# Security and Trust for the Norwegian E-Voting Pilot Project *E-valg 2011*⋆

Arne Ansper[1], Sven Heiberg[1], Helger Lipmaa[1], Tom André Øverland[2], and Filip van Laenen[3]

[1] Cybernetica AS, Estonia
[2] Acando AS, Norway
[3] Computas AS, Norway

**Abstract.** Early 2009, the Norwegian Ministry of Local Government and Regional Development (KRD) decided to start a procurement procedure for *E-valg 2011*, an e-voting pilot project for the municipal and regional elections of 2011. The Norwegian companies Computas and Acando formed a consortium together with the Estonian company Cybernetica, the company behind the successfull electronic elections in Estonia in 2005, 2007 and 2009. The consortium proposed a solution based on an open source approach on the one side, and the knowledge and experience in the field of Cybernetica on the other side. This paper discusses the security and trust aspects of the proposal put forward to the KRD.

**Keywords:** Electronic Voting Schemes, E-voting, Open source, E-valg 2011, Norway.

## 1 Introduction

Early 2009, the Norwegian Ministry of Local Government and Regional Development (KRD) decided to start a procurement procedure for *E-valg 2011*, an e-voting pilot project for the municipal and regional elections of 2011 and some local referendums in the same year. If this pilot project is a success, the project will be continued and extended to be used in the parliamentary elections of 2013, and all futures elections from then on.

KRD, and by extension the Norwegian government, has many reasons to try out electronic voting over the Internet. One obvious reason is the possibility to speed up the counting process of the elections. If all voters would vote electronically, the counting process could potentially finish within a few minutes after the closing of the polling stations. Another reason is to attract some specific segments of the electorate, in particular young voters, so that their participation in the elections could rise. Electronic voting also has the potential to let disabled people, like the blind, vote without the assistance of other people, e.g.

⋆ This research has been supported by the Estonian Science Foundation, grant #8058, and by the European Regional Development Fund through the Estonian Center of Excellence in Computer Science, EXCS.

A. Jøsang, T. Maseng, and S.J. Knapskog (Eds.): NordSec 2009, LNCS 5838, pp. 207–222, 2009.

through the use of screenreaders. A reason that should not be neglected either is the possibility to run elections in a more cost effective way. Indeed, elections on paper may require much more time and money than electronic elections, and the vaste amount of paper that could be saved is enormous.

This paper gives an overview over the security aspects of the Internet voting application client. The client is only a small part of the complete election system, but nevertheless an important one since it will be the only part the general public will use directly and therefore have a strong opinion about. In addition, this paper discusses some thoughts on the trust of the general public in the electronic election system, as this will be of key importance to make the *E-valg 2011* project a success.

## 2 Security in the Voting Application

This section presents and discusses a number of voting schemes relevant for the Norwegian situation. They range from a completely open voting application that has little or no security against Trojans on a voter's computer, to a fully-fletched "blind" voting scheme, putting no trust at all in the voter's computer. In every section, we first describe the scheme and the user interface that implements the scheme. Then we analyze the security of the scheme, concentrating on how well it deals with Trojans. The main concerns we will deal with are confidentiality (guessing a voter's choice) and integrity (changing a voter's choice to a random party and changing a voter's choice to a particular party). After the security analysis we take a quick look at the usability aspects of the scheme and whether or not it can be implemented as an open source project.

### 2.1 Norwegian Situation

Before we dive into the different voting schemes, a few notes should be made on the specific Norwegian situation. These notes may explain why some assumptions in the discussion of the security of the different voting schemes are true in the Norwegian case, even though they wouldn't be true in the contexts of other countries.

- The principle of the double envelope won't be difficult to accept for Norwegians in general, since they are already used to double envelopes in the advanced voting period. Indeed, if a Norwegian voter decides to vote in advance, he puts his ballot inside a blank envelope, which in its turn is put inside another envelope with his name on it.
- Since it is assumed that all voters in Norway will have the opportunity to go to a polling station, either in advance or on the polling day itself, the problem of coercion and vote buying is assumed to be dealt with outside the electronic voting system.
- It follows from the previous point that confidentiality on the voter's computer isn't such a big issue. Indeed, if a voter wants to sell his vote, he will always be able to do so if the government decides to give him the opportunity to vote

over the Internet from the comfort of his own living room. The same can be said for voter coercion. The residual problem is the one of Trojans installed on the voter's computer, but it can be argued that the electronic hygiene of each voter at home isn't the responsibility for the Norwegian government (but sensibilization is). There are also many easier ways to find out which party a voter is likely to vote for, that work even in the absence of electronic voting over the Internet.
- By the time the *E-valg 2011* will be rolled out, it is assumed that a national PKI infrastructure, dubbed *eID*, will be in place. This infrastructure will provide all Norwegians who want to with an electronic ID card, together with a card reader so they can identify themselves on the Internet using the card. There are already other solutions operational, the two main competitors being *BuyPass*[1] and *BankID*[2].

### 2.2 Open Voting Scheme Using Double Envelopes

**Description.** This straight-forward approach to e-voting is taken from the Estonian e-voting system. The voter's computer is fully trusted in this system, the list of parties and candidates is the same for all voters in one district and the ballot to be cast is kept in cleartext in the computer's memory for a short time.

The following steps describe the voting process in the case of municipal elections:

- The voting application authenticates the voter to the voting server with his national ID (eID).
- The voting application receives a list of parties and candidates from the server via a secure channel; this list is not encrypted and the same for all voters in the same district.
- The voting application displays the list of parrties in a point-and-click interface, ordered randomly.
- The voter makes his decision by clicking on the party name, and clicks on the Next button to continue. He has also the option to vote blank.
- In the next step, the client shows the candidate list for the party he selected, and allows him to give a personal vote to as many candidates as he wishes. Additionally, he may also add write-ins.
- In the last step, a summary of the voter's vote is presented. If the voter is happy with his choice, he can click on the Next button to encrypt and digitally sign the ballot, and send it to the server.

**General Security Issues.** This scheme fully trusts the voter's computer and at some point in time the cleartext ballot is stored in the computer's memory.

**Eavesdropping Trojans.** It is fairly easy for a Trojan to detect the contents of the ballot from the memory. As the candidate and party names are known

[1] See http://www.buypass.no/ (accessed August 2009) for more information.
[2] See http://www.bankid.no/ (accessed August 2009) for more information.

beforehand and are the same for all voters in the same district, the Trojan can be 100 % sure for whom the particular voter is voting.

**Trojans Making Random Changes.** A Trojan that wants to perform a simple DoS attack can easily modify the plaintext ballot's contents.

**Trojans Voting for a Particular Party.** The Trojan can easily modify the plaintext ballot's contents. As the party and candidate numbers are well known the Trojan can put its own preferences in the ballot.

**Other Security Issues.** If an attacker has enough time to analyze the implementation of the voting application, and then to design and distribute the Trojan, then the described attacks can be scaled to very large numbers.

This particular scheme is also subject to attacks that try to compromise eID. If e.g. someone gets his hands on someone else's PIN codes, then he can write a Trojan that waits for the ID card to be inserted into the machine and then votes for the ID card owner. These types of attacks can not be executed on large scale though.

It would be possible to somewhat increase the security of this solution by using code obfuscation techniques in order to hide the ballot in memory. However, according to the current project plans, the solution will be on-line during the whole advance voting period. Since this advance voting period stretches over three months, the effect of any code obfuscation techniques will be negligible.

**Usability.** A friendly point-and-click user interface can be designed for this solution, showing both the list of parties and the list of candidates to the user.

**Open Source.** Making this solution open source renders it even more accessible to Trojans, as attackers can search the source code for weak design patterns and vulnerabilities.

### 2.3 Blind Voting Scheme

**Description.** This voting scheme, which may slightly remind of Okamoto's receipt-free voting scheme [1], eliminates the need to trust the voter's computer. In fact, the voter's computer never sees a single party or candidate list, as the voting process on the client computer is performed using anonymous codes, personalized for each voter, hence the name "blind" voting scheme. The personalized codes the voter uses to vote are printed on his polling card, and they are mapped to the common list of parties and candidates using a homomorphic encryption scheme.

The common list may look like this:

| | |
|---|---|
| Red Party | 101 |
| Green Party | 102 |
| Blue Party | 103 |

Voters Alice and Bob receive personalized polling cards showing the following codes:

| Party | Alice's codes | Bob's codes |
|---|---|---|
| Red Party | 234 | 135 |
| Green Party | 567 | 963 |
| Blue Party | 890 | 468 |

Internally, the server maps the codes for the voters to the parties in the following manner:

- 234 to the Red Party for voter Alice,
- 567 to the Blue Party for voter Alice,
- ...
- 135 to the Red Party for voter Bob,
- ...

Since the server uses homomorphic encryption, it knows how to map a code to a party, but doesn't know which party a particular voter voted for.

Before the election can take place, polling cards with the party codes have to be distributed to the voters over a secure second channel. This may be the postal service in the Norwegian case, but in other countries this may not be an option. Note that paper polling cards already are being distributed in Norway before every election, so this requirement doesn't represent an additional cost.

When the voter uses the Internet voting application client, she has to go through the following steps:

- The Internet voting application client authenticates the voter to the voting server using national ID (eID).
- The Internet voting application client displays an input box where the voter can enter the code for a party. The voter has to consult her polling card for the correct party code. Error detection is added to the party codes so that simple mistypings do not prevent the voter from casting her vote.
- When the party code is accepted, the client displays a list of anonymous numbers representing the candidates. The voter can select the numbers of the candidates of her preference, but she has to know in advance which numbers correspond to which candidates. If the client would present the user with a list of names, a Trojan could easily deduce which party the voter is voting for.
- Next the voter can enter write-ins. One way to do this is to enter the party code for the candidate's party from the polling card, and then select the correct candidate number for the candidate.
- When the write-ins are added to, the client shows a short summary, which again can only consist of party codes and anonymous candidate numbers.
- When the voter clicks on the Next button, the cleartext ballot is encrypted and digitally signed, and sent to the voting server for conversion using homomorphic encryption.

**General Security Issues.** This scheme requires no trust in the voter's computer at all. As all the input is personalized there is no way for a Trojan to know what the numbers entered by the voter really mean. However, for this scheme to work, a trustworthy second channel to distribute the personalized codes has to be in place.

**Eavesdropping Trojans.** Trojans can still eavesdrop on a voter's computer, but as long as there is no access to the voter's polling card, no information is gained by eavesdropping. There is a possible leak of confidentiality though: If the voter wants to be sure which order her preferred candidate has on a party list, and she uses a web browser on the same computer she is voting from to determine the ordering, then the Trojan may find out about the voter's political preference that way.

**Trojans Making Random Changes.** Trojans can still make random changes to the ballot, but they cannot be sure whether the change will be accpeted as a valid code for a party or not. Most probably the change will result in a spoiled ballot, leaving the DoS attack to e-voting an unsolved issue. Sending out receipts over a third channel may give feedback to the voter, but if the voter decides to revote as a consequence of the information in the receipt, the Trojan may gain some information about his guesses and have higher chances to succeed in spoiling the ballot the next time the voter votes.

**Trojans Voting for a Particular Party.** Trojans wanting to vote for a particular party have no way to proceed as each user has a specific polling card.

**Other Security Issues.** The scheme needs a second communication channel that has to be trusted as well. It is possible for the attacker to direct his efforts to this channel. If e.g. the postal service is used as a second channel, then malicious postal workers could:

- steal polling cards, resulting in a denial of service, since without polling card one cannot e-vote,
- forge polling cards with access to the originals: if the forging is done to change the codes of Party A to those of Party B, then all the voters for Party A would actually vote for Party B,
- forge polling cards with no access to the originals, resulting in a denial of service without the voters even knowing it,
- copy polling cards: in this case the results collected by an eavesdropping Trojan could later be analyzed by the attacker.

These attacks do not scale well, but on the level of a small community they might have important effects on the results. It's also possible to attack a particular voter.

Another issue that should be addressed is what will happen if a voter loses his polling card. Should the authorities issue a new polling card with the same codes, or should new codes be generated too?

If the second channel is another computer or mobile based, then other kinds of attack are possible.

It might also be problem that the ordering of candidates is in plaintext. A Trojan can always cancel out the fifth candidate on the list to avoid a certain candidate to be elected, or give an additional vote to every sixth candidate. Additional layers of coding for candidate order could be introduced to increase the security, but would also complicate the user interface utterly.

**Usability.** The user interface looks awkward to people who are used to simple and intuitive user interfaces. Additionaly, the user interface is of no use when a voter has lost his polling card. As we already mentioned, if an additional layer of coding for the candidate orders is introduced, then the interface gets even harder to use for an average voter.

**Open Source.** The solution can be fully open source since it wouldn't give an attacker any advantages.

### 2.4 Blind Voting Scheme Using Symbols

**Description.** This scheme is an attempt to increase the usability of the basic blind voting scheme described in the previous section. Instead of numbers, (neutral) symbols or iconic pictures are used and printed on the polling card. For voter Alice, this could e.g. look like this:

Red Party ⋆
Green Party ⊕
Blue Party ⋄

Just like for the codes, these pictures differ from voter to voter. On the server-side there is 3-way mapping: picture-code-party/candidate number.

Again, polling cards with symbols for each party have to be distributred to the voters over a secure second channel, just like for the scenario with party codes. The voter now has to go through the following steps to submit his vote over the Internet:

- The Internet voting application client authenticates voter to voting server using national ID (eID).
- The Internet voting application client displays a list of symbols. The voter consults his polling card for the symbol matching the party of his preference, and points and clicks with his mouse on the symbol that resembles his choice.
- The Internet voting application client then displays a list of anonymous numbers that represent the candidates. The same considerations apply as in the previous scheme: the voter has to know in advance the numbers of the candidates he wants to vote for.
- When the user is done, the cleartext ballot is encrypted and digitally signed, and sent to the voting server who converts it via homomorphic encryption.

**General Security Issues.** The mapping of symbols to codes must be known to the Internet voting application client. It must therefore be sent to the voter's computer by the voting server, and a Trojan will be able to read it. Notice that the server should always send the same set of codes to the client, both valid and invalid. Otherwise a Trojan could send three or four requests to the server, and eliminate the codes that don't reappear in the responses as invalid. After a few times, the Trojan will be able to guess which codes are the valid codes for a particular voter.

**Eavesdropping Trojans.** Eavesdropping is not easier than in the basic blind voting scheme, since a Trojan can't gain any information about a voter's political preferences by observing the symbols he selects in the user interface.

**Trojans Making Random Changes.** As the Trojan sees all the codes associated to the voter, it can make a better guess than in the basic blind voting scheme. If 100 icons are used to present to the voter, and 10 parties participate in the contest, then he has a $10/100 = 10\,\%$ chance of picking a valid code at random.

**Trojans Voting for a Particular Party.** Voting for a particular party is easier than in basic blind voting scheme in the sense that the space from which the Trojan has to make a selection is drastically reduced. However, the Trojan has no information about which symbol belongs to which party for a certain voter, and therefore only has a $1/100 = 1\,\%$ chance of picking the right code for a particular party.

**Other Security Issues.** The same considerations about trust in the second channel apply for this scheme as for the previous one.

**Usability.** This scheme is slightly more usable than the basic blind voting scheme, because party icons can be recognized by illiterate persons as well. On the other hand, it has to be noted that for illiterate people it may be a difficult task to use the Internet voting application client anyway, e.g. because they have to authenticate themselves using the national ID. For visually handicapped people, however, this scheme will be less usable.

**Open Source.** Open sourcing a project using this scheme may have the disadvantage that the mapping between symbols and codes can be found out more easily than otherwise.

### 2.5 Blind Voting Scheme Using CAPTCHAs

There are several other ways to improve the basic blind voting scheme. One such approach would be to eliminate the second channel. Each voter still receives a personalized list of parties, but now in the Internet voting application client in a similar way as in the open voting application scheme. This allows for a more

user friendly interface and offers some sort of protection against eavesdropping and ballot modification. This protection is not very strong though:

- The names of parties and candidates and corresponding codes are in the computer's memory at the same time. Although the numbers are personalized, it is still possible to detect the mapping via memory analysis. It should also be noted that the presentation of the party names and codes is somewhat more complicated than in straightforward case.
- A Trojan on a voter's computer can see all party and candidate numbers available for the voter. A simple Trojan may in fact use it for a more advanced DoS: the voter's choice is altered and this time the new choice is accepted by the server because the Trojan picks the correct number from the computer's memory.

It is possible to overcome these weaknesses by using CAPTCHA technology. CAPTCHAs are mainly used to distinguish between humans and computers on websites. Some sort of challenge that is hard for computer to solve, but easy for humans, is presented to the user. Most applications of CAPTCHA use object recognition: the human brain is very good at telling visual objects such as typefaced letter apart from its background, but for a computer this task is time consuming, and no general working solution is known. If the answer to the CAPTCHA appears relatively fast, then we know we are dealing with a human, otherwise it might be computer. It should be noted that CAPTCHA technology, or more correctly CAPTCHA breaking technology, is developing relatively fast. Therefore, a scheme that is secure today might not be secure tommorrow.

**Description.** The blind voting scheme using CAPTCHAs achieves the same goal as the basic blind voting scheme, but tries to increase the usability. There is no need to communicate any codes to the voters in advance to the election period using a second channel.

When the voter wants to cast his vote over the Internet, she has to follow the following procedure:

- The Internet voting application client authenticates the voter to the voting server using the national ID (eID).
- The Internet voting application client receives an image file in which the voter's personalized list is encoded as a dynamically generated CAPTCHA.
- The Internet voting application client displays an input box where the user enters a party number. The voter has to solve the CAPTCHA in order to get the correct party number.
- The Internet voting application client then displays a list of anonymous numbers that represent the candidates, and in a similar way, the write-in candidates can be added in the next step.
- When the user is done, the cleartext ballot is encrypted and digitally signed, and sent to the voting server for conversion using homomorphic encryption.

**General Security Issues.** The need for a secure second channel is eliminated. The Internet voting application client downloads the image from the server on-the-fly.

**Eavesdropping Trojans.** Eavesdropping is possible: the Trojan saves the image received from the server and the cast ballot. Both of them can be sent to the home base of the Trojan for analysis. This attack can be scaled if the attacker hires human CAPTCHA solvers, or implements a CAPTCHA solving algorithm.

**Trojans Making Random Changes.** If the Trojan can't solve the CAPTCHA, then the scheme is as secure as the basic blind voting scheme.

**Trojans Voting for a Particular Party.** If the Trojan can't solve the CAPTCHA, then the scheme is as secure as the basic blind voting scheme. If a Trojan can solve the CAPTCHA, then the scheme is no longer secure: the Trojan can cast ballots on the voter's behalf.

**Other Security Issues.** The defence against Tojans depends on their ability to solve CAPTCHAs efficiently. Microsoft's Live Mail service has been reported to be broken by spammers with a success rate of 30 % to 35 % [2], while Google's Gmail CAPTCHA was broken with a success rate of 20 % [3]. Notice however that the CAPTCHA challenge in this scheme is easier than guessing just a random sequence of letters like one often sees as protection on websites. Indead, the attacker knows in advance what will be the relevant words to look for in the CAPTCHA, i.e. the names of the parties. Just recognizing the first letter in a word may e.g. be enough to recognize the name of a particular party. It may therefore be necessary to change the CAPTCHA algorithm during the election period.

**Usability.** The user interface still looks awkward to people who are used to simple and intuitive user interfaces. Without modifications, people with disabilities may not be able to use the user interface either.

**Open Source.** The solution can be open source except for the CAPTCHA generation algorithms.

### 2.6 Tamper Indicating Open Voting Scheme

**Description.** This novel scheme [4] tries to solve the vote integrity problem, using a second and a third channel. For each voter, a polling card is generated containing control codes for each party, and these polling cards have to be distributed to the voters over a secure second channel. This may again be the Norwegian postal service, just like in the blind voting scheme, but may vary in other countries. The control codes should be sent back to the voter over a secure third channel, like e.g. over SMS. It is important that these codes aren't sent back over e-mail to the same computer the voter is voting from, since this will give Trojans the opportunity to manipulate the contents of the e-mail.

This time, the user has to do the following to cast his vote:

- The Internet voting application client authenticates the voter to the voting server using national ID (eID).
- The Internet voting application client receives a list of parties and candidates from the server; this list is not encrypted since it is the same for all voters in the same district.
- The Internet voting application client displays the list in a point-and-click interface.
- The voter makes his decision by clicking on the party name, adding additional votes for some candidates and adding write-ins too if he wants to.
- When the voter is done, the cleartext ballot is encrypted and digitally signed, and sent to voting server.
- The server then regenerates a control code from the encrypted ballot and sends it back to voter via the secure third channel (e.g. SMS) in a receipt. The message should contain instructions to verify the code against the code on the polling card.
- The voter checks whether the control code is the one for the party he voted for.

**Eavesdropping Trojans.** This scheme doesn't protect the confidentiality of the voter against Trojans. Since the ballot will still be present in plaintext in the memory of the voting application for at least a small amount of time, a Trojan could be used to post on a bulletin board who voted for which party.

**Trojans Making Random Changes.** A Trojan will not be able to make random changes to the ballot against the voter's will if the voter does verify the control codes he receives over the secure third channel against the one on his polling card.

**Trojans Voting for a Particular Party.** A Trojan will not be able to make targeted changes to the ballot against the voter's will either if the voter does verify the control codes he receives over the secure third channel against the one on his polling card.

**Other Security Issues.** Since this scheme is work in progress, more research needs to be done and will be done on its security before it will be implemented and used.

**Usability.** This scheme is slightly more complicated than the straightforward scheme due to the control codes and the use of a secure third channel.

**Open Source.** It is our goal that the security of this scheme lies in the private and secret keys used in the scheme, not in the mechanisms, such that it can be published as open source.

### 2.7 Discussion

Table 1 shows some rough estimates for the success rates Trojans will have for a number of specified attacks on the voting application. From this table, it is clear that if confidentiality on the client's computer is not an issue, the tamper indicating open voting scheme comes out best. If the confidentiality is an issue, then the blind voting scheme using codes is the best option, but as pointed out in subsection 2.3, the usability of this scheme is quite a challenge. If neither confidentiality nor integrity are an issue, the open voting scheme probably is the best choice.

**Table 1.** Estimates for the success rates of Trojans for the Open Voting Scheme Using Double Envelopes (OVS), Open Voting Scheme Using Double Envelopes, Open Source (OVS-OS), Blind Voting Scheme Using Codes (BVS-C), Blind Voting Scheme Using Symbols (BVS-S), Blind Voting Scheme Using CAPTCHAs (CAPTCHA) and the Tamper Indicating Open Voting Scheme (TIOVS) on eavesdropping, making a random change to the ballot, and making a change to the ballot in favor of a particular party. For simplicity, the figures are based on the assumption that 10 parties participate in a contest. For BVS-C and TIOVS, we assume the codes consist of five alphanumeric chartacters, i.e. digits and capital letters (minus captial O as to avoid confusion with the digit 0), thus allowing for $35^5$ combinations. For BVS-S, the calculations are based on 100 different symbols being available. Finally, we assume the success rate to decipher a CAPTCHA to be somewhere between 70 % and 90 %.

| Voting Scheme | Eavesdropping | Random alteration | Targeted alteration |
|---|---|---|---|
| OVS | 90 – 100 % | 90 – 100 % | 90 – 100 % |
| OVS-OS | 100 % | 100 % | 100 % |
| BVS-C | $1.9 \times 10^{-6}$ % | $1.9 \times 10^{-5}$ % | $1.9 \times 10^{-6}$ % |
| BVS-S | 1 % | 10 % | 1 % |
| CAPTCHA | 70 – 90 % | 70 – 90 % | 70 – 90 % |
| TIOVS | 100 % | $1.9 \times 10^{-5}$ % | $1.9 \times 10^{-6}$ % |

## 3 Trust

### 3.1 Introduction

The previous section discussed the security in the voting application. Many schemes were proposed and discussed, with their advantages and merits, but also their disadvantages and problems. In the end, a trade-off will have to be made if the Norwegian government wants to proceed with the *E-valg 2011* project and try out voting over the Internet.

Applying the right amount of security to a project may be a challenging task, but an even more challenging task for a system like *E-valg 2011* may be to gain trust for it. This trust can be situated on two levels: at the level of the Norwegian government, and in particular KRD, as the customer of the project, and at the level of the general public in Norway, as the user of the system to be developed.

The general public may again be divided into two groups: technical experts who can formulate qualified opinions about the security of the application and whether or not it can be trusted, and the non-experts who, to a certain degree, will rely on the technical experts to accept or reject the voting application.

Since elections are the cornerstone of any modern democracy, it is of paramount importance that the general public trusts the election system. If part of the election process is implemented using a voting system that the general public can't or won't trust, then there may be enormous consequences. The least consequence may be that the public chooses not the use the Internet voting system, in which case the money and time invested in the implemented system can be considered as a loss, and the project a failure. A far more serious consequence could be that the general public simply doesn't accept the legitimacy of the elected body, be it a local district council or a national parliament.

The strategy of the KRD to gain trust with the public for voting over the Internet seem to include three tactics: informing the general public about Internet voting and the concrete solution at hand as best as possible, a formal certification of the solution by a third party, and open sourcing the project so that those who are able to and want can verify for them selves that the system does what it is supposed to do in the correct way.

### 3.2 Information

The KRD, and in particular the *E-valg 2011* project group, has already created a website [5] where the general public can find information about what it is planning to do. It contains a lot of general information about voting over the Internet and electronic voting in general, including studies, reports, and links to other sites. There's also a FAQ section, which even before the end of the procurement process already includes questions about how privacy will be guaranteed, and whether the system should be trusted or not.

Experiences in other countries have shown that providing the general public with clear and understandable general information about the electronic voting system is essential for such projects to become a success. The Austrian case, discussed below, is an example of that. Of course, not everybody in the general public will have enough technical compentences to understand the inner workings of an Internet voting system, let alone the cryptographic aspects of it, but on a general level, people want the system to be explained to them in order to gain some trust in it. If for nothing else, one thing general information about the system definitely will show is whether the people in charge really know what they're doing or not.

### 3.3 Certification

Another, more formal way to gain trust in a system is to have it certified by a third party. For the customer, the Norwegian government represented by the KRD, this is probably the most realistic way to verify that the system has all the necessary security aspects, in addition to its own acceptance tests and a thorough follow-up of the project during all its phases.

The KRD has indicated that it wants the system to be certified through SERTIT, the public certification authority for IT security in Norway. However, the certification doesn't have to be in place for the 2011 pilot project, but rather for the 2013 parliamentary elections if the pilot project is a success and Internet voting to be rolled out over the whole country by then. One reason why KRD doesn't require a certification for the 2011 pilot project is to save time and money in the project, in addition to the fact that it is after all a pilot project that will be tried out on a rather small scale (though large enough to gain enough realistic experience). Another reason is probably that the project will be open sourced too, as discussed in the next section, which may actually put more pressure on the supplier than a formal certification would do. Finally, since a certification will be required for the final project to be delivered in 2013, and certification is a time consuming process, a lot of the preparations for a certification will have to be made during the development of the 2011 pilot project anyway.

### 3.4 Open Source

Finally, an additional way to let the general public gain more trust in a voting system is to publish its source code, e.g. by making it an open source project. The KRD has already decided and publicly announced [6] that it wants the *E-valg 2011* project to be implemented as an open source project. But what are the advantages and the problems with having such a project as open source?

One clear advantage would be that every voter has access to the source code of the project, and can, at least in theory, evaluate for himself whether or not he decides to trust the voting system. In practice, it can be expected that only a limited number of voters will do this exercise. Most probably, this will be a self-selected group of technical experts, and their opinion will have great influence on the general public's opinion about the system. In fact, if a large enough part of the system documentation is available in English, even people who do not have voting rights in Norway could participate in the exercise, and give their opinion too. The key point however is that a voter doesn't have to rely on the Norwegian government and/or a certification authority, and can review the code on his own, if he wants to.

One of the biggest disadvantages of having the project open sourced is of course that malicious hackers would have complete access to the source code too, and be able to carefully craft their attacks on the system. This doesn't make much of a difference for the part of the system that runs on the client side, since any code running there always can be grabbed from the client's memory and decompiled. Obfuscation of the source code could make this task a more difficult one, but considering that the advance voting period in Norway is three months long, it can be argued that the effect of obfuscation would be rather marginal. The problems associated to open sourcing the project therefore lies more on the server side: in an open source project malicious hackers have access to the source code on the server too, including a large part of the configuration necessary to run the system. Security by obscurity sometimes does have its merits, and this

is such an example. We think however that the disadvantages of having this project not open sourced outweight the disadvantages of having it open sourced, as we will discuss in the next section.

### 3.5 Discussion

Indeed, providing general information about the voting system and having it formally certified by a third party are two necessary conditions for the general public to gain any trust in it. Having the project open sourced seems to be more and more important too, especially where new voting systems are being introduced. The 2009 elections for the Austrian National Union of Students (*Österreichische Hochschülerschaft*, ÖH) are an example about how this demand from the general public made the project fail in the end.

What happened in Austria, is that even though the system had been formally certified, a popular movement called *Papierwahl.at*[3] was started against it. It is difficult to say how large the movement really was and how big its influence was, but it certainly was able to gain some attention in the media, and in the end only 0.9 % of the voter mass used the Internet application to cast its ballots [7]. One thing the Austrian government clearly seems to have done wrong is that it selected a rather small group of experts to have a look at parts of the source code for a limited amount of time. This move can be considered to be a compromise between having the project close sourced and open sourced. But instead of gaining some credit for the system, the government only received a lot of critique, both on the selection of which experts could join the code review –especially from people who would have liked to be included in the group– and the modalities on how the code review was organized [8]. It looks like the biggest looser of the election wasn't one of the participating parties, but the organizer of the election, the Austrian government.

## 4 Conclusion

Even though electronic voting seems to be very appealing with all its advantages, there are a lot of problems attached to it too. There have been many attempts in recent history to automate parts or all of the election process, not always with success. One country that has been able to run electronic elections without any major incidents is Estonia. In 2005, 2007 and 2009, voters had the possibility to vote over the Internet, and quite a few of them did so. It is the intent of our consortium, which includes the company that delivered the Estonian solution, to repeat this success in Norway, adapted to the specific Norwegian context.

At the moment of this writing, no decision has been made yet about which voting scheme is the most appropriate for the Norwegian context. The procurement process for the pilot project is due to result in the signing of a contract at the end of 2009.

[3] See http://papierwahl.at/ (accessed August 2009) for more information.

## Acknowledgments

The authors would like to thank all their colleagues who have contributed so far to the *E-valg 2011* tender.

## References

1. Okamoto, T.: Receipt-Free Electronic Voting Schemes for Large Scale Elections. In: Christianson, B., Lomas, M. (eds.) Security Protocols 1997. LNCS, vol. 1361, pp. 25–35. Springer, Heidelberg (1998)
2. Keizer, G.: Spammers' bot cracks Microsoft's CAPTCHA. Computerworld (February 2008), http://www.computerworld.com/s/article/9061558/Spammers_bot_cracks_Microsoft_s_CAPTCHA_ (accessed August 2009)
3. Prasad, S.: Google's CAPTCHA busted in recent spammer tactics. Websense.com (February 2008), http://securitylabs.websense.com/content/Blogs/2919.aspx (accessed August 2009)
4. Heiberg, S., Lipmaa, H., van Laenen, F.: On achieving e-vote integrity in the presence of malicious trojans (working title) (In preparation)
5. Ministry of Local Government and Regional Development: E-valg 2011-prosjektet (August 2008), http://www.regjeringen.no/nb/dep/krd/kampanjer/valg/elektroniskstemmegivning.html?id=437385 (accessed August 2009)
6. Ministry of Local Government and Regional Development: Forsøk med internettvalg i 2011 og bruk av åpen kildekode (July 2009), http://www.regjeringen.no/nb/dep/krd/kampanjer/valg/elektroniskstemmegivning/nytt-om-e-valg-2/nytt-om-e-valg/2009/forsok.html?id=570946 (accessed August 2009)
7. Kleijn, A.: Österreich: Nur 0,9 Prozent Wahlbeteiligung bei E-Voting. Heise Online (May 2009), http://www.heise.de/newsticker/Oesterreich-Nur-0-9-Prozent-Wahlbeteiligung-bei-E-Voting–/meldung/138303 (accessed August 2009)
8. Sokolov, D.A.J.: E-voting ist in Österreich nicht unbedingt geheim. c't Magazin (May 2009), http://www.heise.de/ct/E-Voting-ist-in-Oesterreich-nicht-unbedingt-geheim--/artikel/138049 (accessed August 2009)