Eindhoven, November 17th 2006

i-voting with RIES analyzed

The Dutch government will do an experiment with i-voting for the 2006 elections of the Dutch parliament. This experiment is restricted for the people abroad, as alternative for voting by letter. In 2004 a similar experiment was executed for the elections of the European Parliament. In 2006 a different system has been chosen. The Ministry of Internal affairs choose for RIES. The abbreviation stands for "Rijnland Internet Election System" and has been successfully used for the elections of the "waterschappen". Historically, the Netherlands has a separate government for water related issues. This ensures that long term issues (such as a quality dike) get enough priority and are not dependent of the politics of the regular government. Elections for these water governments have never been very popular.

In this article I will take a look at RIES and try to find the vulnerabilities of the system. However, I will start with a general assessment of the risks of i-voting.

Fraud

The greatest danger of i-voting on a large scale (for a whole country) is that everything is controlled from 1 central place. That means, that if there is a way to manipulate a vote, then the possibility exists to rig the election. This is in contrast with voting by pencil and paper. With this traditional way of voting, it is rather easy to add or to let disappear some ballots, but it is much harder to do this on a large scale.

If an i-voting system has some basic quality and all the known security issues are addressed, then it is rather impossible to commit fraud from the outside the system. This is called external fraud. You just can't enter a good system from outside.

However, with this analysis I explicitly also want to explore the possibilities that some person from inside the election organization wants to do the job (internal fraud). This may sound strange for people that do not deal with security. But it isn't strange. For banks internal fraud is a bigger problem than external fraud. It also appears that often people of the polling station are involved in case of election fraud with pencil paper. Large scale fraud (in Mexico or the communistic countries) need to be organized from the Ministry of Internal affairs.

RIES has the possibility that voter can check their votes afterwards. This suggests that internal fraud is made impossible. If it turns out that this is not true, then a complot is necessary to rig the elections. However, a complot becomes unlikely when lot of people are necessary to execute the complot. A system becomes internally more secure, when the number of people to commit fraud increases. A rule can be set that at least two people of different departments must be necessary to commit fraud (such requirement is rather common in banks). I will analyze RIES from this point of view. I will look which fraud attempts can be detected by the voters and if it can not be detected, I will look how much people of the election organization are necessary for committing the fraud.

It is partly a political issue which requirements for security are set. The opinions can differ whether we want to be dependent on 2 persons or whether we want the possibility that everything can be checked by the voters. However, it is reasonable to say, that we don't want that the integrity of the elections is dependent on just one person.

It should be noted that the probability on internal fraud is quite small. I do not assume that civil workers of the Ministry of internal Affairs use their full days to think how they can rig the elections. But a risk consists of two things, the probability and the damage. The probability is small, the damage is huge. Undetected fraud means a breach in our democracy. I read the documents of the government and I got a little bit the impression that it is assumed that any fraud will be detected. The damage of detected fraud is still substantial (in worst case the elections have to be redone), but it is much smaller than undetected fraud. It is difficult to put the damage in perspective, but the reader can consider what is worse, something like the King’s Cross fire or undetected election fraud on large scale.

For large scale fraud it is not necessary to temper with all votes. I consider a percentage of 5% enough. For the Dutch parliament that would be about 7 seats and that amount can lead to a different coalition. Furthermore, a 5% deviation with the polls is not very suspicious. Also note, that polls are often corrected with the use of real results, which means that on the long term polls will follow the fraud.

The target of 5% means directly that the risk for the experiment is very low. The experiment is restricted to the people abroad and only 20.000 people registered. That is only 0,17% of the votes. The effort and the risk to be get caught is just too high for the pray. But after a successful completion of the experiment, it should not be concluded that RIES is safe to be used countrywide. The history of computers shows that enlarging the scale introduces new security issues.

Preventing fraud by introducing a technical barrier or by keeping the software secret (security by obscurity) improves the security on the short term. However, it only delays things and it is not something to trust on for the long term.

The secrecy of the vote

The second concern is the secrecy of the vote. It should be noted that some people have a wrong perception of the purpose of the secret ballot. The secret ballot is demanded in the 19th century to prevent intimidation or bribery by parties or fanatic followers. The primary purpose was not to prevent a totalitarian state which will check all the votes of the people. The book 1984 of George Orwell has been written a century after the introduction. Nevertheless, the secret ballot is must also make any coercion of the government impossible.

The freedom of expression is not directly at stake when the secrecy of the vote has been violated. Someone has to do something with the knowledge of the votes. A voter has to be intimidated or be bribed for taking advantage of the knowledge. If the knowledge of the votes is obtained illegally, then using this knowledge has the risk that the illegal action becomes revealed. Therefore, a legal way to obtain someone’s vote, is a bigger problem (voting by proxy or by letter). At all, it can be concluded that the security around the secret of the ballot can be much lower then the security to prevent fraud.

Sometimes it happens that by considering the technical aspects of a system, a legal point of view is taken toward the secrecy of the ballot. This means, that no information should be registered by which the vote can be linked with the identity of the voter. To my opinion, this approach is rather absurd. It leads to rather rigid requirement in one area, while a small breach in the security in other area can reveal the way how people voted. I take a security point view, by looking how much effort and persons within the organization are necessary to break the secrecy.

Disorder

The final concern is that someone wants to put the election in disorder. The most obvious way to do so, is by a DDOS (Distributed Denial Of Service) attack. With such an attack thousands of hacked computers start to send requests to the webserver of the elections. This will overload the webserver. However there are several counter measures for sale and to my opinion this problem can be controlled.

There is a reasonable probability that people from outside the election organization wants to disturb the elections. But it should be realized that the damaged is limited. In worst case, the election has to be redone. The probability that someone from inside the election organization wants to disturb, is much smaller, but is not zero.

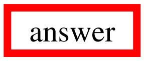
Overview of the RIES system

In contrast of the i-voting experiment in 2004, there is lot of information available about the working of RIES. Therefore, I can honestly say, that this is indeed something that may have the label 'experiment'. The system and the results of the experiment can be scrutinized by experts.

The first impression of RIES is positive. The designer really thought about security when voting with the use of internet. The whole design of RIES is based on security.

One cryptographic method must be explained to make it a little bit clear how RIES works. This method is the 'hash-code' and is used in RIES on several places. This sounds complicated, but it is not that difficult. A hash-code is a mathematical operation on a piece of information (for instance an email, a number or piece of music) with as result a number with limited number of ciphers. The mathematical operation is designed in such way that no information can be easily constructed that will have a certain given result number. So, also the original data can not be found.

This is very abstract, but becomes more clear with an example. Suppose Al and Bob are solving a puzzle. The goal of the puzzle is to find a website on the internet with the answer. At a certain moment, Al has found the answer and wants to proof it to Bob. However, he doesn't want to spoil the puzzle for Bob by giving the answer. Therefore, he performs the hash-code on the website address. Schematically I show this as follows:



So, Bob gets this box, but he can't look in the box. When Bob does also find an answer, Al gives his answer to Bob. Bob constructs the box in the same way Jan did, by performing the hash-code on the website address. Bob compares the result with the number that Jan gave him, when he did not know the answer yet. He sees that the number is equal and concludes that Jan indeed had the answer before he did.

Realize, that a piece of information before hashing (such as the answer in the example) is more 'hot' and 'dangerous' then the number that is the result of the hash.

RIES is build around this cryptographic principle. In RIES there are two computers. I call the first computer the ‘votes generator’ (the documentation I used, didn’t have a clear terminology) and the second computer is the ‘voting webserver’. The votes generator computers is not connected with the internet or any other network. This computer is used in the preparing phase of the elections. The votes generator computer gets as input a database with all the necessary for information of the voters. The votes generator computer produces two outputs. First of all, the information that must be printed on the voting cards (actually this will be in an closed envelopes). This information has to be brought to the printer. On every card an identification of the voter is printed and a ‘voting code’.

Second, the votes generator produces a ‘reference file’. This file contains all possible coded votes for every voter. The reference file will be published after the electronic ballot box is closed. To proof that the file has not been modified during or after voting, a hash-code of the file is published in the ‘Staatscourant’ before voting starts. The ‘Staatscourant’ is the official publish channel of the Dutch government.

After the votes generator has done its job, the voting codes are destroyed (well, if everyone works according to the procedures). This means, that at hat time the voters are the only persons that possess the voting codes. The reference file contains only hashed voting codes.

During voting, the voters makes connection with the voting webserver by using his webbrowser. The webserver will send a ‘Javascript’ program to the voter which be loaded and executed in the webbrowser of the voter. The voter fills in his identification, his voting code and the candidate of his choice. The Javascript program will hash the identification with the voting code and also the candidate. This is called the ‘technical vote’. This is send to the voting webserver. The voting code itself will not be send to the voting webserver.

The voting webserver is not much more than a recording machine. It will record the technical vote. Since the candidate is hashed, the voting webserver can not determine the candidate.

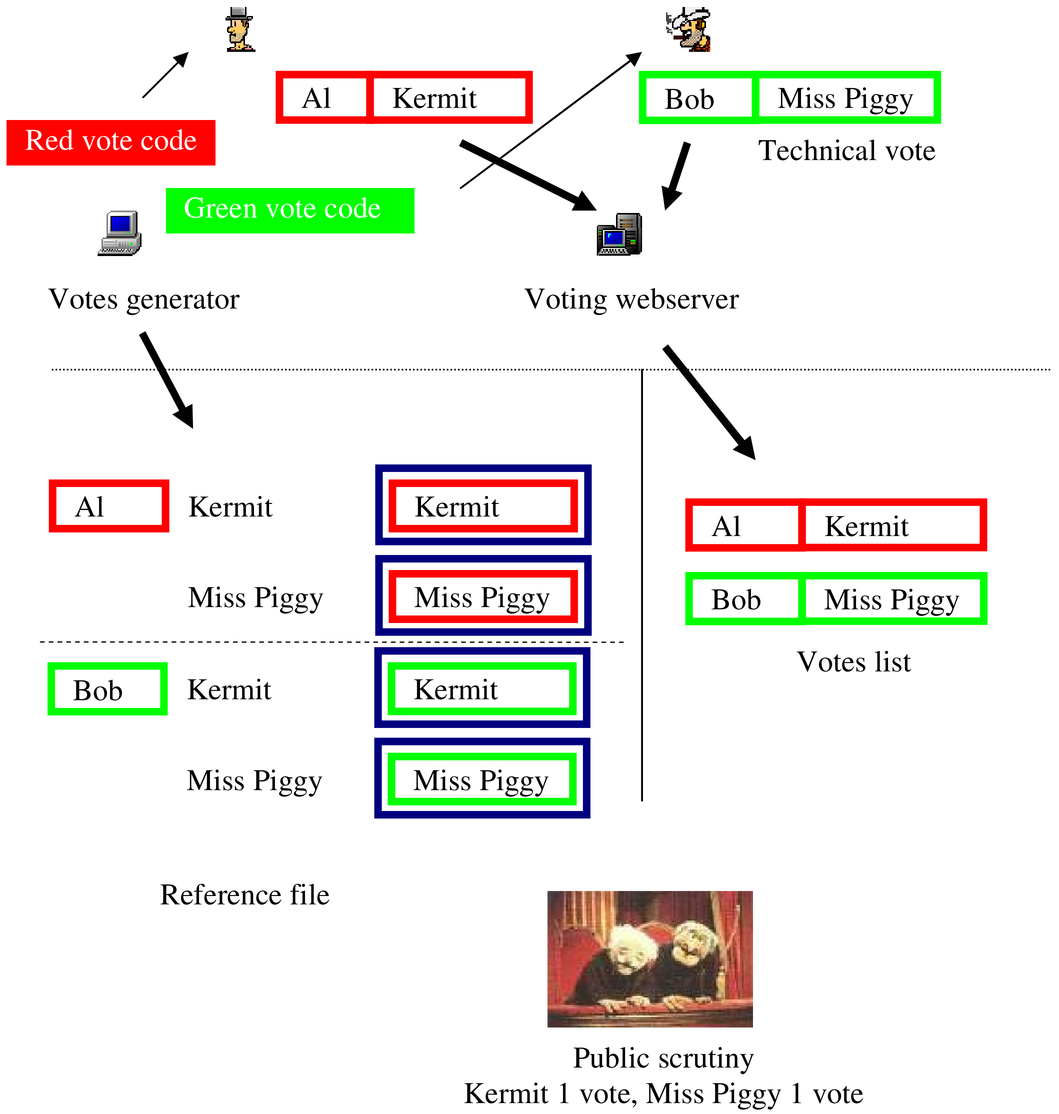
When the voting is finished, all votes collected by the voting webserver are published. I call this the ‘votes list’ (unclear terminology in the documentation). After this the reference file is published by the organization around the votes generator.

With this public information, the counting can start. The votes list is combined with the reference list, to calculate the final result. Of course, this is executed by the election organization, but in principle anyone can do it.

The individual voter can search for his technical vote and see if it was counted for the right candidate.

In principle the security of the system does not depend on openness of the source of the software.

Schematic it looks like this:



Remember, if a text is boxed, then the information in the box can not be seen. Only when the information in the box is known, then the box value can be recalculated and the results can be compared.

Different colors are used in the scheme. It is possible to make the hash-coding dependent of a code. In RIES the hash is dependent of the voting code and different voting codes are showed in different colors in the scheme. Al and Bob have different ways of hashing their identification and choice of candidate.

A blue box is drawn in the reference file around the votes. This means that a 'Kermit' or 'Miss Piggy' with red or green box only, can not be constructed from the reference file. After the destruction of the voting codes on the computers of the voting organization, the voters Al and Bob are the only one that can construct a vote with red or green box. To count the final result, the blue box around the candidates in the votes list should be calculated. This value should be searched in the reference file

Possibilities of fraud with RIES

Now I will take a look whether it is possible to break the security of RIES. It is not very smart to break a system, whether it is a house or computer, on the place where it is most heavily protected. If the front door of a house has many locks, then one can better look for a back door. And when the back door has also many locks, then it should be considered to steal the car in front of the house. I mean, that I will try to manipulate the elections in the most stupid way.

Breaking the cryptographic methods is certainly the hard way. I will not consider this. However, one remark, the voting codes are generated by using one master key. I consider this rather dangerous. All voting codes can be recalculated by obtaining the master key. I would prefer that the voting codes are produced by random numbers (using a piece of hardware that can produce real random numbers).

RIES is very safe in relation to external fraud. Even when I would succeed in breaking into the voting webserver, then still I could not do very much. On the voting webserver only hashed votes are stored, which can not be falsified without the original voting codes. I assume that installing different software on he voting webserver, will not remain undetected. Furthermore, the voting webserver has a simple design, which also contributes to the safety

A second possibility of external fraud is attacking the computer of the voter. I could think of phishing or writing a virus that will have as goal that a different vote is send to the voting webserver than the voter intended. Although I could be successful in misleading some individual voters this way, the total method will probably not remain undetected. Also I do not think large numbers of votes can be altered this way. Therefore I consider such attack as totally ineffective.

But, as I wrote before, internal fraud should also be considered. With internal fraud I assume I get help from inside the election organization. I will call a contact within the election organization a ‘joker’. It is reasonable to say that the system is unsafe when I can achieve something with just one joker.

I can try to manipulate the result by removing or altering votes. My first impression was that this was rather impossible. If voters will check their votes afterwards, they will detect my fraud. Furthermore, it seems I need two jokers. For letting undesirable votes to disappear I need a joker near the voting webserver but I also need the reference file. The reference file will only be published after the votes list has published. Changing votes is even more complicated. The reference file is insufficient. I need also the voting codes, which are destructed after the generation process.

However, in my second thought I saw some possible tricks. The voting webserver sends a program to the browser of the voter. This program is written in Javascript and will do the cryptographic calculations (hash-coding) for making the technical vote. The technical vote will be recorded by the voting webserver.

I can put my joker near the voting webserver and let him alter the webserver in such way that a falsified program is send to the browser of the voter. This Javascript program gets the voting code of the voter (so, I don’t need a second joker for a file with all the voting codes) and the Javascript program registers a vote that I want.

Nevertheless, I have still the problem that the voter can check his vote afterwards. However, this can be solved. The first time a voter casts a vote for a certain candidate, then the vote is registered as normal, but the voting computer will also keep this vote on a separate place. If another voter comes and he wants to vote on the same candidate, then the saved technical vote will be showed to the voter and a candidate that I want will be coded with the voting code and will be recorded by the voting webserver. So, the two voters will see the same technical vote on their screen. When they check their votes afterwards, they will see that their votes are counted for the right candidate, but in fact the vote of the second voter has gone to another candidate.

I consider this an error in the design of RIES that this is possible. For checking the vote afterwards, the voter needs an identification of himself. The identification on the voting card is not anonymous. The identification becomes anonymous when it is hashed with the voting code. However, this coding is executed by the Javascript program that comes from the voting webserver. And that Javascript program can be falsified. If RIES was designed in such way that the identification of the voter was already anonymous on the voting card, then the hashing by the Javascript is not necessary. Then the voter could check his vote by using the identification on his voting card, instead of a code reported back by the Javascript program.

This alternative has also some disadvantage. Someone that finds a voting card in the garbage can directly check what has been voted. In RIES it is also possible to find out what has been voted based on the voting code, but this is more complicated. The identification needs to be hashed with the voting code. When RIES is introduced large scale, then it is not unlikely that programs on the internet will appear that will do this task.

There is another reason, why the way the identification of the voter in RIES is organized is not so smart, but I will explain that later.

Back to the fraud. The way I supposed could be detected by inspecting the Javascript that has been send to the browser. In principle the voter can see that the Javascript does some other things than it should. This inspection could be performed by an independent organization with technical knowledge. It could act as a voter and check the Javascript that has been send and compare it with a reference script given by the RIES organization. For an individual it is practically impossible to do this check. No reference script has been published and no instruction has been given whatsoever. That means that the voter has to find out how the script works and see that it behaves correctly. This is an enormous amount of work. To give an indication, it would take me at least a day to get some idea what the script is doing. And I am not the worst programmer.

Furthermore, if a voter detects that the program is falsified, then he doesn’t have a proof yet of the falsification. If he is capable of detecting the fraud, he is also capable to construct some false evidence. Traces of the fraud on the voting webserver can be deleted by a single action. Since the webserver is connected to the internet, this deletion can be initiated from the internet.

Even worse, I can create counter measurements against this detection. As I wrote earlier, I only need to falsify 5% of the votes. I can try to select a certain group. I will target my fraud against this group and will commit no fraud by the other voters. The group of voters must have the property that they do not much know about computers. It should be unlikely that voters from this group will inspect the Javascript. If the identification of the voter is changed as I suggested then I have to select a group that not only will not check the Javascript, but also not check their vote (of course, this is much harder).

For this selection I have a few possibilities. If a browser makes connection with the voting webserver, the webserver obtains some information. This information contains the operating system of the voter, the browser of the voter and the IP-address of the voter. It is clear that I avoid any Linux users. Those people can actually start checking the Javascript program. Too many different people use Microsoft. So, I can't make any conclusion when the voter is using Microsoft. However, the group of Mac users might be interesting. These are often not developers, not people that like to see the inside, but want a nice look and feel. This group is also about 5%, just the percentage I needed.

However, after the first (Dutch) version of this article I got some feedback of the developers of RIES. They are all Mac users. Okay, point taken, not so good idea.

In the future their might be a significant group of mobile users. It is not likely that mobile users will do very complicated things on their small screens.

The other possiblity is the IP-address. For instance, I could take the IP-addresses of a very cheap ISP that has such a bad service, that the more professional computer specialist would never use it. And if that does not work, I could try to construct a database with IP-addresses of a certain group. For instance older people. This is not easy. I could try to collect this information on a website that is mainly visited by older people or a website that has also personal information about the visitors. Examples of such sites are certain webshops, dating sites or profile sites. Of course I need a contact there, but that is not high security. Unfortunately IP-addresses are often assigned dynamically with DHCP. This makes this trick less effective. But, as I said before, I only need 5%. And the situation might become better (for fraud) with the introduction of IPv6.

Anyways, an independent organizations that checks the Javascript, should take the computer of a disabled old granny of an etnic group, who did all kinds of senior webshopping. Of course, with the use of a Mac.

I also want to no the that it is important that the Javascript program is inline the webpage. There should be no use of the 'src' attribute of the '<script>' tag (if RIES uses this, then this should be changed). When the Javascript is inline, the voter can directly see the Javascript when the source code is requested. When the Javascript would be separately loaded, then the voter has to load this script again. The voting webserver could detect a second load from the same IP-address. In such case I let the webserver return the original Javascript. The normal behavior of the browser would be to show the Javascript from cache (showing the falsified Javascript initially loaded). However, I can suppress this behavior by setting the right HTTP headers.

Taking everything together, the conclusion can be made that with a single joker near the voting webserver fraud is not completely unthinkable.

A whole different strategy is to manipulate the result, by adding additional votes. With RIES the voter can check his vote afterwards, but for the people is hardly impossible to check whether any illegal votes are added to the votes list. To my opinion this check afterwards gives some false feeling of security.

Adding votes is limited, because it affects the turnout of the election. If i-voting would be introduced on a large scale, then an additional of 5% would not be suspicious. For the voting experiment only 20.000 voters did register of a possible 500.000. The number of registered voters could be altered to 100.000 (which is more than one seat in the parliament). However, it certainly would be suspicious when 80% of the votes would go to one party.

To do this fraud, I need a database of all voting codes (this requires one joker near the votes generator). Then, just before the closing of electronic ballot box, I need to add unused voting codes to the votes list. The votes list is published by the organization of the voting webserver. So, I need a joker near the voting webserver.

So, two jokers are necessary and that is rather impossible. However, it proofs that for the security we are dependent on the separation of the organization around the votes generator and the voting webserver. This fraud can not be detected by an individual voter or an independent organization from outside. So, the idea that everything can be checked afterwards, is not true.

It is possible to improve RIES on this issue. If the votes list is published every hour (without the reference list, we can't do much with it), then at the end voters can check whether a party got many votes in the last hour. Although, I could delay the votes of other parties to the last hour. Still, it will be suspicious that many votes are cast in the last hour.

Still, some more advanced tricks are possible. In general the same people will not vote. If I have a joker in two elections (possible with an interval of 4 year) near the votes generator. If I get a list of all voters with their voting codes. Based on the reference file I can construct a file with non-voters. In the second election I use the list of non-voters. With an automatic voting program I will cast the votes of the non-voters. For the voting webserver this will look as a regular vote. I can use a botnet, to prevent that this large number of votes comes from the same IP-address. In this way I can commit large scale fraud, with the use of only one joker. The constructed file of non-voters will probably not completely reliable. This will result in double votes (which will become invalid), where at least one vote goes to the candidate I have chosen. I can mask this by also cast some votes for other parties for people that are expected to vote. Those votes will also become invalid. Anyways, the success of this method depends on how reliable the database of non-voters can be made. The reliability can be improved by using more elections, but this means that it is a fraud for the long term.

The advantage of this fraud is that is rather safe. I only have to get data and I don't need to alter software or databases (with the risk of leaving traces). If I get the voting codes from my joker on a SD-card (a memory card used in digital camera's, not larger than a post stamp) and avoid any contact with my joker, then this is a nightmare for the person that has to investigate this fraud. Both my joker and I are rather safe. The joker does not have any program at home for casting the votes, while I will simply not be found.

A different way to add votes is to work around the security of RIES. As I wrote before, I can't enter the house, I can try to steal car parked in front of the house. Someone has to feed the votes generator with the list of voters. If I put my joker there. I extend this list with 5% "ghost voters". Those are voters that do not exist at all. When my joker receives back the list with voting codes, he will filter out all the ghost voters. This cleaned file is send to the printer (this is important, because otherwise the undeliverable voting cards of ghost voters will be send back by the post). An automatic program will use the voting codes of the ghost voters to cast votes of my choice (again using a botnet). In this way, I can commit fraud on large scale with just one joker.

In practice I think the method will probably not work. The person that delivers the list of voters does not need to be the same person that brings the voting codes to the printer. Also, some easy checks can be performed. Such as the number of printed voting cards, the number of send voting cards and the number of voting codes in the reference list must all equal. This information has not the problem of the "secrecy of the ballot". So, it can be checked by multiple election officials.

However, currently no rules are defined that such checks are really performed. So, this is a weakness. One should keep an eye on what the real source is of certain information and if a single person can compromise this information. Especially when information of voters becomes more centralized and redundancy of information is reduced, this might become a security risk. Also note that the fact that the voting cards are send by normal post is part of the security. If the voting codes would be send by email, then there is no need to filter out the ghost votes, before the list is send to the printer. Thousands of false email addresses could be used.

One could argue that ghost voters are also a problem with traditional ways of voting. However, there is a significant difference. With i-voting the actual casting of the votes, can be automated, while in traditional voting ways, the votes of the ghost-voters requires a 'body' to be present in the polling station. This makes it hard to do it on large scale.

Although fraud by ghost voters can easily be prevented by adding additional checks, it again proofs that this can hardly be checked by individual voters. For the experiment for the voter abroad about 20.000 persons did register. If this number would be 100.000 or 200.000 I would not be surprised. If RIES would be introduced for the whole Netherlands, than a voter could try to guess the number of legitimate voters. But this is not so easy. According to numbers out of the Wikipedia the turnout of the elections in 2002 weas about 300.000 more than in 1998. This amount is worth about 4 seats in the parliament. This proofs how difficult it is for the individual voter to check on ghost voters, especially when this is executed in steps of 100.000 ghost voters.

In general it can be concluded that, *an intrinsic weakness of the possibility to check the vote afterwards is that it will not detect any illegally added votes.*

RIES and the secrecy of the vote

The primary goal of the secrecy of the vote is to prevent bribery and intimidation. I will comment this aspect later. First I will look to the possibility that a list on the internet appears with the names of all voters together with their vote.

If that happens, then the result of the election is not invalid, since the votes were cast under the idea that everything was safe. The free expression of the will was not compromised. However, it will give a shock effect through society. It is likely that someone will go to the judge to ask to for complete ban on i-voting. There is good chance that the judge will decide in favor of the ban.

The motivation to do so, is much smaller than to falsify the results of the election (any motivation to get power, must be considered big). However, such action would get lots of publicity, is without violence and targets the image of the government and the responsible politicians. This is the perfect act of someone with strong anarchistic feelings.

To achieve this goal, a file is needed with all the names of the voters together with the voting codes . This file can be combined with the reference file to get a list of names with the real vote. If I think in jokers again, then I need one joker for that. The organization around the votes generator does possess this file for a certain period. This file has also to be send to the printer, so the printer also has access to this information.

I did mention that the security around the secrecy of the vote, can be lower than the security related to fraud. It is important to realize that this security is indeed no very high. If some other i-voting system would require identification (for instance an electronic passport) when casting the vote, this might give the perception that name and vote can be registered together. But, be aware, this is not much safer or more unsafe than the security of RIES in relation to the secrecy of the vote.

Furthermore, the secrecy of the vote is not temporary but the secret should be kept years after the elections. If 10 years after the election a list on the internet appears with the names and votes, then this will also give a shock effect through society. And now I return to the aspect of voting identification in RIES. The voting identification that is on the voting card is not anonymous. The name of the voter can related to the number. Although this number is probably not public, it is possible that a large number of people within the government have access to this information. Anyways, this information is outside the ‘security area’ of RIES (not classified). The identification of the voter becomes anonymous when it is encoded with the voting code. In the reference file there is a whole list of candidates that is encoded with the voting code. In theory (but not in practice, that is the idea of cryptography), the voting code can be found out of this list. With the voting code the non-anonymous voter identification can calculated.

Currently this is not possible, because the cryptographic method is ‘safe’ for this moment. However, RIES is dependent on the future. There is no guarantee that the cryptographic method will remain safe. This method can become unsafe, due to new mathematical ideas, improvement of computer power, grid-computing or quantum computing.

This may result in a funny situation, when someone within the government puts the list with names on the internet with the uuencoded voters identification (which is not highly classified information). This file together with the reference file is a gigantic and interesting cryptographic puzzle. Academic people around the world will love these kind of things. Students in their brilliant phase, can with an intellectual achievement become famous. “Who breaks RIES 2006?”. I am curious how long the cryptographic method will hold.

This is also for another reason a concern. I wrote earlier that a list of non-voters can be used to commit fraud. Breaking the code will also reveal the non-voters. By making the voter anonymous this way, RIES becomes unnecessary dependent on the future safety of the code.

The voting webserver does not register data such as the IP-address for reasons of the secrecy of the vote. This is an example where the secrecy of the vote is taken from a legal point of view. The IP-address can be used (with great effort) to find the owner of the computer (not the actual voter), but only with special authority. The organisation of the voting webserver does not have this authority. If a secret service wants to know the votes of the people, it can better tap the information by the printer. This is a much more easier and accurate way. The information of the IP-addresses can be used to detect the origin of fraud. If there is procedure to destroy this information, then this should be enough.

Putting in RIES in disorder

As I wrote before, the probability that someone wants to put the i-voting system in disorder is rather high. A DDOS attack is the most likely method. In the RIES system the voting webserver is rather is simple of design. No sessions will be maintained between webserver and browser. The voting webserver is not much more than retrieving some pages and the recording of the technical vote. This simplicity is of significance to withstand a DDOS attack. A DDOS attack can be made more sophisticated by recording all kinds of (invalid) votes. Since the voting webserver does not check on the vadility of the vote (only records it), the voting the votes list becomes very long.

Nevertheless, I think with the right measurements the DDOS problem is controllable.

Remains the possibility of disorder with the use of someone from inside the election organisation. If someone has obtained the file with all the voting codes, then he can place this file anonymously on the internet. If that happens, the elections have to be postponed.

Again a very attractive action for a anarchistic activist.

Instead of putting the file on the internet, all votes can be cast with this file. This will result that if a legitimate voter casts his vote, his vote will appear in the votes list together with the illegal vote. The total result will be called invalid.

I need one joker for getting a file with the voting codes. This can be someone in the organisation around the votes generator or someone of the printer.

The organisation around RIES

Until now, I looked to the technical aspects of RIES. Even important is the organisation that is executing the elections with RIES. If an organisation does not follow procedures, any security principles in the system can become ineffective. For instance, if a system is protected with a password, then this measurement is ineffective if all systems has word ‘SECRET’ as password or when the password is written down on a piece of paper. Therefore, I take a look how the Ministry organized everything around RIES.

The basic principles of RIES are that the concept is public, that certain files are published and the voter checks his vote. This will only work, if the Ministry actually explains how the system works and certain files publishes.

The Ministry did a good job on this point. The site 'www.kiezenuithetbuitenland.nl' contains explanation of the system and on this site the results will be published.

Another aspect of the security of RIES, is that organisation for the votes generator and the voting webserver are separated. On this point the Ministry did a less good job. From the legal framework and the rules it is unclear which measurements are taken to separate these two organizations. How it really is, can be much better than from the rules can be concluded, but formally I have to say that the organization is insecure.

Personally, I would prefer that any procedure for the votes generator are executed by the city of The Hague (which also handles the votes by letter) and the Ministry supervises the voting webserver. If the city locks the votes generator away, makes the publication of the reference file and if this is all codified in the legal framework, then the people can really see that those two organizations are separated.

A more closer look on the legal framework shows that there are still many challenges on this point. The legal framework is out of balance. According to the legal framework there are election officials that supervise the voting webserver during the voting time. However, there are no rules in the legal framework for the votes generator. In this article I showed that this is as critical as the voting webserver. The same counts for the information that goes to the printer.

There is a tendency to define supervising election officials in the legal framework, but leave out anything that has to do with technicians. Since security and functionality is moved from the hands of the officials to the technology, the situation is created that the legal framework describes 'ceremonial acts' of the officials, while the things that matter are outside the legal framework. For instance, I learned from a seminar on RIES that there is a 'technical judge' that must decide on complaints from individual voters who check their votes afterwards. This technical judge is not described in the legal framework. To me, this is a clear violation of the principle that elections may only be held by a legal framework. This principle is internationally recognized and codified in the Dutch constitution, article 59.

Also, the Electoral Council (Kiesraad), has no authority in case of the i-voting experiment. This became clear when I filed a complaint in the experiment of 2004, where the Council denied any authority. The Council does not have any technical member and is not capable to do some supervision.

There is currently no penalty laws that codifies punishment for violation of the secrecy of the vote. Probably, it is assumed that the secrecy of the vote is guaranteed. However, this becomes important with i-voting. The lawmakers forgot to codify the penalty rule in the law for the i-voting experiment.

The interaction of the voter with RIES

Until now, I looked to the technology and the organization around it. The final thing that has to be looked at, is the interaction with the end-user, the voter. The way a voter interacts with the system, may nullify the security aspects of the system.

As first I want to say, that the problem of voting by the internet is that the secrecy of the vote must be guaranteed, while it is desirable to have an open and verifiable system. These two requirements are conflicting, which makes it difficult to create a satisfying system. The choice of RIES is to give the voter the possibility to check his voter afterwards. This improves the possibility to verify the elections, but at the same time the secrecy of the vote deteriorates. If there would be no possibility to check the vote afterwards, then voting together behind the PC is possible. However, once the vote is cast and the browser is closed, the secrecy of the vote is sealed. When RIES is used, the voter can proof its vote in eternity if he keeps his polling card. The voting code is only a piece of information, however the possessor of the polling card with the voting code, is a strong evidence.

As I said in the beginning of this article, the secrecy of the vote was introduced in the 19th century to prevent bribery and intimidation. The primary concern was not to prevent a big brother government. There is no restriction on bribery anymore, with the possibility to check the vote afterwards. To my opinion, this is totally unacceptable. If we really want this, we have to discuss what we want with the principles of the secret ballot.

The second remark I want to make, is that the secrecy of the vote in relation with the government, can not be observed by the voter. This in contrast with casting a vote with pencil and paper.

Third, the government did not forbid that third parties provide tools that can cast the vote for the voters. The website of a political party, could provide such tool. Where voters can enter a voting code and the voter has only the choice of the party that provides the tool. The tool will do the final registration by the voting webserver. Such tools could also be placed on datingsites or social network sites. There might be even reasons to allow such tools, but the consequences of this are unclear.

Fourth, it is unclear how many people will actually check their voter afterwards. However, the designer of the system told that 10% of the voters checked their vote with the ‘waterschappen’. According to calculations of the designer, only 0,1% is necessary to add some security. I don’t want to make any calculations here, but I agree that 10% is very high and sufficient to detect any systematic fraud (if it can be detected by this method). The percentage can be improved by some ‘advertisement’ of the government. It is necessary to keep this percentage on a reasonable level for the long term.

Fifth, in RIES some actions are executed on the computer of the voter (Javascript). This has only some meaning when at least some voters actually check whether the right script is loaded from the voting webserver. This is highly technical procedure and the government did not publish any procedure how to do so.

Sixth, the general check on the reference file and the votes list is cumbersome. I have to write my own software to do the checks. Although, it must be said, it is enough if only one independent person checks the files. Still, I have the opinion that the software that checks the files must be open source. An independent organization should not rely on compiled code from RIES.

Seventh, I doubt whether the whole is simple enough for the average voter to have trust in. Three levels of trust can be distinguished for a person. The first level is that he can observe the security measurements by himself. This is the case of voting by pencil and paper using a ballot box. The level of trust changes, when the voter can not observe or does not have technical capabilities to conclude from an observation. The second level is when the voter can observe that independent people that have the position and technical capabilities to observe, trust the system. The third level of trust is when it solely depends on statements of the government. The third level should be avoided and will eventually put the government in trouble, even when the elections are fair and safe. By making things public, RIES aims on the second level of trust. To my opinion, the first level of trust can not be reached by any i-voting system. Note that a majority of trust is insufficient. If 10% of the people really distrust the integrity of election, you still have a problem.

Eight, it is unclear what should be done, when a voter claims that his vote is not registered correctly. There are no procedures for this situation. Should something happen by 1 claim? Or by 10, 100, 1000? And what can be done with such a claim? There is a technical judge that must decide on these claims. However, in many cases there is not much to check for this technical judge. This results in a yes/no game between the organisation of the election and the voter with the claim.

Ninth, the possibility to check the voter afterwards introduces possibilities to discredit the government. The party leader of a radical party can intentionally vote on another party. Then he can show on television, with his voting card in his hand, that his vote is falsified. Other voters that voted on another party, but changed their mind, can start to support him. They can take their voting card out of the garbage and also claim that their vote is falsified. This will come on television. People believe in aliens, UFO’s, crop circles and big complots.

The party leader of the radical party can make his trick more subtle. By voting twice, once on his own party and once on party he want to make suspect. This suggest that someone had access to the file with voting codes. In principle more schemes are possible. Those claims are almost impossible to rebut by the election organisation.

Tenth, according to the procedure the voter has to destroy his voting code after casting his vote and must directly decide whether he wants to keep his technical vote (for instance, by printing it). Any suggestions of fraud are likely to come after the electronic voting box is closed. That means, that these suggestions will come after the voter has decided to keep is voting code or technical vote. That means, that the honest voters (that do not believe in big complots), that did not initially intended to check their votes, do not have this possibility anymore when these suggestion come in the media. To my opinion it is better to suggest the voter to keep the voting code. It can be expected that any bribery or intimidation of the voter happens before the elections. So, keeping the voting code afterwards, does not make much difference anymore.

Conclusions

The following conclusion can be made:

- RIES contains the possibility to check the vote afterwards. This promises that integrity of the elections can be fully checked by the voters. Although this check enhances the security, it does not fully fulfill this promise. The integrity of the election is still dependent of the integrity of the elections organisation using RIES.
- The split up of the system in a votes generator and a voting webserver introduces additional security. However, it is incorrect to assume that fraud can not be committed on one of the parts alone.
- The method of making the voter anonymous must be reconsidered. Currently this is achieved by encoding the voter identification with the use of the vote code. I showed that this is problematic for two reasons.
- The generation of the voting codes using one master key, is rather dangerous.
- The security of the secrecy of the vote is rather low.
- The secrecy of the vote is based on trust and can not be observed by the voter.
- Using a fully automated system for voting, introduces many challenges to the legal framework. In the experiment this was not recognized. Codifying ceremonial acts for the election officials, while technicians and technical issues that matter are outside the legal framework, is a very undesirable situation.
- In theory the security of RIES is not dependent on the possibility to see the source of the software (open source). But in practice it is. The Javascript program and the code to process the voting results should be open source.
- The possibility to check the vote afterwards, conflicts with the principles of secret ballot. There is no limitation anymore on bribery and intimidation.
- It is unclear what should be done when a voter claims that his vote is incorrectly registered.
- The possibility of checking the afterwards can be used to discredit the government.

I already wrote that using RIES for voters abroad introduces little risk, just because there is not enough gain for someone that want to commit fraud. The biggest danger I see is that after some years the election organization becomes sloppy in following procedures. I would say, forget all my points and codify in law that every 5 years the security must be investigated.

It is a different story when politicians want to introduce i-voting on a large scale. The check on the vote afterwards is a strong deterioration of the secret ballot. A debate about this issue is necessary. Furthermore, I showed that there are still many areas where fraud is possible. If the Dutch government want to continue with i-voting, it must realize that it is necessary to make a ‘high security area’.. This area includes the votes generator, the printer and the voting webserver. A body search is necessary for the technicians entering the area. SD-cards as big as a stamp are a security problem.

A final word

In discussion about i-voting people sometime tell me “then *you* go to the polling station”. My vote has no value when my vote is cast correctly, but where the election in general is a mess. That is why I value that all votes are cast according to democratic values.

Lucas Kruijswijk
Finished his study computer science on the Vrije Universiteit of Amsterdam in 1992.
The study included cryptography.