UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science
Informatics Curriculum

Joonas Lõmps

Applying Estonian Internet Voting Individual Verification System to Other Electoral Systems

Bachelor's Thesis (9 ECTS)

Supervisor: Sven Heiberg

Tartu 2015

Applying Estonian Internet Voting Individual Verification System to Other Electoral Systems

Abstract:

The current paper gives an overview of the Estonian internet voting individual verification system and introduces different ballot styles. It proposes and describes modifications to the Estonian system, so it could be used for individual verification with the introduced ballot styles and multiple elections.

Keywords:

Estonian internet voting, internet voting, individual verification

Eesti interneti hääletamise individuaalse verifitseeritavuse süsteemi ühildamine teiste valimissüsteemidega

Lühikokkuvõte:

Selles töös antakse ülevaade Eesti interneti hääletamise individuaalse verifitseeritavuse süsteemist ja tutvustatakse erinevaid valimissedeleid. Töös esitatakse ja selgitatakse muudatused Eesti süsteemile, et seda oleks võimalik kasutada tutvustatud valimissedelite ja mitme korraga käimas oleva valimise individuaalseks verifitseerimiseks.

Võtmesõnad:

Eesti interneti-hääletamine, interneti-hääletamine, individuaalne verifitseeritavus

Table of Contents

Introduction

Estonia has been using internet voting as a secondary way of letting their citizens vote in elections since 2005. Over the years the popularity of it has risen and in the 2015 elections for the Riigikogu 30.5% [1] of the votes were given over the internet. In the year 2013 individual verification was added to Estonia's internet voting system [2], via smart device. Every i-voter can use the National Election Committees' verification application to verify that their vote was sent to the vote storage server (VSS) with the correct choice.

Estonia's electoral systems ballot requires the voter to mark their most preferred choice in their voting district. Also there are no documented cases of having more than one election at the same time. The author of this thesis intends to find out whether a similar system with some modifications and assumptions for individual verification could be used for multiple elections at the same time and with more complex ballot styles.

The thesis consists of three parts. In the first part Estonia's internet voting and vote verification system is described. The second part of the thesis gives an overview of the different ballot styles considered. Third part describes the assumptions and modifications made for the protocol to support individual verification with multiple elections and more complex ballot style. A possible protocol is also described.

1. Estonian system

In this chapter the Estonian internet voting individual verification is explained which works under following assumptions:

1. The voter must have a computer with the voting application.
2. The voter must have the right to vote.
3. The voter must have an ID-card or Mobile-ID and the necessary PINs for authentication and signing.
4. The voter must have a smart device with the verification application, internet access and camera.
5. RSA-OAEP encryption system is used for encrypting the vote.
6. The vote must be successfully cast and the vote reference must be transferred to the verification application via QR code [3].
7. The verification application must know the voting application's public key.

1.1 Vote casting

As the thesis is interested in the verification of the voter's choice, it is assumed that the voter is successfully authenticated and the voting application has received, *Choices*, the list of choices for his or her district.

The voter makes his or her choice c and proceeds to sign with PIN. The application then adds padding to the choice c using RSA-OAEP [4] and randomness r. Then it encrypts the padded choice with the election public key and the encrypted vote is signed with the voter's private key. The encrypted and signed vote $Enc_{Sig}(c)$ is sent to the vote forwarding server. The vote forwarding server then sends the received vote to the vote storage server. The vote storage server assigns a unique token, *voteID*, to the received vote and sends it back to the vote forwarding server that forwards *voteID* to the voting application. The application displays a QR code containing r and *voteID*. Vote casting process can be seen on figure 1.

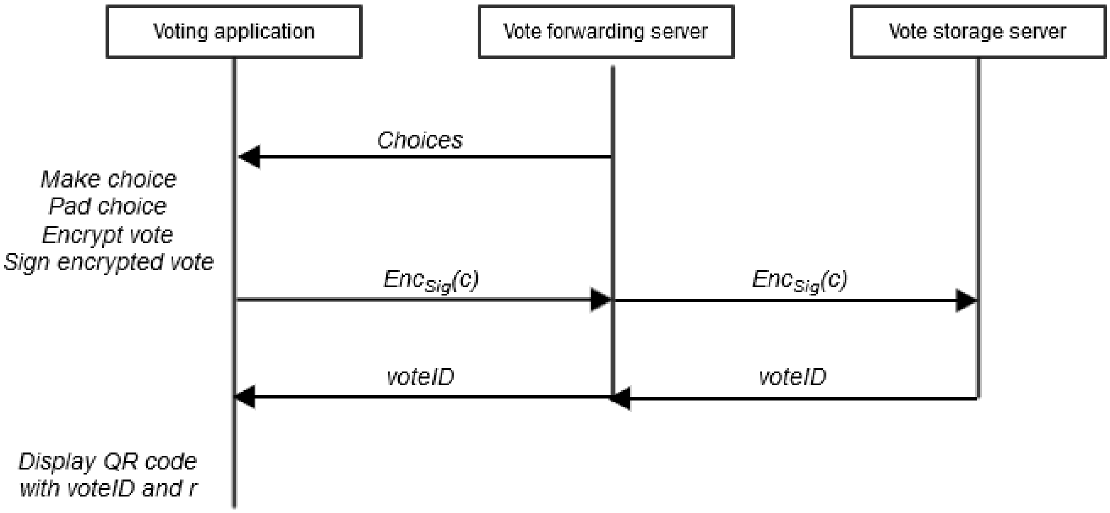


Figure 1. Vote casting process

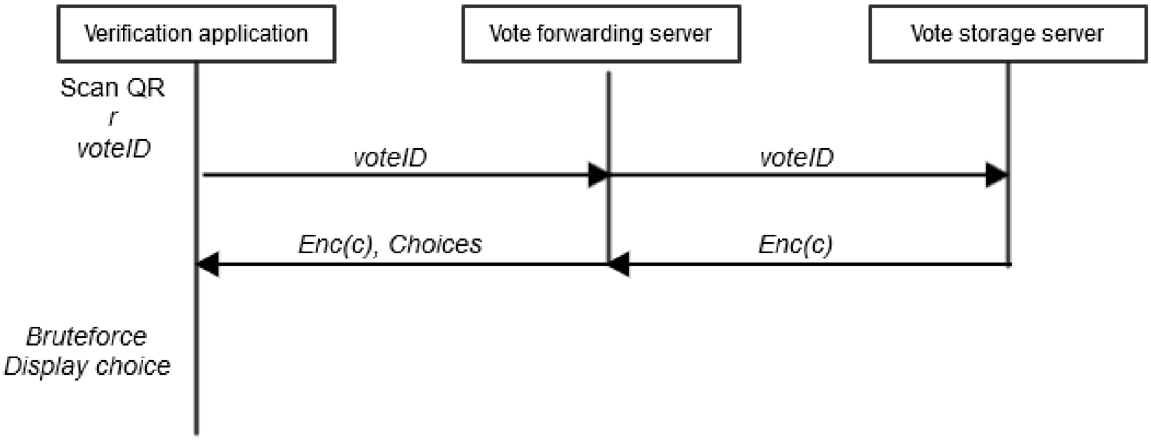


Figure 2. Vote verification process

1.2 Vote verification

The voter can confirm that his or her vote was recorded as cast using a smart device application provided by the National Electoral Committee. The voter uses the application to scan the QR code displayed in the voting application to get *r* and *voteID*. The latter is sent to the vote forwarding server, which returns the encrypted vote *Enc(c)* and the choice list *Choices*. It then uses the scanned randomness *r* to encrypt every possible choice and compares the result with the encrypted vote v received from the server. If they match, the corresponding choice is displayed and the voter checks it against his or her deliberate choice. Vote verification process can be seen on figure 2.

The voter can vote multiple times, but only the last vote will be counted. The voter can perform verification up to three times and up to 30 minutes after voting.

2. Electoral systems and ballot styles

An electoral system is the method used to translate the votes cast in an election into the number of seats acquired by individuals and parties. Reynolds and his colleagues [5] believe that the key elements of an electoral system are the electoral formula, the ballot structure and the district magnitude. First of them sets how the seat allocation is calculated, second states whether the voter votes for a party or a candidate and specifies the voter's choice possibilities. Third defines how many seats every district elects. Ballot structure is what the current thesis is interested in.

2.1 Simple ballot

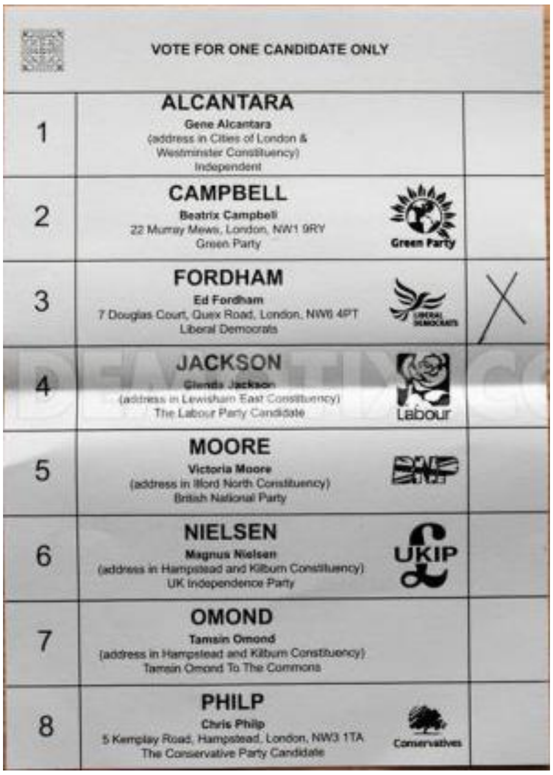
In case of a simple ballot the voter has a single choice. According to Reynolds et al. [5] simple ballot style is used in many different electoral systems such as first past the post, party block vote, the two-round system, list proportional representation and the single non-transferable vote. Example ballot can be seen in figure 3.

2.2 Multiple choice ballot

Electoral systems using multiple choice ballot style allow the voter to make multiple choices, but not unlimited choices. Choice limit is declared by the electoral system in use. According to Reynolds et al. [5] approval ballot style is used by the Limited Vote and Block Vote. Example ballot can be seen in figure 4.

2.3 Ranked ballot

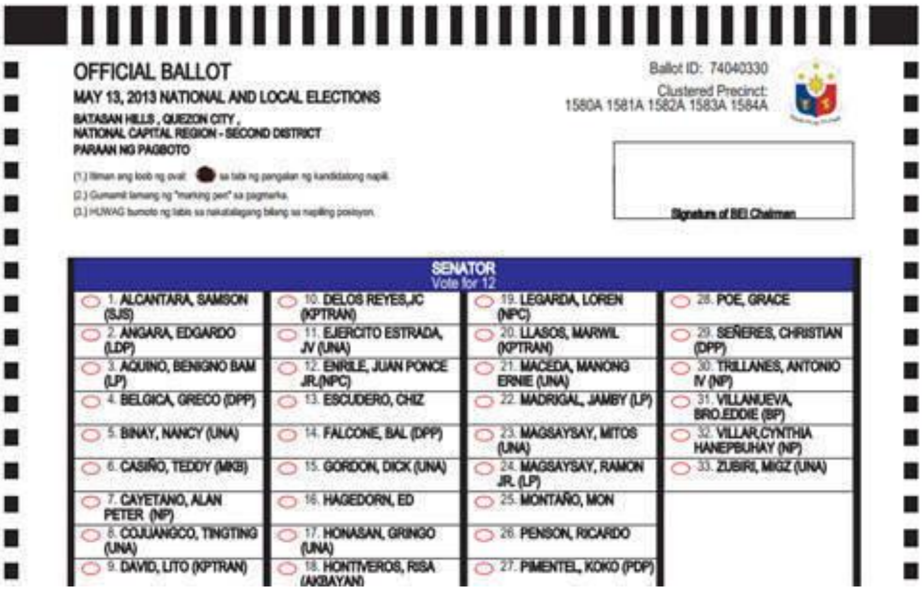
Ranked ballot requires the voter to rank the possible choices in order of preference. The preferences must be unique choices. Reynolds et al. [5] and Farrell [6] described this ballot style for electoral systems like alternative vote, the single transferable vote and Borda count. Example ballot can be seen in figure 5.



VOTE FOR ONE CANDIDATE ONLY

1	**ALCANTARA** Gene Alcantara (address in Cities of London & Westminster Constituency) Independent	
2	**CAMPBELL** Beatrix Campbell 22 Murray Mews, London, NW1 9RY Green Party Green Party	
3	**FORDHAM** Ed Fordham 7 Douglas Court, Quex Road, London, NW6 4PT Liberal Democrats LIBERAL DEMOCRATS	X
4	**JACKSON** Glenda Jackson (address in Lewisham East Constituency) The Labour Party Candidate Labour	
5	**MOORE** Victoria Moore (address in Ilford North Constituency) British National Party BNP	
6	**NIELSEN** Magnus Nielsen (address in Hampstead and Kilburn Constituency) UK Independence Party UKIP	
7	**OMOND** Tamsin Omond (address in Hampstead and Kilburn Constituency) Tamsin Omond To The Commons	
8	**PHILP** Chris Philp 5 Kemplay Road, Hampstead, London, NW3 1TA The Conservative Party Candidate Conservatives	

Figure 3. A ballot paper from the UK 2010 General Election listing 8 candidates for The Hampstead and Kilburn constituency. [7]



OFFICIAL BALLOT

MAY 13, 2013 NATIONAL AND LOCAL ELECTIONS

BATASAN HILLS , QUEZON CITY ,
NATIONAL CAPITAL REGION - SECOND DISTRICT
PARAAN NG PAGBOTO

(1.) Itiman ang loob ng oval ● sa tabi ng pangalan ng kandidatong napili.
(2.) Gumamit lamang ng "marking pen" sa pagmarka.
(3.) HUWAG bumoto ng labis sa nakatalagang bilang sa napiling posisyon.

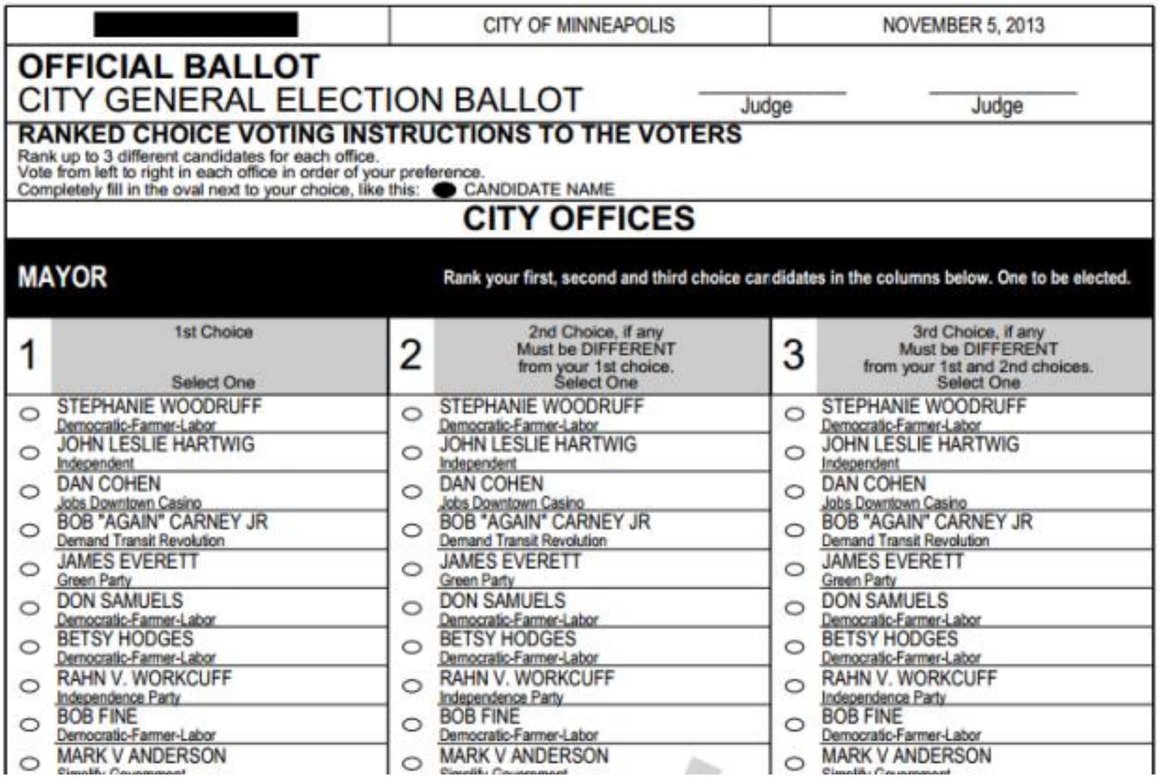
Ballot ID: 74040330
Clustered Precinct:
1580A 1581A 1582A 1583A 1584A

Signature of BEI Chairman

SENATOR
Vote for 12

○ 1. ALCANTARA, SAMSON (SJS)	○ 10. DELOS REYES,JC (KPTRAN)	○ 19. LEGARDA, LOREN (NPC)	○ 28. POE, GRACE
○ 2. ANGARA, EDGARDO (LDP)	○ 11. EJERCITO ESTRADA, JV (UNA)	○ 20. LLASOS, MARWIL (KPTRAN)	○ 29. SEÑERES, CHRISTIAN (DPP)
○ 3. AQUINO, BENIGNO BAM (LP)	○ 12. ENRILE, JUAN PONCE JR.(NPC)	○ 21. MACEDA, MANONG ERNIE (UNA)	○ 30. TRILLANES, ANTONIO IV (NP)
○ 4. BELGICA, GRECO (DPP)	○ 13. ESCUDERO, CHIZ	○ 22. MADRIGAL, JAMBY (LP)	○ 31. VILLANUEVA, BRO.EDDIE (BP)
○ 5. BINAY, NANCY (UNA)	○ 14. FALCONE, BAL (DPP)	○ 23. MAGSAYSAY, MITOS (UNA)	○ 32. VILLAR,CYNTHIA HANEPBUHAY (NP)
○ 6. CASIÑO, TEDDY (MKB)	○ 15. GORDON, DICK (UNA)	○ 24. MAGSAYSAY, RAMON JR. (LP)	○ 33. ZUBIRI, MIGZ (UNA)
○ 7. CAYETANO, ALAN PETER (NP)	○ 16. HAGEDORN, ED	○ 25. MONTAÑO, MON	
○ 8. COJUANGCO, TINGTING (UNA)	○ 17. HONASAN, GRINGO (UNA)	○ 26. PENSON, RICARDO	
○ 9. DAVID, LITO (KPTRAN)	○ 18. HONTIVEROS, RISA (AKBAYAN)	○ 27. PIMENTEL, KOKO (PDP)	

Figure 4. Ballot Template for the 2013 National and Local Elections in the Philippines. [8]



[redacted]	CITY OF MINNEAPOLIS	NOVEMBER 5, 2013

OFFICIAL BALLOT
CITY GENERAL ELECTION BALLOT

Judge

Judge

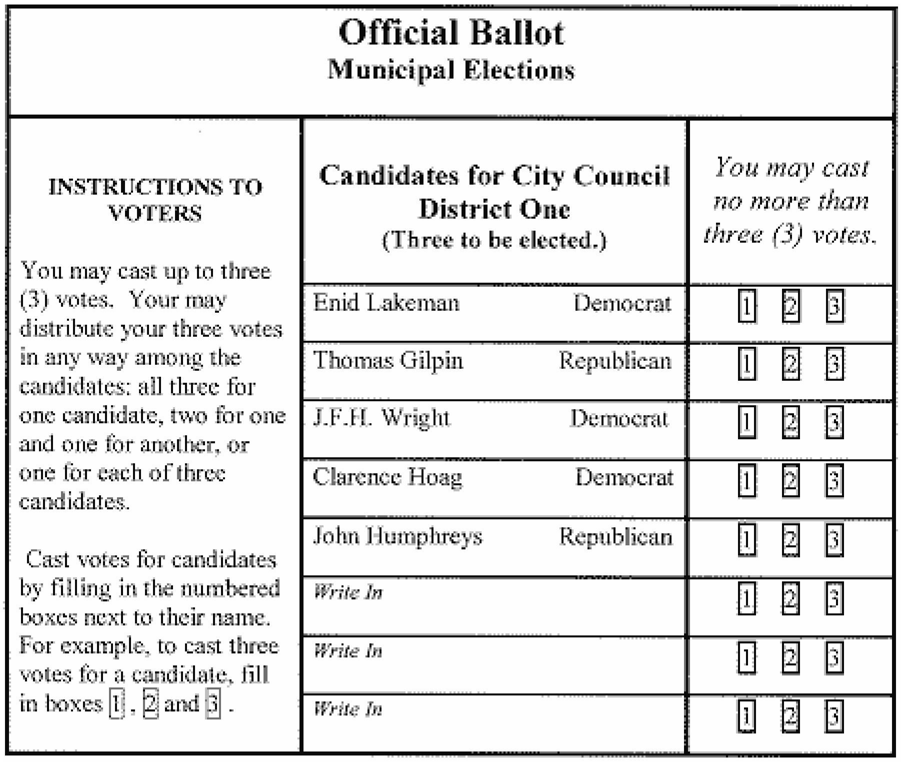
RANKED CHOICE VOTING INSTRUCTIONS TO THE VOTERS
Rank up to 3 different candidates for each office.
Vote from left to right in each office in order of your preference.
Completely fill in the oval next to your choice, like this: ● CANDIDATE NAME

CITY OFFICES

MAYOR Rank your first, second and third choice candidates in the columns below. One to be elected.

1 1st Choice Select One	2 2nd Choice, if any Must be DIFFERENT from your 1st choice. Select One	3 3rd Choice, if any Must be DIFFERENT from your 1st and 2nd choices. Select One
○ STEPHANIE WOODRUFF Democratic-Farmer-Labor	○ STEPHANIE WOODRUFF Democratic-Farmer-Labor	○ STEPHANIE WOODRUFF Democratic-Farmer-Labor
○ JOHN LESLIE HARTWIG Independent	○ JOHN LESLIE HARTWIG Independent	○ JOHN LESLIE HARTWIG Independent
○ DAN COHEN Jobs Downtown Casino	○ DAN COHEN Jobs Downtown Casino	○ DAN COHEN Jobs Downtown Casino
○ BOB "AGAIN" CARNEY JR Demand Transit Revolution	○ BOB "AGAIN" CARNEY JR Demand Transit Revolution	○ BOB "AGAIN" CARNEY JR Demand Transit Revolution
○ JAMES EVERETT Green Party	○ JAMES EVERETT Green Party	○ JAMES EVERETT Green Party
○ DON SAMUELS Democratic-Farmer-Labor	○ DON SAMUELS Democratic-Farmer-Labor	○ DON SAMUELS Democratic-Farmer-Labor
○ BETSY HODGES Democratic-Farmer-Labor	○ BETSY HODGES Democratic-Farmer-Labor	○ BETSY HODGES Democratic-Farmer-Labor
○ RAHN V. WORKCUFF Independence Party	○ RAHN V. WORKCUFF Independence Party	○ RAHN V. WORKCUFF Independence Party
○ BOB FINE Democratic-Farmer-Labor	○ BOB FINE Democratic-Farmer-Labor	○ BOB FINE Democratic-Farmer-Labor
○ MARK V ANDERSON	○ MARK V ANDERSON	○ MARK V ANDERSON

Figure 5. Minneapolis sample ballot 2013 [9]



Official Ballot
Municipal Elections

INSTRUCTIONS TO VOTERS

You may cast up to three (3) votes. Your may distribute your three votes in any way among the candidates: all three for one candidate, two for one and one for another, or one for each of three candidates.

Cast votes for candidates by filling in the numbered boxes next to their name. For example, to cast three votes for a candidate, fill in boxes 1, 2 and 3.

Candidates for City Council District One (Three to be elected.)		*You may cast no more than three (3) votes.*
Enid Lakeman	Democrat	1 2 3
Thomas Gilpin	Republican	1 2 3
J.F.H. Wright	Democrat	1 2 3
Clarence Hoag	Democrat	1 2 3
John Humphreys	Republican	1 2 3
Write In		1 2 3
Write In		1 2 3
Write In		1 2 3

Figure 6. Cumulative vote ballot [10]

2.4 Cumulative ballot

Cumulative ballot style gives the voters a certain amount of votes. The voter can distribute those votes as desired - all on once choice or spread in any other pattern between the possible choices. According to Farrell [6] cumulative ballot style is used in the cumulative vote. Example ballot can be seen in figure 6.

3. Unified protocol

3.1 Standardizing ballots

The different ballot styles viewed in the second paragraph can be standardized quite a lot. All the mentioned ballot styles can be viewed as multiple choice ballot:

1. Simple ballot is a multiple choice ballot, where the voter can make a single choice.
2. Preferential order ballot is a multiple choice ballot, where the order of the choices is fixed.
3. Cumulative vote ballot is a multiple choice ballot, where the voter can choose the same choice multiple times.

3.2 Assumptions

In this chapter changes to the Estonian internet voting individual verification system needed for it to work with the described ballot styles are described and the assumptions are generalized.

1. The voter must have a computer with the voting application.
2. The voter must have the right to vote
3. The voter must have the means necessary for a successful authentication and signing.
4. The voter must have a smart device with the verification application, internet access and camera.
5. A probabilistic public key encryption system must be used for encrypting the vote.
6. The vote must be successfully cast and the vote reference must be transferred to the verification application via QR code.
7. The verification application must know the voting applications public key.

If a deterministic public key encryption would be used the cipher texts of the same choice would be same. Since there is a limited number of choices an eavesdropper can easily brute force the voters vote and the privacy of the vote would be broken. To protect the vote's privacy a probabilistic public key encryption must be used to encrypt the voter's choices. Public key encryption system is required so that the verification application could use the same public key and the provided randomness for the vote verification process.

It is required that the vote verification process should not take too much time – e.g. more than 10 seconds. To speed up the vote verification process every choice is encrypted separately. This means that the worst case scenario for an election with multiple choice or preferential order ballot where there is n choices and r choices to be made is $\sum_{i=0}^{r-1} n - i$ encryptions. Whereas if all the selected choices were to be encrypted together there could be up to $\mathrm{P(n, r)}$ encryptions, because the order of the choices made by the voter is not known. In case of a cumulative vote ballot the worst case scenarios would respectively be $\sum_{i=1}^{r} n$ and n^r. It is easy to see, that when encrypting all the choices together, there is an exponential growth in the number of encryptions needed to find the voter's choice.

It would be practical to encrypt the choices separately. For every encryption a new randomness is needed, otherwise an adversary could gain information about the plain texts and the vote's privacy would be broken.

The Estonian system uses random, with a size of 160 bits, that is put into the QR code in hex encoding which results in 40 characters per random. This means that in case of an election where the voter has to make 10 choices the length of the QR code would exceed 400 characters as the randoms are only a part of the data in the QR code. Older smart devices can have problems with scanning QR codes longer than 300 characters. This sets a limit to how much data can be put into the QR code. Depending on the encryption system used the length of the random varies, for example in ElGamal encryption system [12] the random is the same size as the key, which is suggested to be 2048 bits or 512 characters in hex encoding.

Instead of putting the randoms into the QR code it would be more efficient to generate one random seed for a randomness stretching function and use it in both, the voting and verification application. Randomness stretching function takes a seed and stretches it to given length and the results from the same seed and to same length are the same. With randomness stretching the amount of characters in the QR code could be reduced a large margin.

3.3 Suggested protocol

It can be assumed that the voter is successfully authenticated and is given valid tokens that are needed to send the vote to the vote forwarding server. In addition, only the parts that are relevant to the vote verification are mentioned.

After adding the introduced changes the following protocol is suggested. It is believed that it could be used for individual verification for most of the commonly used electoral system ballot styles.

3.3.1 Vote casting

Figure 7 shows the process for casting a vote. Voting application has received the choice list *Choices* from the choice list server. The voter makes 0 to n choices $C_1, ..., C_n$ where n is the maximum number of choices granted.

Choices are padded to 2048 bits. A random R is generated, this R is stretched with the SHA-256 [11] function to generate randoms $R_1, ..., R_n$, one for every choice. Padded choices are encrypted with the ElGamal encryption system using random from $R_1, ..., R_n$, resulting in cipher texts of two integers α_i and β_i, where $i = [1, ..., n]$. If voter does not use all the granted choices the remaining choices will be encrypted as an empty string.

Integers α_i and β_i, where $i = [1, ..., n]$ are then put into ASN.1 [13] notation that can be seen in the figure 8.

The application then submits the vote with all the necessary tokens to VFS. VFS validates the tokens and the vote. VFS then forwards them to the vote storage server.

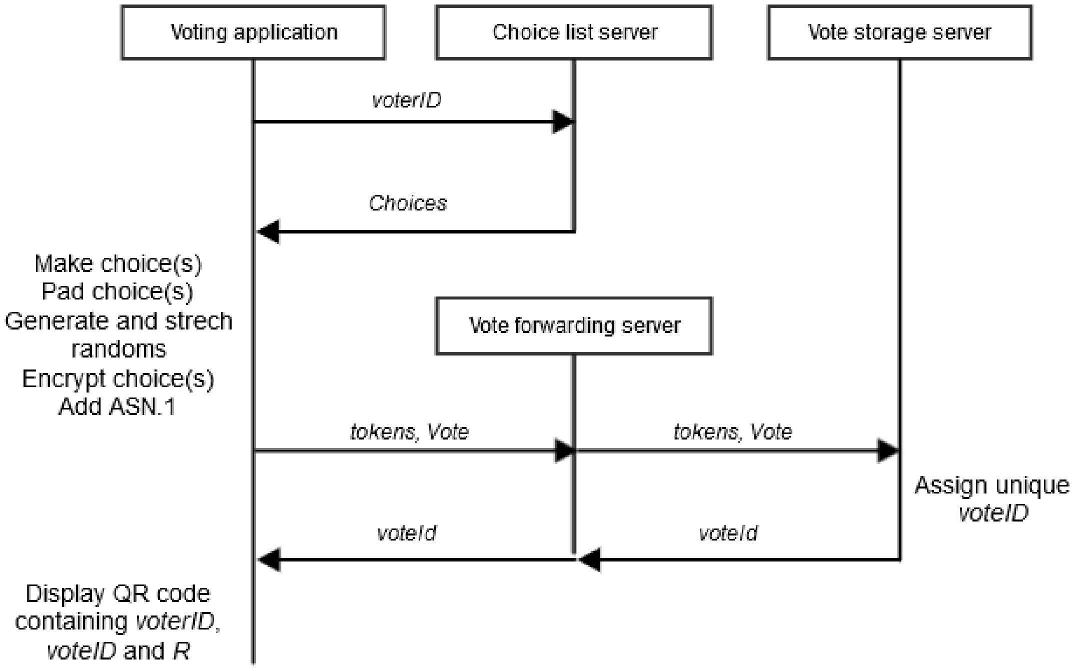


Figure 7. Vote casting process

The vote storage server saves the received vote and answers with *voteID*. VFS forwards the *voteID* to the voting application that generates a QR code for the vote verification. The QR code contents can be seen in the figure 9.

```
Vote::= SEQUENCE {
   C_1::= SEQUENCE {
      α_1 INTEGER,
      β_1 INTEGER,
   },

   ... ,

   C_n::= SEQUENCE {
      α_n INTEGER,
      β_n INTEGER,
   }
}
```

Figure 8. Used ASN.1 notation for cipher text

```
QR = voterID LF voteID LF random

voterID - token received from authentication server
voteID - hash received from the vote storage server (via VFS)
random - random that was used to for SHA-3 stretching
```

Figure 9. Contents of the QR code

3.3.2 Vote verification

The verification application on a smart device that is connected to the internet is used to scan the QR code provided by the voting application. The QR code is verified and *voterID* with additional information from the application configurations are used to request the choice list from the choice list server. It then uses *voteID* and *voterID* to receive the vote *v* from the vote storage server (via VFS). If the vote storage server does not find a vote with *voteID* error is returned and displayed in the verification application (Figure 10).

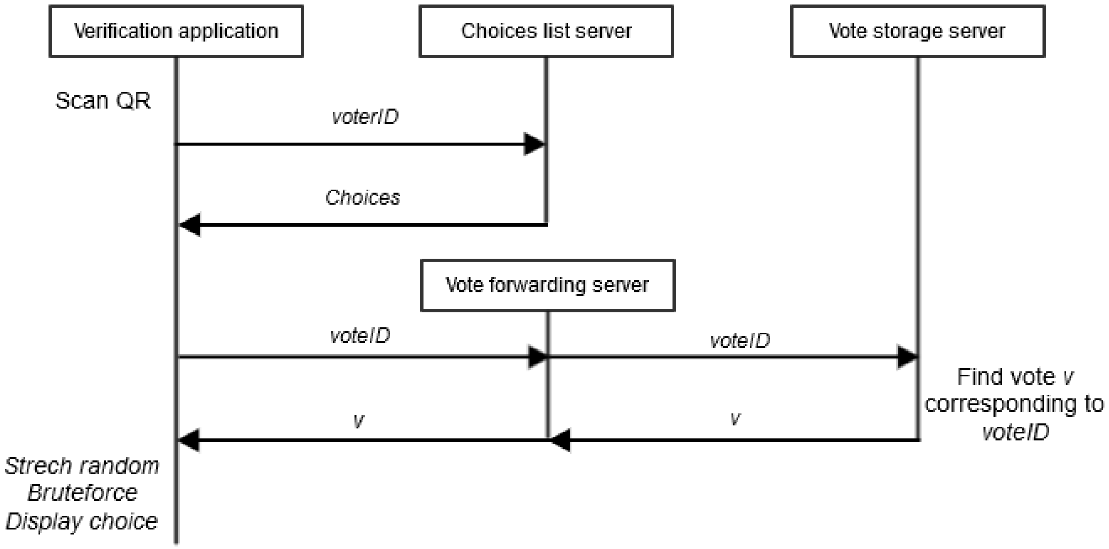


Figure 10. Vote verification process

The application then uses the SHA-256 stretching function on the *random* provided in the QR code to generate randoms $R_1, ..., R_n$, where n is the maximum number of choices granted. These randoms are then used to encrypt all the choices $C_1, ..., C_k$ on the choices list, where k is the number of choices on the choice list. Pseudo code in figure 11 is used to find the matches.

```
choicesList = [$C_1$, ..., $C_k$]
encryptedChoices = [($\alpha_1$ , $\beta_1$), ..., ($\alpha_n$ , $\beta_n$)]
randoms = [$R_1$, ..., $R_n$]
matches = []
i = 0
for encryptedChoice in encryptedChoices {
  if encrypt("", randoms[i]) == encryptedChoice {
    break
  }
  hasMatch = false
  for choice in choicesList {
    bruteforce = encrypt(choice, randoms[i])
    if bruteforce == encryptedChoice {
      matches.push(choice)
      hasMatch = true
    }
  }
  if !hasMatch {
    return Error("")
  }
  i++
}
if len(matches) < 1 {
  print("No matching candidate found")
}
for match in matches {
  print(match)
}
```

Figure 11. Pseudo code for identifying voter choices

Since the leftover choices are encrypted as empty strings and they are encrypted after the made choices it can be assumed that after matching an encryption of an empty string there are no more choices made by the voter. When all the choices have been identified they are displayed in the application.

3.4 Multiple elections

There are countries where multiple elections are held at the same time. To support all elections at the same time some additions to the protocol have to be made. For every concurrently running election there can either be a separate set of servers or a shared server with assigned ports for different elections. The latter will definitely be a cheaper option.

Only the differences to the single election protocol will be brought out. All elections have their own choice list $Choices_1$, …, $Choices_k$, where k is the number of elections that the voter takes part in. The voter makes his or her choices for every election.

Like before, a random R is generated, but the way it is stretched is different. Every election gives the voter a certain amount of choices n_1, …, n_k, where k is the number of elections that the voter takes part in. The random R is stretched with SHA-256 stretching function to generate R_1, …, R_{n1+1}, where the first n_1 are used to encrypt the voters choices for the first election he or she takes part in and the last random R_{n1+1} is used as the input for the next election random stretching function and so on for all the elections. Stretching in the verification application is changed the same way.

Encrypted choices per election are put into the ASN.1 notation as described in figure 9 and sent to the vote storage server (via VFS). The vote storage server assigns a unique *voteID* to the received vote and sends it back to the voting application. After all the votes have been successfully sent a QR code is generated for the voter to use for the vote verification. The QR code contents can be seen in the figure 12.

```
QR = voterID LF [voteID_1, …, voteID_k] LF random

voterID – token received from authentication server
voteID – hash received from the vote storage server (via VFS)
k - number of elections the voter took part in
random – random that was used to for SHA-3 stretching
```

Figure 12. Contents of the QR code

As the length of the content in the QR code is important, it could be further shortened by replacing the *voteID*s with a single unique hash, that when sent to the VFS is responded with the list of *voteID*s. Another possible way to shorten the content of the QR code is to split it into many different QR codes.

Conclusion

The aim of the current thesis was to find out whether the individual verification protocol used in the Estonian internet voting system could be expanded for other electoral systems with different ballot styles.

The thesis explained how the Estonian individual verification works. In addition to that the different ballot styles used in local or national level worldwide were introduced and made clear. Changes and additions to the protocol were analysed and justified. Problems that had occurred were described and resolved. A possible protocol for individual verification that could be used for the covered ballot styles was introduced as well as an extension for situations where multiple elections are held at the same time.

Estonian individual verification system can be modified to suit most of the commonly used electoral systems worldwide.

References

[1] Detailed overview of the voting results
http://rk2015.vvk.ee/detailed.html

[2] Vabariigi Valimiskomisjon – Electonical voting system overview, 2013 [Online]
http://www.vvk.ee/public/dok/elektroonilise-haaletamise-systeemi-yldkirjeldus-EH-03-03-1_2013.pdf

[3] QR code [Online]
http://www.qrcode.com/en/about/

[4] Bellare, Mihir, and Phillip Rogaway. "Optimal asymmetric encryption." In *Advances in Cryptology—EUROCRYPT'94*, pp. 92-111. Springer Berlin Heidelberg, 1995.

[5] Andrew Reynolds, Ben Reilly, Andrew Ellis, 2005, "Electoral system Design: The New International IDEA Handbook", International IDEA, Stockholm [Online]
http://www.idea.int/publications/esd/

[6] David M. Farrell "Electoral Systems A Comparative Introduction", 2001

[7] Ballot papers from the UK 2010 General & Local Elections, Ellis Nadler, 2010 [Online]
http://www.demotix.com/news/313012/ballot-papers-uk-2010-general-local-elections

[8] 2013 Election Guide: Automation Nation, [Online]
http://www.spot.ph/newsfeatures/53263/2013-election-guide-automation-nation

[9] Does ranked-choice voting guarantee a majority winner?, Tony Petrangelo, 2013 [Online]
https://www.minnpost.com/minnesota-blog-cabin/2013/10/does-ranked-choice-voting-guarantee-majority-winner

[10] Semi-proportional voting System, Douglas J. Amy, 2005 [Online]
https://www.mtholyoke.edu/acad/polit/damy/BeginnningReading/semiproportional.htm

[11] Bertoni, G., Daemen, J., Peeters, M., & Van Assche, G. (2009). „Keccak specifications“

[12] ElGamal, Taher. "A public key cryptosystem and a signature scheme based on discrete logarithms." *Advances in Cryptology*. Springer Berlin Heidelberg, 1985.

[13] Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation, 2008 [Online]
http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.680-200811-I!!PDF-E&type=items

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina **Joonas Lõmps** (sünnikuupäev: 18.12.1992)
(*autori nimi*)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
   Applying Estonian Internet Voting Individual Verification System to Other Electoral Systems,
   (*lõputöö pealkiri*)

   mille juhendaja on Sven Heiberg,
   (*juhendaja nimi*)

   1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
   1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **14/05/2015**