Net Voting

J O A K I M G U S T A V S S O N

KTH Computer Science
and Communication

Bachelor of Science Thesis
Stockholm, Sweden 2012

Net Voting

J O A K I M G U S T A V S S O N

DD143X, Bachelor's Thesis in Computer Science (15 ECTS credits)
Degree Progr. in Computer Science and Engineering 300 credits
Royal Institute of Technology year 2012
Supervisor at CSC was Henrik Eriksson
Examiner was Mårten Björkman

URL: www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2012/gustavsson_joakim_K12030.pdf

Royal Institute of Technology
School of Computer Science and Communication

KTH CSC
SE-100 44 Stockholm, Sweden

URL: www.kth.se/csc

Abstract

The aim of this essay is to provide a complete system for conducting elections over the internet. The classical urn-based election model is examined and a set of scurity parameters identified that need to be replicated in the net-based election scheme in order to provide the same degree of security and anonymity as the classical model. Two main focus points are brought up: that of being able to securely verify one's identity over the internet, a core issue for conducting a fair election, and that of retaining voter anonymity within the system, in order to guarantee that no one will be able to associate a particular vote with a particular voter. These seemingly conflicting requirements are examined, and a solution is proposed based around a so called *Mixnet* as suggested by Park, Itoh and Kurosawa[1]. In order for voter identity to be verified a template for a national infrastructure for public key cryptography is suggested based on work by Wang and Liu[2]. Mixnet is integrated together with a cluster of vote storage servers and a voting application installed locally on the computers of every voter in order to create a complete system for net-based voting. The system is focused around heavy redundancy in order to be able to resist attempts against internal tampering. The final system is analyzed for security and ease-of-use against the only internet-based voting system currently being used internationally, namely the Estonian e-voting system, as presented in a master's thesis by Mägi[3]. The proposed system is deemed to be more secure against internal tampering than the Estonian system, but also deemed significantly more complex and harder to implement.

Referat

Nätröstning

Målet med denna uppsats är att tillhandahålla ett komplett system för genomförandet av röstning via internet. Den klassiska urnbaserade valmodellen studeras och en grupp säkerhetsparametrar, som behöver återskapas i den internetbaserade lösningen för att behålla samma nivå av säkerhet och anonymitet som den urnbaserade modellen, identifieras. Två kärnproblem tas upp: problemet med att säkert kunna bekräfta någons identitet över internet, ett viktigt problem för att kunna garantera valsäkerhet, samt problemet med att bevara de röstandes anonymitet i systemet, ett krav för att kunna garantera att ingen given röst kan associeras med en given röstande. Dessa två till synes motsägelsefulla problem behandlas, och en lösning baserad på det s.k. *Mixnet*, så som framlagt av Park, Itoh och Kurosawa[1], föreslås. För att en röstandes identitet skall kunna bekräftas föreslås en nationell infrastruktur för skapandet av asymmetriska krypteringsnyckelpar baserad på forskningsresultat av Wang och Liu[2]. Mixnet integreras tillsammans med ett kluster av röstlagringsservrar och ett röstningsprogram som installeras lokalt på de röstandes datorer för att skapa ett komplett röstningssystem för internet. Systemet har starkt fokus på redundans för att öka resistansen mot internt fusk. Det resulterande systemet analyseras med avseende på säkerhet samt användarvänlighet i förhållande till det enda internetbaserade röstningssystem som används internationellt, nämligen det estländska e-röstningssystemet, så som beskrivet i ett mastersarbete av Mägi[3]. Det föreslagna röstningssystemet bedöms som säkrare mot internt fusk än det estländska, men bedöms också som varande avsevärt mer komplext och svårare att implementera.

Contents

Figures

Chapter 1

Introduction

1.1 Background

Technology has come a long way in our modern society. We trade stocks online, order food and groceries online and use the internet for sharing our social experiences. Yet in one particular area the reality is far behind what it could have been, and this area is our elections. During every election the process of getting people to go to election places, the counting of the votes and the final anouncement involves a staggering amount of work.

One way to get rid of this whole process would be to use modern technology to allow the voters to vote from home, using the internet. So far very few entities have been able to realize such a system in practice as there have been significant security and integrity flaws with previously proposed solutions. This essay will attempt to identify the main issues of previously suggested schemes, as well as suggest possible solutions and propose a secure voting system solution for the internet.

1.2 Problem statement

1.2.1 Identity

The first major problem we encounter when trying to design an election scheme is that of making the system able to accurately identify and authenticate its voters. The system needs to be able to guarantee that the submitted votes have originated from the intended group of people. What this means is that some manner of authentication scheme for the voters will need to be established, so that a person can not falsely submit a vote under the identity of someone else. The scheme also needs to take this one step further, in such a way that the system is resistant to tampering of votes done by the election officials themselves, after the votes have been submitted during the initial stage of the election. The problem statement follows:

> How should an election system be designed in order to guarantee that votes can not be submitted under a false identity? How should the

internal workings of the system, namely the authentication process as well as the vote counting, be designed in order for the system to be resistant to internal tampering by election officials?

1.2.2 Anonymity

The second major problem we encounter is a direct result of the first problem. How can we make sure a vote was submitted by a given person without disclosing the identity of that person? This may sound impossible at first; in order to verify that a vote originates from a given person, that vote would need to be tagged in some way by a tag unique to the person who claims o have submitted the vote. However such a tag would also disclose the identity of that person to anyone that can view the votes. This dilemma needs to be solved in order to provide an election scheme that maintains the integrity of the voters. The problem can be defined as:

> How should an election system be designed in order to maintain the anonymity of its voters, while still being able to insure that the criteria mentioned in section 1.2.1 are met?

1.3 Source material

1.3.1 Park, Itoh and Kurosawa

In their article, Park, Itoh and Kurosawa[1] propose an algorithm for implementing an anonymity mixer known as *Mixnet*. The purpose of Mixnet is to handle the aquisition of votes, and verify that no tampering has occured to the votes, without revealing the contents of the vote. The basic idea behind a Mixnet is to use a number of mixer-nodes, that will receive an encrypted version of the vote, encapsuled with a verification code. The mixer nodes will then independently open the verification code of the vote. If the verification code differs on any of the servers the election will be called off due to tampering. Once all votes have been verified for authenticity, the votes are submitted to the vote-counter.

1.3.2 Wang and Liu

In their essay *A Practical Key Issuing Scheme in Identity-based Cryptosystem*, Wang and Liu[2] propose a scheme for creating authority-signed private keys for users based on a one-time physical authentication at an authority office. In a normal setting the private key generator will by generating the key also gain knowledge of that key, which causes a security concern. In this scheme the key is partially generated by multiple key generators, and distributed over multiple databases. A user would need to enter an authority office once in order to verify her identity using a physical media such as an ID-card. Partial key fragments will then be generated by numerous private key generators and stored independently. When a user needs to access her private key, she will need to query numerous key servers, which will yield parts of

the key using a blidning technique. This insures that the user can receive her key at any given time, without the authority being able to use her full private key in order to compromise security.

1.3.3 Mägi

In a master's thesis, Mägi[3] analyzes the security of the Estonian voting system, as well as the Secure Electronic Registration and Voting Experiment(SERVE) run in America. The security of the systems are analyzed in comparison to the traditional voting system with sheets of paper and a figurative urn. Mägi claims that the traditional voting system has to be considered secure, and that any e-voting system would need to be at least equally secure in order to be considered. The analysis is based around society and human characteristics, with the assumption that an atacker would attack the system only to affect the outcome of the vote. Using these characteristics, Mägi deems the Estonian voting system sufficiently secure, while concluding that the SERVE project has several vulnerabilities in the design of the system which could be exploited.

Mägi finishes the thesis with the conclusion that the approach taken uses human behaviour and society patterns as a basis for the claims of security; an approach that could be deemed disputable. In order to make a more conclusive security estimate further study into the workings of society and of human nature with respect to voting would need to be conducted.

Chapter 2

Proof of identity

2.1 The physical situation

In order to be able to evaluate a potential scheme for internet-based voting, we must first look at how voting for national elections are currently conducted. In order to establish what a physical election model might look like we will use the Swedish election model. Using this particular national model will not cause any loss of generality as the Swedish model closely resembles the classic urn-based election model.

Typical procedure Before the election any individual that fulfills the criteria set to participate in the election will receive a document in the mail indicating which parts of the election he is allowed to cast votes in. The document will also contain the location of the local Voting Office (VO), to which he has to travel to cast his vote. When the election day arrives he must travel to the VO specified on the document he received. Once there he will have to show the document to an election official that will hand him a unique envelope for each of the elections in which he can cast his vote. He will then step in behind a protective screen that effectively hides which vote-cards he puts in the envelopes. Once the envelopes are sealed he will show his voting document, his personal ID-card, as well as his envelopes to an election official. The election official will verify that the voter is who he claims to be and that he is allowed to cast his vote in the elections for which he has envelopes. The election official will call the information as he reads it, and another official will double-check everything against a secondary manifest to verify that the first official is not altering any information. Once both officials have confirmed the necessary information, each envelope containing a vote is deposited into an urn.

Baseline The physical election scheme is considered to be sufficiently secure to use for elections of parliment. In order to guarantee that a potential net-based voting scheme will be secure enough to use in a practical situation, it will need to be at least as secure as the physical model.

After a thorough look at the physical voting procedure described in the previous paragraph, we can identify a number of security parameters that are in place, that we need to reproduce in the net-based voting scheme:

1. The voter is only able to cast his vote in elections in which he is eligible to vote.
2. The voter is not able to cast a second vote after initial submission.
3. The submission, and information related to the submission, of a vote is verified by two election officials.
4. The contents of the vote are secret for anyone but the voter.
5. There is no way to associate a particular vote with the voter who cast it.

The first three parameters are related to the first problem we posed in this essay, while the last two parameters are related to the second problem. For the remainder of this chapter we will focus on the first problem, and then cover the second problem in the following chapter.

2.2 Online identification

The first problem we face when trying to devise a net-based voting system is how to properly prove our identity online. Looking at the Estonian net-voting scheme presented by Mägi[4], this problem is solved by using an already existing national infrastructure for electronic identification. The aim of the net-voting scheme presented in this essay is to be general enough to be deployed anywhere, and we can as such not assume that a similar infrastructure will be in place.

2.2.1 The extent of trust

The core of the problem of proving one's identity is how strong the trust is between the party verifying the identity and the party issuing the proof of identity. In the case of a physical identity card, the trust between the parties tends to be seen as strong due to the issuing party being a governmental agency and due to inherent difficulty of forging a physical identity card. When looking at the naïve way of online authentication, namely a username-password scheme, the strength of trust tends to be severely weaker.

Let us assume that a username and password would be used as authentication and proof of identity in an election scheme. The agency responsible for determining who is eligible to vote, and for sending out the documents related to the election, would send out a document containing a username-password pair that the voter would then use to authenticate on a website where he can cast his vote. This system could easily be manipulated by the election official sending out the username-password documents, through copying the information and using it for personal

gain. The system could also be compromised by the carrier of the authentication tokens, in this case the national post service, either through the loss of the documents containing the usernames and passwords, or through a postal worker opening the letters in transit, copying the information, and then resealing the letters.

2.2.2 Multi-level authentication

In order to prevent tampering with online identity tokens, multiple levels of authentication need to be in place. It is not possible to create an authentication scheme that is fully resistant to tampering without requiring a user to physically identify himself at one point during the authentication process. If identification tokens are sent to a user by means of a carrier (such as the postal service or through electronic transfer protocols such as email) there is no guarantee that the carrier will not have been compromised. The risk of this being the case could be reduced by using several independent carriers, where the full identity token is only partially transmitted by each carrier, and then reassembled by the voter. This technique is known as *blinding*. The problem with using a blinding technique is that the original sender will have access to the entire, unblinded, identity token before splitting it up so as to blind it for the carriers. For our election scheme this would result in the election being compromised if the election official responsible for sending out identity tokens is compromised, even if all the carriers are honest.

2.2.3 Digital signatures

Even assuming that the authentication process is tamper-proof, there is still the matter of being able to associate a particular vote with a particular voter. In order for the system to be able to prove to a voter that his vote has been taken into account, there needs to be some way to associate a vote with a particular voter, without disclosing the contents of the vote to the system. This would mean that the vote would need to be encrypted in order to prevent disclosure of content, and to be digitally signed by the voter in order to prove to him that his vote has been taken into account. If no proof is handed off to the voter then there is no way to prove to him that the submission system is not compromised, and that his vote is not just simply discarded as soon as he submits it. Through the use of a digital signature, the voter can see that his vote is still present in the system.

2.3 A complete authentication system

I will now propose a complete procedure for generating identity tokens, as well as using these tokens to prove the identity of the voter to the election system, and then finally use a part of the identity token to digitally sign the vote submitted by the voter. The procedure is based around a simple two-step authentication, where the first step is a user-password authentication and the second step is a public key cryptography solution, using physical identification tokens, as proposed

by Wang and Liu[2]. The system resembles the Estonian election scheme as described by Mägi[4], except that this system also provides a basic electronic identification authority that is meant to be easy to integrate into any sort of existing physical identification infrastructure.

Pre-step. A number of Authentication Service Offices (ASOs) will need to be established. These could easily be incorporated into any existing identity token handler, such as police stations or banks. After this, a set number of key privacy authority servers (KPAs) need to be set up. These servers should be physically sepparated to prevent tampering by the server host.

Step 1. The election authority will generate a username-password pair for every eligible voter in the system. These pairs will be sent out to the respective voter through the use of the national post service.

Step 2. Every voter will need to visit a local ASO in order to create a key-pair used for digital signatures. The key-pair is created as described by Wang and Liu[2]. The voter will need to provide a physical identification token, such as an ID-card or a passport, during this step. The partial private keys are sent out to the KPAs, which should be in place after the pre-step. The voter will after this step posses another security password used to query the KPAs in order to obtain his private key. This step will only need to be completed once per voter, and the same key-pair can be used for numerous elections.

Step 3. When the election starts the voter will use the username-password pair, that he was sent by mail, to authenticate to the voting application. Once he has been successfully authenticated the system will ask for the security password used to obtain the voter's private key from the KPAs. Once this key has been entered, the user will be authenticated to the system and be in possession of his private key.

Step 4. The user will cast his vote, which he then splits into two parts, v_1 and v_2, such that $v = v_1 \oplus v_2$. Each partial vote is then encrypted together with a security number using the public key of the vote counting server. This will prevent the vote storage servers from knowing the contents of the vote. The voter will then sign the vote-tuple using his private key. After this the vote is successfully cast. The voter will be able to complete steps 3-4 again should he change his mind, and will then be able to change the contents of his vote. This can be done until the election closes.

2.4 Discussion

Resistance to tampering The authentication scheme presented above ensures that half of the identification token used to vote is blinded from any given party, except for the voter himself. This is done by sending half of the token via regular

mail, and using the Authentication Service Office to generate the other half of the token. Ultimately this means that no individual party can assume the identity of a voter without involving several other parties. For the full token to be reassembled an attacker would need access to the username-password pair sent in the mail, as well as access to at least k of the key privacy authority servers used to store the partial private keys. The security parameter k can be set as needed. Even with a security factor of $k = 1$, two parties would need to work together in order to compromise the result of the election, which is the same number of parties needed to compromise the classic urn-model. This can be scaled up as needed. Using this scheme would as such provide at least as good security against internal tampering as the classic model.

Resistance to illegal voting In order for a vote to qualify it will have to be digitally signed by a voter's private key. This means that every vote in the system can be accounted for. If the system identifies two votes signed by the same key, the system can raise an error that tampering has occurred. The system of digital signatures can also allow a voter to change his vote before the closing of the election. If a vote signed by the voter's private key is found on the storage server, the system can query the voter on whether to replace the old vote with the new one, and if a positive response is given the old vote will simply be deleted and replaced by a new signed and encrypted vote.

The system itself will be able to determine which elections a voter is qualified to vote in by comparing an authenticated voter to a list containing which elections a given voter is allowed to vote in. The system will then only offer the voter to submit votes for elections he is qualified for. The list of qualified elections can be verified by the vote counting server at the end of the election, and any tampering to the list will be flagged and the election can be aborted.

Drawbacks The main drawback of the proposed system is the fact that a voter still has to make a physical appearance at an Authentication Service Office (ASO) in order to utilize the system. One could argue that the biggest advantage of using a net-based voting system is that a voter does not have to leave his home in order to vote. This drawback has two mitigating factors. The first one is that a voter only needs to visit the ASO one time, and will after that have a private key set up for any future election. The issuing of the private key can be seen as sepparate to the actual election, in the same way as one would say that applying for a physical ID-card is not part of the election in the classical urn-model, eventhough the ID-card is required in order to be able to vote in that model. The second mitigating factor is that the private key infrastructure established in the essay could be used for any sort of electronic authentication and/or digital signing if the authority issuing the key is sufficiently trusted (which it ought to be if used for an election), the infrastructure is not limited purely to elections.

One might also argue that setting up an entire public key infrastructure is largely

unnecessary for a procedure that only occurs once every few years. In response to this I will claim that the existance of an established infrastructure for public key cryptography is essential to the success of any net-based election scheme. This infrastructure needs to be homogenous for the entire population that is eligible to vote. The simplest way to ensure this is to introduce a suggestion, or template, for how such an infrastructure could be established, which is what has been done in this essay. In countries with an already established infrastructure, such as Estonia, there is no need for a sepparate infrastructure for elections alone, however it can not be assumed that all countries will have this infrastructure set up, and in order to provide a generally applicable scheme this essay is forced to provide a solution where this is the case.

Chapter 3

Retention of anonymity

In order to understand the problem of anonymity, let us first recall the last two security parameters provided in a classical urn-model election scheme, as mentioned in the previous chapter:

- The contents of the vote are secret for anyone but the voter.
- There is no way to associate a particular vote with the voter who cast it.

It was hinted at in the last chapter that the contents of the vote could be kept secure by splitting the vote into two partial votes, and then encrypting the two parts using a public key. However the last chapter offered no way for a submitted vote to remain anonymous due to the fact that every vote is digitally signed in order to prove to the voter that the vote has been aquired. This dilemma will be the focus of this chapter.

3.1 The drawback of digital signatures

In order to guarantee voter anonymity the digital signatures somehow need to be stripped from the votes after the election has closed (in order to allow for changing one's vote), but before the contents of the votes are disclosed. The intuitive way to solve this would be to use a two-step verification of the votes. At first the encrypted votes are stored together with the digital signatures on a set of servers for the duration of the voting process. Once the election is closed the vote-storage servers will strip the digital signatures and forward the encrypted votes to the vote counting servers.

This process would ensure that the contents of the vote are not disclosed if any of the vote storage servers are compromised, which will in turn make sure that a potential re-election will not be biased by a portion of the votes being disclosed prematurely. It would also ensure that none of the digital signatures will be attached to the votes when the votes are later opened by the vote counting server, thus perserving perfect voter anonimity as required by the classical urn-model.

3.2 Vote counting using Mixnet

The problem of stripping the digital signatures is that we can no longer be certain that the contents of the votes are intact, i.e one of the vote storage servers could have stripped the digital signature and then exchanged the vote with one with its own content. This would be possible due to the contents of the vote being encrypted by a public key, which every participant of the election system will have access to. In order to compromise the integrity of a vote, a dishonest vote storage server would simply need to create a vote string v and calculate the two partial votes v_1 and v_2 as described earlier, then encrypt each part of the tuple using the public key of the system.

In order to prevent this we will use a scheme known as *Mixnet* as proposed by Park, Itoh and Kurosawa[1]. Mixnet will distribute the votes reported by the vote storage servers over a number of vote counting MIXes (server nodes implementing the Mixnet protocol). When the votes are being counted each MIX would randomly select one element from each vote tuple, and open the partial vote and security number therein contained by decrypting the tuple element using a shared private key between all the MIXes. The MIX would then verify that the security number is the same for all MIXes. If it is not that means that one or more MIXes have been compromised and the election will be aborted. However if all the security numbers check out, the second half of the vote would be decrypted. The votes can then be reassembled by XOR-ing the two partial votes together. The counting of the votes can be performed simultaneously on all the MIXes by increasing a counter every time a vote gets verified. The results of the election will be the joint, verified result submitted by all MIXes.

3.3 Discussion

The combination of using Mixnet with using a set of vote storage servers, for stripping digital signatures, should prove sufficient to secure voter anonymity. This will ensure that no server that knows the identity of a voter (a vote storage server) will be able to know the contents of the vote (which is only known to the MIXes). In a real world scenario this would correspond to the vote storage servers being election officials holding the ID-card of a voter in one hand and the sealed letter containing the vote in the other hand. The election official would then hand off only the letter to another official (the MIX equivalent) that would in turn open the letter without ever having seen the ID-card of the voter.

Internal tampering One thing that readers will quickly notice is the heavy redundancy present in the system; numerous independent servers are used for storing votes, and then numerous MIXes are used to calculate the results of the election. This is one of the major differences between the Estonian system and the one proposed in this essay; the Estonian system assumes that all internal servers are tamper-

proof and that all election officials are honest. This means that only one server is required for storing the votes, and only one server is required to count the votes. The system presented in this essay assumes that any given internal server could be compromised and that any election official can be dishonest (compromised in this sense refers to the server looking like it is playing by the rules to the outside world, while it in fact stores sensitive data past expiry or tries to modify data present in databases). Each step of the election scheme presented has a security factor; in the case of storage servers and MIXes this comes down to how many servers that are active in the system. In order for a single step of the election to be compromised, one would need to compromise *all* servers that are a part of that step. This means that as long as a single honest storage server and a single honest MIX is present in the system, it will be able to flag inconsistencies and call for the election to abort. This does not offer full protection against tampering, but an attacker would need to affect all servers in a given step of the election, a number that can be scaled as needed to provide sufficient security, in order to affect the election. This makes an internal attack exponentially more difficult.

Large number of servers One of the main concerns with using the approach of having numerous vote storage servers in addition to the servers hosting the MIXes is the large number of server computers needed to conduct the election. This can not be circumvented without losing the guarantee for voter anonymity. Let us for a moment assume that the same servers could be used to both store the votes and act as a MIX. This may sound riddiculous at first after all the trouble we went through to sepparate what knowledge the vote storage servers have and the knowledge the MIXes have. However we can exploit the fact that the MIXes will not necessarily know the shared private key used to decrypt the votes at the start of the vote counting. We can let let two MIX servers, none of which has access to the shared private key, act as storage servers as described earlier. Once the election ends the two servers will strip the signatures from the submitted votes and shuffle the order of the votes. Each server will then swap its list of votes with that of the other server. This may sound like it guarantees anonymity, but this is not the case.

If a given MIX is dishonest in such a way that it stores the digital signatures of the votes (as in retains the exact data it had access to when it acted as a storage server) it can easily use the private shared key once given out to decrypt the original saved votes, and match the contents with the digital signatures it has stored. This compromises voter anonymity. As such the storage servers and MIXes need to be sepparate servers.

A mitigating factor to this problem would be that relatively few MIXes are needed to guarantee the same level of integrity as the classical urn-model. We would need one MIX and one vote storage server for every person that double-checks a submitted vote in the classical model, this assumes a one-to-one relation between vote storage servers and MIXes (as in one vote storage server only submits its data to one MIX). This number of servers is independant of how many votes that

are being processed, i.e if five election officials are used to double-check the votes when counting, that would require 10 servers, however these servers would be able to serve the entire election system, due to the counting of votes being done in batch and the processing being quick, rather than just one particular district, which is the case in the classical model. The set of MIXes could be fed one batch of votes for every district sequentially, and could as such be used to calculate election results on a district-by-district basis just as easily as on a nation-wide basis. This is not the case for human vote counters as they can only process a small amount of votes in a timely manner, and one such group would as such be needed per district. This makes the seemingly large number of servers turn out to be relatively reasonable compared to the hundreds, if not more, human vote counters currently employed during urn-model elections.

Chapter 4

Putting it all together

For the final chapter of this essay we will combine the components devised in earlier chapters into a complete voting system. The chapter will be split into three parts, namely what needs to be done before the election can start (Pre-election), what will happen during the election (Election) and finally what will happen after the election (Post-election). For every section the administrative work with configuring the system will be explained, followed by us accompanying an imaginary first-time voter, Alice, through the election to see what she needs to do in order to be able to cast her vote.

4.1 Pre-election

Setting up In order for the election to be able to commence the Public Key Infrastructure will need to be in place. This would mean that Authentication Service Offices (ASOs) will need to have been established around the country, each with access to a number of key privacy authority servers (KPAs) that will be able to store the partial private keys of the voters. The number of KPAs can be adjusted as needed, but the recommendation in this essay would be to have at least four, where two servers can recreate the private key. This setup offers the same level of security as the ID-card control performed during the classical urn-model, while having two backup servers in case of forced downtime.

The second step would be to make sure that the vote storage servers and MIXes are in place and functional. An overview on how these servers can be set up is depicted below:

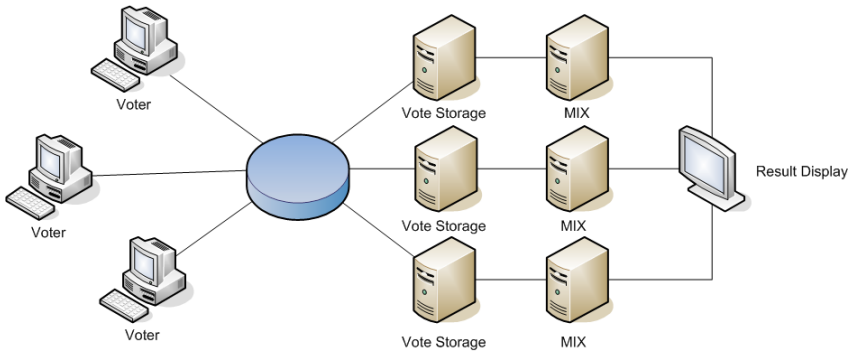

Figure 4.1. Vote Storage Server- and MIX configuration

Every voter will have a tiny application installed on their computer that will be used to submit votes. This application will need to be run locally on the voter's computer, meaning that an ActiveX- or Java-application would be recommended. This application will be able to simultaneously connect to all the vote storage servers over connections secured using Secure Socket Layer (SSL). Each vote storage server will be connected to a single MIX, resulting in a 1:1 ratio between vote storage servers and MIXes. The MIXes will be connected to some form of output media used to display the results of the election; in the figure above that media happens to be a display, but could easily be a printer or even another sepparate system used for broadcasting and/or processing the results of the election.

Finally a database containing username/password pairs will need to be generated, and copied to every vote storage server. This database will server as authentication when voters connect to the vote storage servers in order to submit their votes. The username/password pair belonging to a particular voter will also need to be sent out to the voter through regular mail.

Alice A few weeks before the election would start Alice received a letter containing a username and a password, as well as instructions on how to download the voting application, and what the username and password will be used for. The letter also states that since this is Alice' first time voting using the new system, she needs to visit a local Authentication Service Office. The addresses of the closest few are included in the letter. Later during the week Alice stops by the local police station on her way home from work, one of the locations listed as providing the Authentication Service. Once there she is asked to show her ID-card to the clerk, and receives a sheet of paper listing the password of her new private key, as well as instructions on how this key will be used during the election. Alice is now ready to vote, and returns home pondering which party she should cast her vote on.

4.2 Election

Setting up The hardware for the net-voting system will already be in place after the pre-election step. As the election commences all of the vote storage servers are now turned on, and instructed to receive connections from voters. The MIXes are not needed during this step and will remain offline. The voter applications will be configured to connect to all of the vote storage servers, and authenticate as per the diagram below:

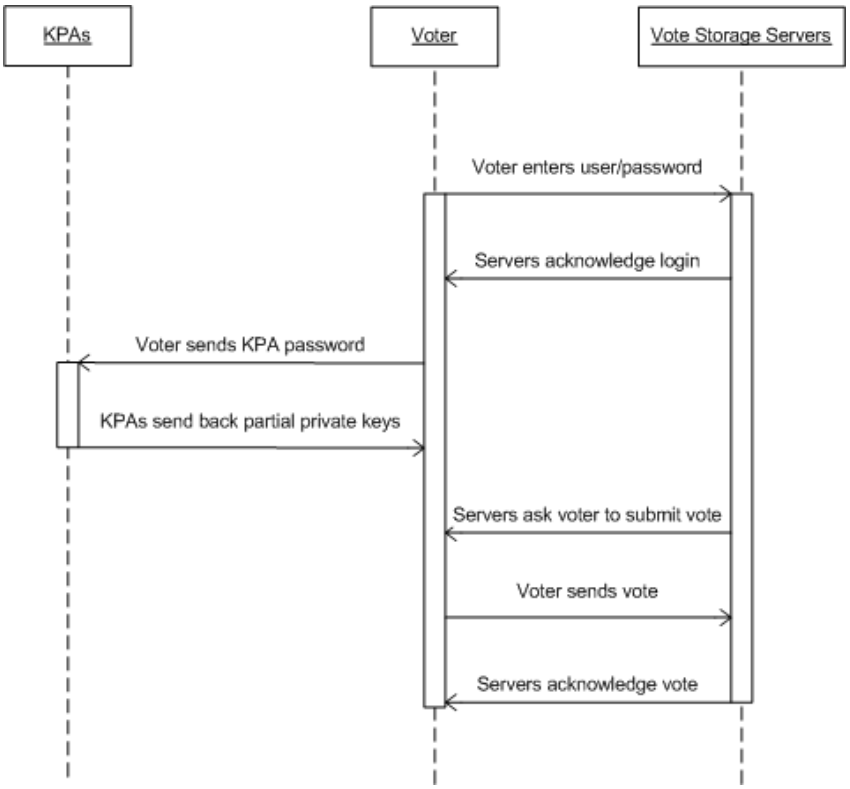

Figure 4.2. Voter application authentication procedure

As we can see in the figure above, the voter will begin the authorization procedure by submitting the username and password that she was sent during through regular mail. If the information submitted by the user checks out when compared to the database present on the vote storage servers the voter can proceed to the next step. If any of the vote storage servers can not find the username/password pair in its database while others can, it will raise an error and the election will be aborted. Once the voter has passed initial authentication the voter application will ask for the password associated with the private key of the voter. This password will when entered cause the voter application to connect to the key privacy authority servers (KPAs) in order to receive the partial private keys contained there. Once enough partial keys have been received, the full private key will be reassembled. The voter

will now use the private key to sign her vote and submit the vote to all vote storage servers, which will store the votes until the election has closed. If at any time the voter wants to change her vote, she can connect to the vote storage servers the same way as she just did, and then recast her vote.

Alice The day of the election opening has arrived and Alice is standing by at her home-computer with the letter she received in the mail, as well as the sheet of paper she received at the police station. She already installed the voter application a few weeks earlier. Once the election begins Alice starts up her voter application and approves the security warning about launching an ActiveX-application. She enters the username and password from the letter when prompted to by the application, and after a short time receives a message back that she is now connected to the servers and needs to create her private key in order to vote. She submits the password to her private key, and sees the message "Private key reconstructed, you are clear to vote" appear on her screen. She selects the party she decided earlier to vote for, and selects the members of that party that she would like to see in the parliment. She presses the Submit-button and is asked if she is sure about her choice. She presses "Yes", and the screen shows her which party she voted for and that her vote has now been submitted.

A few days later Alice decides that she would rather see another member of the party she voted for in the parliment, and once again starts her voter application and logs in. She once again selects the same party, but this time select another member of the party instead. She once again presses "Submit". She is told that there already is a vote in the system cast by her, and asked if she wants to replace her old vote with the new one. She selects "Yes", and logs out.

4.3 Post-election

Setting up Once the election closes the remaining step is to open the votes and count them. The MIXes are brought online and the vote storage servers are configured to no longer accept connections from voters. Every vote storage server goes through the list of signed votes stored on it and makes sure that all the other vote storage servers have the same digital signatures present. If the sets of digital signatures for each server are not identical, the vote storage servers raise an error and the election is aborted. Once the signatures have been verified, each server strips the digital signatures from the encrypted votes and forward the list of encrypted votes to the MIX they are paired with. The MIXes will then jointly open half of the votes and verify their integrity. If at any point a compromised vote is detected the MIXes will raise an error and the election is aborted. If the votes check out, the other half of the votes is opened and the votes reassembled. Each MIX counts the contents of the votes and verifies this with all other MIXes. If the counts are not identical the MIXes raise an error and the election is aborted. If the counts are

identical, the MIXes forward the results to the display media, as shown in figure 4.1.

Alice There is not much to do for Alice once the elections have closed other than to eagerly wait to see if her party won the election, once the results are made public. She is now free to uninstall the voter application should she choose to, since it will not be used again until the next election. The username/password pair has expired, so she is safe to throw away that letter. The password for her private key is something she should save, since it can be used for other forms of electronic authentication, and will be needed for the next election. If she happens to lose the password she can create a new private key at a local Authentication Service Office, and thus receive a new password.

4.4 Conclusions

Setting up The first thing we notice by going through this chapter is that most of the setup, and thus manual work, has to do with installing the servers, and getting the public key infrastructure in place. The actual election is handled almost completely automatically. Since the infrastructure only needs to be set up once, it is very easy to conduct future elections using the same hardware setup; the servers need to have their harddrives wiped in order to get rid of any old election results, and the voter application and vote storage servers need to be configured slightly to allow for the parameters of a new election (parties, party representatives or actual question if it is a non-parliment election). This ease of use makes the system suitable for conducting any kind of elections, not just elections for parliment. If the voter application is made general enough, the agency hosting the net-based election servers can even let others use their infrastructure to conduct votes for entities such as company boards, various kinds of sporting events etc. This can all be done by one instance of the infrastructure nation-wide rather than every district needing their own infrastructure in addition to the national umbrella infrastructure, which is the case with urn-model elections.

The voter perspective As we can see from following Alice through the election there is not a whole lot she needs to do in order to cast her vote. The main roadblock is the creation of the private key, but as mentioned earlier this can be substituted for existing public key infrastructure, and even if no such infrastructure is in place, the step only needs to be completed once in order to provide a life-time key, that can in turn also be used for other forms of electronic identification e.g banking. The usage of the voter application would be the equivalent of paying your bills online in terms of effort, since both require the user to log in and then select what data to submit (in the election case the party to vote for and in the banking case what funds to transfer and how much). This low level of effort needed on behalf of the voter

would hopefully encourage voters to use this system in favor of the old urn-based model.

Security The focus of this essay has been to construct a secure system for net-based voting. Using the overview of the Estonian system provided by Mägi[4], we can see that the two systems are very similar. The main difference is the heavy focus on redundancy present in the system presented in this essay as opposed to the Estonian system. This makes the system equally secure to outside tampering, but more resistant to internal tampering. The drawback of this is the increased number of servers required to handle the election.

Comparing the system presented in this essay to the urn-model, we can see that our system offers answers to all the security parameters presented in Chapter 2 of this essay, as well as offering numerous scalable parameters that can be altered to increase resistance against internal tampering.

Resistance against voters It should be noted that the system offers no resistance against external tampering affecting the voter, a prime example of this would be the lack of a protective screen around the voter, protecting her against people looking over her shoulder when she votes. This kind of resistance would be difficult, if not impossible, to offer voters due to the environment in which the election takes place, that being the homes of the voters, being impossible for the election agency to control. This should be considered to be the main weakness of the system proposed in this essay, in addition to the natural drawback of it being somewhat complex due to the redundancy.

The Estonian system It should be noted that the Estonian system is more streamlined and requires a less complex setup in order to work. The major drawback of it, the reliance on existing public key infrastructure, can easily be circumvented, and the scheme for issuing public keys presented in this essay (based on work by Wang and Liu[2]) can easily be integrated into the Estonian voting system. The Estonian voting system is preferable if there is little to no concern that internal tampering will take place, due to the simplicity of the system making it easier to implement and use. However if the risk of tampering is to be considered moderate to high, the system presented in this essay will be preferable due to the stronger resistance.

Bibliography

[1] Park, Itoh and Kurosawa, *Efficient Anonymous Channel and All/Nothing Election Scheme*, T. Helleseth (Ed.): Advances in Cryptography - EUROCRYPT '93, LNCS 765, pp. 248-259, 1994, Springer-Verlag Berlin Heidelberg 1994

[2] Wang and Liu, *A Practical Key Issuing Scheme in Identity-Based Cryptosystem*, Computing, Communication, Control, and Management, 2008. CCCM '08. ISECS International Colloquium on , vol.1, pp.454-457, 3-4 Aug. 2008 doi: 10.1109/CCCM.2008.302

[3] Mägi, *Practical Security Analysis of E-voting Systems*, Master's Thesis, Tallinn University of Technology, Faculty of Information Technology, Department of Informatics, Chair of Information Security, 2007

[4] Ibid. pp. 21-22.