multiple nodes so the network is strong against DDoS attacks. MultiChain has a Java API library, but it does not seem to be finished or have very much support [26].

The advantage of using a personalized blockchain such as MultiChain or OpenChain is that those projects can be forked. Therefore the transaction signing implementation can be altered, so that it uses the same cryptographic keys as the ePassport, eliminating the need for a mapper. This can be changed so ePassport public keys are actual wallet addresses and a transaction signed by the document could be broadcast to the network immediately. This offers great simplification compared to a setup using smart contracts.

When using existing running blockchain platforms, transaction fees apply. This includes running smart contracts on Ethereum which costs gas [27], and on the Bitcoin chain using counterparty, which costs regular transaction fees [19]. These transaction fees can be seen as paying for electricity. When the government hosts its own blockchain they would have to pay for their own electricity and hardware to keep the network online. It would be hard to analyze these costs beforehand, but since running a node or mining can be seen as the conversion of electricity to cryptocurrency [28], the costs of running a capable blockchain or paying for the smart contract are in the same ballpark. Therefore, these transaction fees play no significant role in deciding on a blockchain implementation. However, maintaining a specialized blockchain would require buying the hardware in the first place, which is costly, and additionally it would require a lot of extra processing power just to protect the network against monopoly attack.

The Tribler Multichain is a blockchain developed by S.D. Norberhuis in a master thesis for the TU Delft. It features (near) instant verification times, each transaction is immediately put in a block in the chain. In this Multichain, nodes do not have a full copy of the chain, only of those blocks where they were one of the parties that made a transaction. Since most of the transactions will be with one party, the government, there won't be a complex network of chains so it defaults to a regular blockchain. The added benefit for voters of not having to store the entire chain is lost, since voters probably won't run their own nodes, because of issues with uptime, and the government will run the nodes for them. Crawlers can be used to verify the integrity of the chain. Using this blockchain would require taking it out of the Tribler codebase and adapting it to the needs of this project. This is considerably more effort than a solution like MultiChain or Smart contracts. In Table 1 an overview can be found of some of the most important features that were considered for choosing a blockchain implementation.

6.2 Conclusion

Due to the fast reforming landscape of blockchain technologies, a right decision made today, might be completely fallacious within a short period of time. This makes it difficult to make choices based on the current available information. Is the platform still actively maintained within one year?

Based on this, the preferred choice goes to the MultiChain.com platform, because of the decentralized structure, the available API's and documentation. If this approach turns out to be suboptimal, then we might continue the project with the OpenChain platform which is not decentralized, but seems to be much easier to configure. This platform will be modified to support transaction signatures with dutch ePassports.

Feature	MultiChain	Tribler Multichain	Smart Contract	OpenChain
Confirmation times	Configurable, minimum 2 seconds	Instant	Depends on blockchain (ETH ∼15sec, BTC ∼10min)	Instant
Robustness / Security	Strong: decentralized, multiple government nodes	Strong: Decentralized, multiple nodes	Strong: existing network with many nodes	Questionable, single server node that does verification
Cost	Moderate, multiple verify nodes needed to protect against DDoS	Moderate, government also hosts nodes of voters	Expensive transaction fees	Cheap, single server node for verification
Language and ease of Java integration	C++ with a Java API lib	Python, in production currently	Solidity, with easy Java integration	C# with easy Java integration
Documentation	Moderate	Scarce	Lots	Moderate

Table 1: Comparison of important features

6.3 Blockchain-identity

A blockchain can be seen as a database where every move can be originated to a user. Due to this, there needs to be a solid identity which handles the access control and signs every transaction. In most cases, blockchain accounts consist out of a private and public keypair, which can be used to send and receive transactions. There is a wide range of options to store these keypairs. Bitcoin for example uses software keys which are stored in the client. This makes it vulnerable for attackers to confiscate these keys an perform unauthorized transactions. Because of this, it is desired to store the keypair in a more secure environment.

6.3.1 Biometric passport key pair

Identification using PKI is something which is implemented by several governments. For example the Belgian National Identity Card, which is introduced in 2005, contains a PKI by default [29]. This is used for signing electronic documents and logging in to several governmental services, like tax returns. There is not much criticism about the security grade of this implementation, but one of the major drawbacks of this implementation is that a specific card reader is required for these cards, which almost no citizens owned. This is the reason why it is not widely used [29].

Dutch drivers licences, identity cards and passports are designed according to the ePassport (or biometric passport) ICAO standard [30]. This means that there is a NFC-chip implemented in each of these documents which can be read with freely accessible hardware. The rise of the NFC applications in all kinds of industries, from banking to travelling has brought us to point where almost everyone has access to a reader [31].

The ICAO standard has described several obligatory and optional specifications. One of these specifications is the implementation of a so called AA (Active Authentication). This implementation consists of a private key stored into the chip which cannot be copied, a readable public key, and a signing function. This makes it possible to sign data using the private key in the chip [32].

The chip in Dutch travel documents is manufactured by a French company and is not implemented exactly the same as other European Union passports. Due to this, AA was not implemented with RSA like supported by most ePassport reading libraries. After some research and different experiments it turned out that Dutch travel documents are based on ECDSA (Elliptic Curve Digital Signature Algorithm) combined with a BRAINPOOLP320r1 curve. Elliptic Curves are fortunately really common in blockchain. This makes it easier to interchange those signing mechanisms.

One major issue in this system is that this signing function is only able to sign eight bytes. A typical Bitcoin transaction has a size of 200-250 bytes [33]. Due to this, it is not possible to sign a complete transaction or a SHA-256 hash, which results in a 32 bytes hash, of it. There are also other hashing

algorithms available which create smaller hashes, but they are not recommended for cryptographic usage, because of the higher risk of duplicates. To still make it possible to use the travel document for signing transactions, the signing and verifying algorithms will be modified so that they will process partial signatures of the hash. This is less secure than signing the entire hash. The next ICOA travel document standard should contain a signing method for 32 bytes to make it more secure.

Other papers discourage the use of the passport as a digital smartcard, because of the lack of any additional authentication such as a PIN [34]. This is not completely true, to have any interaction with the passport, there needs to be a Basic Access Control performed. This consist of sending a part of the MRZ (Machine Readable Zone) which is printed in the passport to the chip, so that it is guaranteed that the person accessing the data on the passport also has physical access to it.

6.3.2 Alternative identity solutions

If transaction signing using the ePassport proves to difficult to implement, another technique to securely sign transactions is needed. There needs to be another identity vehicle with the following specifications:

- Basic Access Control. This protects the communication between the chip and reader using encryption. Before data can be read from the chip, a key needs to be provided. In the ePassport this key consists of the date of birth, the date of expiry and the document number [32].
- A signing function. For signing data with the card itself using a private key stored inside the card. This private key is not readable from the card itself, it can only be used to sign data with the card which can then be verified. This ensures that the private key can not be stolen unless the entire document is stolen.

There are different physical smart cards available which are suitable for this purpose. A major drawback of this decision is that it is expensive and complex to provide these smart cards to the voters.

Instead of a smart card, a smartphone could also be used to sign transactions. Most devices contain a so called secure element in their hardware. This secure element makes it, like the ePassport solution, possible to sign documents using the inaccessible chip. However, the API of this secure element is not available for public usage (needs system level permissions) [35].

Another option is to store the key in memory of the app. In this scenario a voter would receive or generate a private key which is stored on the persons device. This private key is then used for singing transactions in the blockchain. However, a private key on a device will pose some security problems, as the private key can be stolen with malware for example. If the private key is compromised, an attacker can take away the individuals' right to vote by signing away the vote token. Since the government has only distributed the tokens to addresses it knows belong to people that have signed up online and so will not receive a physical ballot paper, this attack can not add votes for a specific party. The attacker could only take ballot papers (tokens) away from voters and possibly use them himself to get extra votes for himself.

7 Double voting

Double voting is something that should obviously not be possible in any election system. The digital voting-pass solution should not allow this. Since it is initially assumed that the digital voting pass can be used in parallel with the paper one, it needs to be made sure that a person can not cast a vote using both systems. Obviously the digital system itself should also not allow for multiple votes to be cast by a single person (unless if that person is warranted to vote for another).

If a blockchain transaction is made to resemble a person having cast his or her vote, this problem is analogous to the double spending problem. The transaction in question must first be verified in order to be sure that the voter is not trying to double-vote. Only after it is verified that the person had not already cast a vote, a ballot paper can be handed out. This verification time should be considered when choosing a blockchain platform, because waiting a long time at the voting booth is very impractical.

This waiting time is a direct result of the blockchain's choices for block generation time. Since these shorter confirmation times also allow for faster transactions in crypto-currencies which is desirable, one can expect that these delays will only decrease over time. Furthermore there are other blockchain

technologies which make use of (near) instant verification, so verification time can be considered as not-an-issue.

8 Proxy-voting

Because the digital voting pass should be a complete replacement of the paper voting pass, proxy-voting should be possible with the digital voting pass. A short description of proxy-voting can be found in section 2.

8.1 Problems and constraints

Dutch law and the characteristics of proxy-voting raises certain problems that the system has to deal with.

A big issue with proxy-voting is checking if someone freely approved of giving someone else authority to vote on his behalf. There are two constraints against forcing proxy-voting in the current system that are important to note:

- A signed voting pass and a copy of an ID card are required [36]
- It is illegal to force someone to authorize another person to vote for him and to actively recruit for proxy-voting [37]

The digital system should at least give the same protection as the current system, and should strive to give better protection.

Death of voters poses another issue. When a voter dies, his voting pass is rendered invalid [38]. The unique identifier of his voting pass is then put on a list of invalid voting passes and no one will be able to cast a vote with his voting pass. The digital solution should propose a way to render votes invalid, even when they are transferred to another person. If the proxy-voter dies the proxy-votes should be returned to their original owner.

In the current system a warranter can revoke the proxy-voters right to vote on his behalf if the proxy-voting is done by handing over a voting pass. If the proxy-voting is done via a written request at the local municipality, this is not possible. The digital system should provide a way to revoke the proxy-voting rights.

8.2 Possible solutions

When creating a cryptocurrency, either an entire new network or a cryptocurrency based an existing blockchain like Ethereum, it is possible to set certain restraints on the network. In the system it can be controlled how much coins are handed out, but also some transaction constraints like a maximum wallet size, which in turn can limit the amount of proxy-votes a person can cast. Disabling authorization of proxy-voting a few days before election day makes it easier to detect if votes were somehow stolen. Proxy-voting for someone in another area with local elections won't be possible, since each local election would correspond to a different system. This is also not possible in the current voting system, because by law a proxy-voter is required to cast his own vote at the same time as the proxy-vote, which wouldn't be possible if the votes are for different local elections.

8.3 Proposed systems

8.3.1 Proxy-voting via regular transactions

The simplest system for proxy-voting is regular transactions of crypto-currencies.

In crypto-currencies like Bitcoin and Ethereum the basics of the transaction of a 'coin' or token come down to the following. A token is transferred from one owner to another. A transaction uses the previous transaction of the token and the public key of the receiving party in a hash. The sender signs this hash with his private key. This allows the system to check if the sending party is actually the owner of the token. Because a hash of the previous transaction of the token is included in the new transaction, the token can be tracked [28, 39].

This transaction system can be used for proxy-voting. Let's say someone we'll call Voter 1 (V_1) would like to authorize a friend, Voter 2 (V_2), to vote on his behalf. He would make a simple transaction to move his voting token to the wallet of V_2. All he would need for this is the public key of V_2 and the transfer can be made.

This system, however, has quite a few weaknesses.

- A person has no say in whether or not he becomes a proxy-voter, he could however decide to just not use the voting power.
- Because a transaction is a very simple and fast operation, there is not much time to think about the decision. This also leads to a higher risk of people authorizing proxy-voters under pressure.
- A person should verify manually if he submitted the right target address for the token, which could lead to errors and sending voting tokens to the wrong person.
- The token of V_1, which was sent to proxy-voter V_2, could be sent to Voter 3 (V_3). This leads to a situation where V_3 can cast a vote on V_1's behalf without his consent.
- There is no possibility to revoke the rights to proxy-vote, once a transaction is sent and accepted the original sender has no influence over the token.

The current paper system does deal with these problems and thus this digital solution can't be used as a proper replacement.

8.3.2 Proxy-voting with smart contracts or modified transaction functions

The way transactions work can be influenced or modified to enable the use of many more rules and constraints on transactions. This can either be done via smart contracts like the ones Ethereum uses, by modifying the transaction function, or adding specialized transaction functions to a blockchain implementation like Multichain.

Revoking proxy-voting rights - Let's say Voter 1 V_1 enters into a proxy-voting contract with V_2. This proxy-voting contract says that V_2 gets the voting token of V_1 a certain time before election day if V_1 doesn't submit some kind of objection (this could be implemented in many ways). Alternatively V_1's token can be transferred to V_2 at the start of the contract and can be transferred back when the contract is voided.

Additional verification - To prevent giving proxy-voting rights to the wrong person or V_1 giving proxy-voting rights to V_2 without consent, smart contracts can be used to implement a verification system. In this system V_2 should perform a verification action to confirm that he accepts to act as a proxy-voter, after that V_1 should confirm that he actually wants to give his voting rights to V_2.

Resending a token - To prevent V_2 from sending V_1's token to V_3, a simple addition to the contract concerning transactions is needed. This contract should check whether the sender of a token is the original owner, if this is not the case the token shouldn't be sent.

To make it easier for the user to set-up proxy-voting an app should be developed, which calls the appropriate functions or creates a smart contract. This could create a vulnerability in the system when the device of an user is compromised, this could allow an attacker create a fake app which executes modified smart contracts. Some additional thought should be put into how to protect voters from this vulnerability. The upside is, that these transactions can still be traced on the blockchain and stolen tokens can be rendered invalid, just like it would happen if a paper voting pass is stolen. Thus, the integrity of the voting process wouldn't be compromised.

8.3.3 Voting with someone else his wallet

Another way to allow proxy-voting is to allow people to vote with the actual wallet of the warranter. This would need to be done with some kind of written statement that a person is authorized to proxy-vote. This system would mean no improvement in comparison to the current proxy-voting system. Also, depending on the implementation of the wallet/identity verification this could run into legislative issues. For example, if a wallet is recognized based on ID-card verification, as a proxy-voter, a person would need to bring the ID-card of the warranter with him, which is inconvenient for the warranter and also illegal [40]. A situation where a person can borrow the identifying markers of another person, and can basically act like he is another person, is not the right path to go down from a legal point of view.

8.4 Best system

Three options for implementations to allow for proxy-voting in digital voting passes were proposed. The third option, voting with someone else's wallet, is the worst of the three choices, because of legislative and fraud issues. There would be virtually no trustworthy check to see if the warranter has actually allowed the proxy-voter to vote or if the proxy-voter just stole or copied the needed accessories for verification. This would cause people to distrust the system and therefore it would be hard to get this system widely accepted.

Using regular transactions wouldn't be a proper replacement of the paper voting pass, cause of the mentioned weaknesses and problems. The option with specialized transaction functions or smart contracts expands on the regular transactions and by doing that, tackles the main issues with that option. The system does get more complex and thus has higher development cost, but since trust and precision are very important factors in a digital voting solution, this increase in development costs will be worth it. The added verification for proxy-voting and added simple revoking rights make this solution much better than regular transactions and the increased ease of use makes this an improvement to the current paper proxy-voting system. This option is the way to go for the digital voting pass system.

9 Device security

One of the arguments against digital voting is that the devices that are used, could be tampered with. To overcome this issue, a device that is used in the voting process must be secure. Two techniques that can be used to secure a device and app are anti-tampering and obfuscation. Besides this, static analysis tools can be used to improve security.

9.1 Anti-tampering

Anti-tampering techniques can be used to ensure that the app itself has not been altered. Three methods that can be used are [41]:

- Verifying app sign at runtime
- Verifying the installer
- Environment checks

The first method, verifying the app sign at runtime, uses the fact that each Android app from the PlayStore should be signed with the developers private key [42]. Verifying the installer is a method that retrieves the name of the installer (such as com.android.example.app) from Android and checks this with a hardcoded value. The last method is mainly used to test if a debugger is attached to the app.

SafetyNet is an API from Android which provides access to Google services. With these services it is possible to determine the safety of a device [43]. This API can thus be used to ensure that device itself has not been tampered with.

9.2 Obfuscation

Since all software created in this project is open-source, reverse engineering is not needed to comprehend how the app works. This means that code obfuscation techniques are useless, but if a decision is made to switch to a closed-source application, code obfuscation is an useful technique. Two well-known obfuscators for Android and Java are DexGuard[2] and SafeGaurd[3].

9.3 Tools

The use of security static analysis tools can improve security of application by detecting missed faults of the developers. These static analysis tools can find for example faults where *java.util.Random* is used instead of the more secure *java.security.SecureRandom*, or hardcoding private keys in the source code. Examples of tools that can be used are Find Security Bugs and QARK.

[2] `https://www.guardsquare.com/en/dexguard`
[3] `https://www.guardsquare.com/en/proguard`

10 Glossary

Some jargon used in the Netherlands to describe items used in the voting process do not have a well-defined English translation. This short glossary defines the translations used in this report.

- ballot paper - stembiljet
- proxy-voter - gemachtigde
- proxy-voting - volmachtstemmen
- secret ballot - stemgeheim
- voting pass - stempas
- warranter - volmachtgever

References

[1] CBS. (2017) Bijna 13 miljoen kiesgerechtigden op 15 maart. Accessed: 2017-05-03. [Online]. Available: https://www.cbs.nl/nl-nl/nieuws/2017/07/bijna-13-miljoen-kiesgerechtigden-op-15-maart

[2] Rijksoverheid. Kan ik iemand machtigen om voor mij te stemmen bij verkiezingen? Accessed: 2017-05-03. [Online]. Available: https://www.rijksoverheid.nl/onderwerpen/verkiezingen/vraag-en-antwoord/iemand-machtigen-om-te-stemmen-bij-verkiezingen

[3] Rijksoverheid. Hoe kan ik stemmen bij verkiezingen? Accessed: 2017-05-03. [Online]. Available: https://www.rijksoverheid.nl/onderwerpen/verkiezingen/vraag-en-antwoord/hoe-en-waar-kan-ik-stemmen-bij-verkiezingen

[4] ACE. Punch cards. Accessed: 2017-05-09. [Online]. Available: https://aceproject.org/ace-en/topics/et/eth/eth02/eth02b/eth02b1

[5] V. Voting. Votamatic. Accessed: 2017-05-09. [Online]. Available: https://www.verifiedvoting.org/resources/voting-equipment/ess/votamatic/

[6] C. M. Yang. Presidency hinges on tiny bits of paper. Accessed: 2017-05-10. [Online]. Available: https://cseweb.ucsd.edu/~goguen/courses/275f00/abc-chads.html

[7] Kiesraad. Rood potlood en elektronisch stemmen. Accessed: 2017-05-10. [Online]. Available: https://www.kiesraad.nl/verkiezingen/inhoud/tweede-kamer/stemmen/rood-potlood-en-elektronisch-stemmen

[8] RTL. Zo werkt het softwaresysteem dat onze stemmen telt. Accessed: 2017-05-10. [Online]. Available: https://www.rtlnieuws.nl/nederland/politiek/zo-werkt-het-softwaresysteem-dat-onze-stemmen-telt

[9] R. Gibson, "Elections online: Assessing internet voting in light of the arizona democratic primary," *Political Science Quarterly*, vol. 116, no. 4, pp. 561–583, 2001. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0348238547&partnerID=40&md5=fd266dc88d06b5450e0a0a6d57f8ca46

[10] D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcat, H. Hursti, M. MacAlpine, and J. Halderman, "Security analysis of the estonian internet voting system," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 703–715. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84910610575&doi=10.1145%2f2660267.2660315&partnerID=40&md5=f4325abbf47127701145b5df2409bc1f

[11] L. Mui, M. Mohtashemi, and A. Halberstadt. Accessed: 2017-05-02. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.127.7256&rep=rep1&type=pdf

[12] (2011) Foundation wij vertrouwen stem computers niet. Accessed: 2017-05-02. [Online]. Available: http://wijvertrouwenstemcomputersniet.nl/

[13] D. Stokmans, "Overheid was naïef met stemcomputer," *NRC Handelsblad*, p. 2, 18 April 2007.

[14] R. webwereld. (2007) Stemcomputers afgeschaft, actiegroep blij. Accessed: 2017-05-02. [Online]. Available: http://webwereld.nl/overheid/35564-stemcomputers-afgeschaft--actiegroep-blij

[15] R. Gonggrijp and W.-J. Hengeveld. (2006) Nedap/groenendaal es3b voting computer, a security analysis. Accessed: 2017-05-02. [Online]. Available: http://wijvertrouwenstemcomputersniet.nl/images/9/91/Es3b-en.pdf

[16] Kiesraad. Accessed: 2017-05-02. [Online]. Available: https://www.kiesraad.nl/verkiezingen/inhoud/tweede-kamer/uitslagen/stemmen-tellen

[17] T. Kamer. 20ste vergadering 2e kamer. Accessed: 2017-05-02. [Online]. Available: http://wijvertrouwenstemcomputersniet.nl/images/b/be/HAN8061A02.pdf

[18] Y. Cai and D. Zhu, "Fraud detections for online businesses: a perspective from blockchain technology," *Financial Innovation*, vol. 2, no. 1, p. 20, 2016.

[19] Counterparty. Accessed: 2017-05-03. [Online]. Available: https://counterparty.io/

[20] MultiChain. Multichain. Accessed: 2017-05-09. [Online]. Available: http://www.multichain.com/

[21] OpenChain. Openchain. Accessed: 2017-05-09. [Online]. Available: https://www.openchain.org/

[22] Tribler. Multichain. Accessed: 2017-05-06. [Online]. Available: https://github.com/Tribler/tribler/wiki/Multichain-specifications

[23] A. Norta, *Creation of Smart-Contracting Collaborations for Decentralized Autonomous Organizations*. Cham: Springer International Publishing, 2015, pp. 3–17. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-21915-8_1

[24] Ethereum. Creating a cryptocurrency contract in ethereum. Accessed: 2017-05-02. [Online]. Available: https://www.ethereum.org/token

[25] OpenChain. Anchoring and ledger integrity. Accessed: 2017-05-09. [Online]. Available: https://docs.openchain.org/en/latest/general/anchoring.html#anchoring

[26] H. Marteau. Multichainjavaapi. Accessed: 2017-05-09. [Online]. Available: https://github.com/SimplyUb/MultiChainJavaAPI

[27] E. devs. (2014) A next-generation smart contract and decentralized application platform. Accessed: 2017-05-09. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[28] G. Wood. (2014) Ethereum: A secure decentralised generalised transaction ledger. [Online]. Available: http://gavwood.com/paper.pdf

[29] D. De Cock, C. Wolf, and B. Preneel, "The belgian electronic identity card (overview)." in *Sicherheit*, vol. 77, 2006, pp. 298–301.

[30] RVIG. Kenmerkenfolder reisdocumenten 2011. Accessed: 2017-05-10. [Online]. Available: https://www.rvig.nl/binaries/rvig/documenten/brochures/2011/11/02/kenmerkenfolder-2011/kenmerkenfolder-2011.pdf

[31] Statista. Forecast installed base of nfc-enabled phones worldwide from 2013 to 2018. Accessed: 2017-05-10. [Online]. Available: https://www.statista.com/statistics/347315/nfc-enabled-phone-installed-base/

[32] ICAO. (2015) Machine readable travel documents. [Online]. Available: http://www.icao.int/publications/Documents/9303_p9_cons_en.pdf

[33] B. Group. (2015) Block size increase. [Online]. Available: https://bravenewcoin.com/assets/Whitepapers/block-size-1.1.1.pdf

[34] Z. Říha, V. Matyáš, and P. Švenda, "Electronic passports," *Sborník příspěvků*, p. 5, 2008.

[35] N. Elenkov. Accessing the embedded secure element in android 4.x. Accessed: 2017-05-08. [Online]. Available: https://nelenkov.blogspot.nl/2012/08/accessing-embedded-secure-element-in.html

[36] (1989) Kieswet artikel l 14. Accessed: 2017-05-02. [Online]. Available: http://wetten.overheid.nl/BWBR0004627/2017-04-01#AfdelingII_HoofdstukL_Paragraaf3

[37] (1989) Kieswet artikel z 4&8. Accessed: 2017-05-02. [Online]. Available: http://wetten.overheid.nl/BWBR0004627/2017-04-01#AfdelingVI

[38] (1989) Kieswet artikel j 7a. Accessed: 2017-05-02. [Online]. Available: http://wetten.overheid.nl/BWBR0004627/2017-04-01#AfdelingII_HoofdstukJ_Paragraaf2

[39] S. Nakamoto. (2008) Bitcoin: A peer-to-peer electronic cash system. [Online]. Available: https://www.bitcoin.org/bitcoin.pdf

[40] (1881) Wetboek van strafrecht artikel 447b. Accessed: 2017-05-02. [Online]. Available: http://wetten.overheid.nl/BWBR0001854/2017-03-01#BoekDerde_TiteldeelIII

[41] S. Alexander-Bown. Android security: Adding tampering detection to your app. Accessed: 2017-05-02. [Online]. Available: https://www.airpair.com/android/posts/adding-tampering-detection-to-your-android-app

[42] Android. Sign your app. Accessed: 2017-05-02. [Online]. Available: https://developer.android.com/studio/publish/app-signing.html

[43] Android. Safetynet. Accessed: 2017-05-02. [Online]. Available: https://developers.google.com/android/reference/com/google/android/gms/safetynet/SafetyNet