An Analysis Of The Neff’s Voter Verifiable Election Scheme

Bachelor Thesis of Veneta Velyanova

Department of Mathematics

Supervisor: Prof. Dr. J. Buchmann

14. Februar 2008
Inhaltsverzeichnis

1 Introduction 4
  1.1 Motivation .............................................. 4

2 E-voting Basics 7
  2.1 Voting Phases ........................................... 7
  2.2 Security objectives in e-voting ........................ 7
  2.3 Technical components .................................. 9

3 Overview of Internet Voting 12

4 Neff’s cryptographic protocol 14
  4.1 Features .................................................. 14
  4.2 Overview of the election procedure ................. 15
  4.3 Neff’s scheme ............................................ 16
  4.4 Possible Attacks ........................................ 20

5 Obstacles to Adoption 21

6 Conclusion 22
Erklärung

Hiermit versichere ich, dass ich meine Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Darmstadt, den 14.02.08

Veneta Spasova Velyanova
1 Introduction

1.1 Motivation

Voting is a vital expression of the people’s power which needs to be secure and secret. The vote is how every single citizen can wield real and immediate power. The simple act of marking a ballot tells our leaders what we think about decisions that affect our lives, such as how much taxation we think is fair or what issues (such as health care or the environment) we think are most important.

Free and fair elections are of critical importance in a democracy. It is important that everyone can vote without interference, safe in the knowledge that it will be counted. Recently there have been major initiatives to ‘modernise’ the voting process across the world. Electronic voting with precinct-based electronic machines has been suggested as a potential improvement over the way we currently conduct elections.

E-voting allows a voter to record his secure and secret ballot electronically. It has been introduced for a number of reasons. Some have implemented electronic voting machines in an effort to speed up ballot counting. Once an electronic ballot is cast, the count can be done in an efficient and accurate manner. There is also a demand for better accuracy. In the 2000 USA presidential election, it is estimated that between 4 and 6 million votes were lost, mainly because of “unreadable ballots”[1]. Electronic elections have the potential to offer high accuracy, combined with verifiability and error recovery. An electronic user interface can alert the voter of and prevent under- or over-voting, as well as facilitate voting for those with various disabilities. With the years, the number of people who vote in an election has been steadily decreasing and some have argued that electronic voting would increase the turnout[2]. The younger generations are probably turned off by the idea of voting with a pencil and piece of paper but they would find voting more interesting if the casting of the ballot involve computers. The cost reduction is also a major benefit of e-elections. They eliminate the expense of printing ballots and avoid manual work. In India the Election Commissioner stated that the electronic voting saved 8000 tons of paper during the 2004 elections[3].

A growing fraction of the society has access to Internet. Therefore if internet voting is an option, the access to the process will be much easier for a number of people. It is very likely to increase voter convenience and therefore the potential voter turnout. One of the reasons people are interested in internet voting systems is that they can be mobile. People are living increasingly busy lives with growing work and family commitments. Having to
go to an old school to vote is difficult to fit into the day. By Internet voting people would have several ways to vote. They could easily cast votes from computers in their homes, offices, schools, and libraries or from centralized polling places. The process will be calculated quickly and efficiently, with less chance of human error.

However, if not carefully designed, electronic voting systems can be easily compromised, thus corrupting results or violating voters’ privacy. The major complaint is that there is no true way of controlling, tracking, or auditing the behavior of an electronic machine. The info about the way electors voted (which button they pressed or which part of the screen they touched) is collected and stored in the form of anonymous intangible human-unreadable string of bytes. Voters currently have no way of ensuring that their vote has been recorded correctly or counted, and the public has no way of verifying that the tally was computed correctly. Such computer procedures are not verifiable by humans as we are not equipped for verifying operations occurring within an electronic machine. Thus, for people who did not program them, computers act just like black boxes and their operations can truly be verified only by knowing the input and comparing the expected output with the actual output. To have confidence in the election results, people want to trust that the election tasks are performed properly.

Recent official selections show that computers are insecure and unreliable. Manipulation of electronic data or machinery can potentially be done remotely and efficiently, and could affect more votes than an attack on physical ballots or machinery at a polling station. Furthermore, an attack on an electronic voting machine has a higher chance of going completely undetected, due to the lack of an auditing mechanism. Numerous reports of election irregularities surrounded the 2006 congressional elections[4]. These included voter fraud, efforts to suppress turnout, problems with electronic voting machines, and reports of illegal calls around the country. That shows that computers could be manipulated to affect the output of the elections. With today’s hardware and software architectures, a malicious code on a voting client can actually change the voter’s vote, without the voter or anyone else noticing, regardless of the kind of encryption or voter authentication in place. This is because the malicious code can do its damage before the encryption and authentication is applied to the data. The malicious module can then erase itself after doing its damage so that there is no evidence to correct, or even detect the fraud. Any application that users are lured into downloading can automatically install a Trojan horse (a program that acts maliciously pretending of doing something else) that later interferes with voting. This includes browser plug-ins, screen savers, calendars, and any other program that is obtained over the Internet. There are dozens of software vendors whose products run on many peoples’
home machines. For example, there are millions of personal computers running Microsoft office, Adobe Acrobat, RealPlayer, WinZip. Denials of server (DOS) attacks are also possible.

In my bachelor thesis is focused on the problem “How can we trust computers, when we vote via Internet”. For that purpose it will be discussed a cryptographic model that offer the promise of verifiable voting without needing to trust the software in the system. Many electronic voting systems were proposed so far. Many of them assume that the machines used for voting are honest. This approach seems to be unsuited for implementing electronic elections - one would require a detailed audit at least of the operating system and of the application used for voting. Such a verifications of voter’s hardware and software is practically impossible. Secure electronic voting is fraught with many problems, including the security of voting machines and their millions of lines of private codes. Many have suggested equipping the existing electronic voting machines with printers, to produce a receipt of each ballot. These receipts would be retained by the polling station. Although this does provide a way to conduct a recount, thereby creating an auditing mechanism, these paper audit trails “provide, at best, a short term fix to a fundamentally flawed approach”[5]. There are schemes that provide a privacy preserving voter-verifiable election system that exclude vote buying, vote selling and threat from a cheating party. Andrew Neff has proposed a cryptographic scheme that uses receipts and zero-knowledge protocols that allow voters to prove that their votes are accurately recorded and at the same time voters cannot be coerced to vote in a particular way. In all stages of Neff’s Voter-Verifiable election scheme is possible to audit each step, detecting error or fraud. The advantage of having these verification procedures is that any post-election question about its legitimacy is answered in the form of proof. If the election is proven to be trustworthy, then there is no possibility for controversy. As a result such systems do not require voters to trust computers when they vote since voters can detect malicious behaviour. Although these schemes represent a significant security improvement over current paperless systems that use a direct recording electronic voting machine(a machine that refers to an electronic device whose purpose is to record votes), there are a number of potential weaknesses arising from the fact that they are not fully specified at systems and human interaction level which became apparent when considered in the context of an entire voting scheme.

In the following section an overview of the different e-voting phases and requirements is provided and the cryptographic techniques that are used in Neff’s scheme are outlined. In section 3 is presented the existing problems of Internet voting. Neff’s Voter-Verifiable election scheme is explained in section 4 and analyzed, based on the requirements defined. Chapter 5 discusses the
non-technical obstacles that impede the implementation of Neff’s scheme in real elections.

2 E-voting Basics

Electronic voting promises the possibility of a convenient, efficient and secure facility for recording and tallying votes. Generally speaking, e-voting refers to both the electronic means of casting a vote and the electronic means of tabulating votes. There are a wide variety of e-voting set ups, ranging from the casting of the vote with the aid of an electronic device (voting machines) inside a polling station to casting a vote anywhere outside the polling station at a PC and transmitting the vote via the Internet.

2.1 Voting Phases

Each election involves four distinct stages:

Registration: In the registration stage the authorities determine who is eligible to vote, maintain proper lists of the registered voters and provide them with voting certificates that will enable them to participate in the tally. During a certain complaining period, voters should be able to post objections. After it the final version of the electoral roll is published by the voting authority.

Validation: Once the election begins, voters are authenticated before casting their votes. Also, during this phase the voter will obtain the necessary credentials to allow him/her to participate in the next phase. This may involve asking voters for identification cards or passwords.

Casting: During that stage, voters cast the desired ballots using the communication facilities of the network.

Tallying: At the end of the voting phase, the ballot collecting authority stops accepting ballots. The counting process is initiated. Finally the tally is published and made available through the network.

2.2 Security objectives in e-voting

Electronic voting systems may offer some advantages over traditional voting techniques. But these complex systems are facing a number of compulsory
security requirements. If not carefully designed, electronic voting systems can be easily compromised, thus corrupting results or violating voters’ privacy. What does it mean for an electronic voting system to be secure? That system should provide security properties that promote confidence in the election. At the end the result of the election should be consistent with the unmanipulated free will of the eligible voters. I define security rigorously by describing a set of security properties. Thus, the security of a particular scheme is measured by the degree to which it adheres to these defined requirements:

**Accuracy:** A system is accurate if it is not possible for validated votes to be altered or eliminated from the final tally, and it is not possible for an invalid vote to be counted in the final tally.

**Democracy:** A system is democratic if only authorized voters are able to vote and it ensures that each eligible voter can vote only once.

**Anonymity:** A voting scheme is private if neither election authorities nor anyone else can link any ballot to the voter who has cast it.

The notions of receipt-freeness and uncoercibility were introduced to deal with vote selling and coercion in e-voting systems.

**Receipt-freeness:** Receipt freeness is a strong form of privacy, where the secrecy of the ballot is maintained even if a voter cooperates with an adversary. A scheme is receipt-free if it is impossible to reconstruct a provable receipt of a vote outside the voting booth. It is important to emphasize that this property does not prevent the creation of a receipt in the voting booth, as long as it is not readable outside the booth. Instead, it does not allow a voter to possess or create proof which can be used after the election to prove to a coercer that he voted in a certain way.

**Uncoercibility:** The precise definition of uncoercibility is that each voter must be able to decide his intention. This requirement is important for prevention of bribery and disturbed voting for some unknown reasons. As a result, verifiable vote buying is impossible. Both definitions are similar because they deal with the fact that votes may be bought and sold or voters may be intimidated in order to cast a particular vote. By receipt-freeness the voter should not be able to extract the value of the vote, even if the voter wants to (e.g. for reward). With uncoercibility, the adversary is the coercer; the coercer should not be able to extract the value of the vote from the voter, even if the voter is forced to (e.g. threatened).
Fairness: That property requires that all ballots stay secret until the voting phase has ended. The objective is to prevent anyone from knowing intermediate results of the voting. Such knowledge could be used to affect other voter’s behaviour.

Verifiability: A scheme is verifiable if any observer can be convinced that the election is accurate and that the published tally is correctly computed from votes that were correctly cast.

Robustness: The voting scheme should have the capacity to tolerate partial failure or malicious behaviour by any reasonably sized coalition of parties (voters, authorities, outsiders) at any stage of the translation process. The scheme should be able to continue the election without termination and produce an accurate tally regardless of these failures. Furthermore, when error or fraud is detected, the voting scheme should have a mechanism or procedure in place to contain and, ideally, correct the error. Note that it may be impossible to isolate the exact votes which were corrupted without sacrificing anonymity.

2.3 Technical components

After outlining the security objectives of an e-voting system, I want to discuss some cryptographic techniques that try to satisfy them when we vote via Internet.

Anonymous channel: Every electronic voting scheme relies on some type of communication channel(s) between the voters and other players in the election. These channels can either be realized through cryptographic or physical means, depending on the circumstance. An anonymous channel is a communication channel between two parties where the sender of a message remains anonymous.

Mix-Net: The first paper to introduce the idea of a mix-net, as well as an electronic voting protocol altogether, is that of Chaum in 1981 [6]. The goal of a mix-net is to accept a set of inputs and anonymize them via a secret shuffling process, such that the outputs cannot be traced back to their corresponding inputs. This creates an anonymous channel. In an application of a mix-net to electronic voting, the inputs are encrypted votes, and the outputs are the corresponding plaintext votes. There are \( n \) (Mix) Servers, or Trustees which lie between the inputs and outputs of the mix-net. Each
Server has a private and a public encryption key. The first Server received the messages, partially decrypts each vote in the set with its own private key and performs a secret shuffle to the set of partially decrypted votes. Then the Server forwards all of the votes to the next Server, who functions in a similar manner, until the last Server in the mix-net has fully decrypted each vote. The result is an untraceable path from input to output. In the context of voting, that means that it is impossible to reconstruct the one-to-one correspondence between voter and vote, thus preserving anonymity. In any mix-net scheme, it is imperative to verify the actions of the Servers in order to ensure integrity of the decrypted votes. To do this, there must be an auditing process. The Servers must produce proofs of correctness of their computations. Achieving this, while still maintaining anonymity of the votes, is difficult. The process is inherently inefficient, and many attempts have been made to produce methods to increase the efficiency of this step[7].

**Randomness:** All messages need to be encrypted in such a way that successive encryptions of the same message give different result. In that way the cryptographic security of the encryption process is increased. This can be easily achieved using enveloping procedures where a random session key is involved such as the randomness used in El-Gamal encryption. That is, to each plain text $m$ is added a random parameter and encrypted with the public key. In that way same messages have different encryptions because of that randomized factor.

**Multiple votes:** Many Internet voting schemes allow a voter to cast a vote only once (or from a single machine). There are still fears that an online ballot makes it far easier to influence elections. If people do not go to a polling station, you cannot tell who is using whose ID card or if a voter is being put under pressure when they cast their ballot. A solution to this problem was implemented in the Estonian Internet voting system[8]. It is allowed multiple online votes to be cast, with each subsequent vote cancelling out the previous one. In that case voter can revoke the previous ballot and cast a new one. The main problem of this system is that it provides no verifiability of the election results and that vote selling is possible.

**Blind signatures:** E-voting schemes based on blind signatures are closely related to those based on a mix-net. The concept of blind signatures was introduced by Chaum [9] as a mechanism that allow some party to get a message digitally signed by another party, without revealing any information about the message to the signer. The effect is similar to placing a document
and a sheet of carbon paper inside an envelope. If somebody signs the outside of the envelope, they also sign the document on the inside of the envelope. The signature remains attached to the document, even when it is removed from the envelope. Blind signatures may be useful in a voting protocol to perform the registration stage - where the registrar signs the ballot of a voter (after verifying the voter is eligible), without knowing its content. Then the signed ballot may be anonymously sent to the tallier who can then verify the signature and count the ballot. A blind signature is secure if it can be proven that the identity of the holder of the signature is never revealed nor the content which is signed. The unconditional anonymity of the holder of the signature must be guaranteed even in the case of collusion. This is known as the blindness property[10]. For a blind signature to be secure it must also be proven that the blind signature cannot be forged. Even if a number of blind signatures are collected it must still be impossible for an attacker to forge the signature. Formally stated this means that if you have received \( j \) blind signatures, it is impossible to compute signature number \( j + 1 \). This is known as the non-forgability[11] property. An advantage of blind signature election schemes is that their communication and computation overhead is fairly small even when the number of voters is large. However the following problem can arise: Without knowing the contents of the messages the validator can sign very dangerous strings. To prevent such attacks, cut-and-choose techniques are used. Cut-and-choose techniques are based on sending a certain number, say \( p \), of blinded messages. The validator chooses at random one of the messages and asks the other party to reveal the blinding factor of all the other messages. By checking that all \( p - 1 \) messages are not offensive, the validator is convinced that the message that is still blinded is inoffensive too.

**Threshold cryptography:** Threshold cryptosystems distribute the functionality of cryptographic protocols to establish robustness. In the election, the tallying process can be shared among \( n \) voting authorities by using a threshold public-key encryption system. In this case there is only one public encryption key, while each of the \( n \) authorities has a share of the private decryption key. Each voter posts his/her vote encrypted with the public key of the authorities. The final tally is decrypted by the voting authorities jointly. Privacy of the votes and accuracy of the tally are assured provided at least a threshold of the authorities is not faulty (or corrupted).

**Interactive proofs and zero-knowledge proofs:** Informally, an interactive proof is a protocol between two parties in which one party (the prover),
tries to prove a certain fact to the other party (the verifier). An interactive proof usually takes the form of a challenge-response protocol, in which the prover and the verifier exchange messages and the verifier outputs either “accept” or “reject” at the end of the protocol. It is useful for interactive proofs to have the following properties, especially in cryptographic applications. The verifier always accepts the proof if the fact is true and both the prover and the verifier follow the protocol. The verifier always rejects the proof if the fact is false, as long as the verifier follows the protocol. The verifier learns nothing about the fact being proven (except that it is correct) by the prover that he could not already learn without the prover, even if the verifier does not follow the protocol (as long as the prover does). A zero-knowledge proof is an interactive proof that allows a prover to prove the knowledge of a secret to a verifier without revealing it. The verifier can try any strategy it likes to learn the secret but after any possible interaction, it learns nothing more than it could have learned from the common input taken by itself. Not all interactive proofs have this property. Existing ZKPs are iterative in nature, their protocols require multiple communication rounds. A typical round in a zero-knowledge proof consists of a “commitment” message from the prover, followed by a challenge from the verifier, and then a response to the challenge from the prover. The protocol may be repeated for many rounds. Based on the prover’s responses in all the rounds, the verifier decides whether to accept or reject the proof.

3 Overview of Internet Voting

Over the past decade or so, researchers have given serious consideration to remote voting over the Internet. By Internet voting the vote is cast over the Internet and the voting client is unsupervised during voting (the voting client may be at home, at work, in a library, etc.). Registration may be either physical (at the elections office) or electronic (with some form of digital identification). Validation, casting and tallying are electronic. Utilizing the vast capacity of the internet has multiple perceived benefits, including increased convenience and flexibility, along with reduced cost and labor. The hope is that i-voting may also result in an increased voter turnout.

Certainly, internet voting is an attractive possibility. However, there are some serious obstacles that prevent the facilitation of a secure election over the internet. The foremost obstacle is the same as that of any remote voting system: the privacy of the voter at the moment of vote casting cannot be guaranteed in a remote setting. Threats of vote buying or voter coercion
are a serious concern and violate the anonymity requirement. Even with the ingenious solutions that cryptography provides us, there is no way to ensure that a voter will vote in physical privacy in an internet based election (or in any other remote voting based election).

An another problem stems from the fact that a scheme must deal with the existing platform of the internet itself. In practice, the internet is simply not secure enough to use in an election. The reader may wonder why an election cannot be conducted satisfactorily since we have an (arguably) successful implementation of financial transactions over the internet. It turns out that elections differ from financial transactions in a number of critical ways. First, the receipt-freeness requirement of elections prevents the creation of take-home receipts (unless they are encrypted), which are crucial for disputes in financial transactions. Second, because an election is conducted during one day only, there are time constraints in real elections that are not as pertinent in financial transfers. For this reason, denial of service attacks over the internet are a major threat. During a time-restricted election, the density of attempted attacks on the election servers would likely be greater than regular attacker behavior on other sensitive servers. The stakes of an election are sometimes worth more to certain individuals or groups than a major financial transaction. While a denial of service attack is not likely to change the results of an election, it can certainly cause a significant disruption to the process. For these and other reasons, voting over the internet is a hard problem.

Other possible internet-based attacks include web site spoofing, automated vote buying, and PC viruses. Each of these is a real and potentially catastrophic threat. More potential threats include man-in-the-middle attacks, insider attacks, software bugs, and client-side computer vulnerabilities [12]. For any attack on an i-voting system, detection is more obscure and even if detected, voter confidence would be greatly affected.

The lack of a Public Key Infrastructure, or PKI [13] cause also problems. Authenticating an eligible voter before giving him the appropriate vote-casting privileges over the internet often requires the use of the voter’s own public/private key pair. This assumes that each voter has a cryptographic key and that the entire infrastructure is secure and widespread, which is not the case today. Distributing keys to all eligible voters is a difficult problem that is possible to overcome, yet unlikely to be solved in the near future.
Another problem is that the testing and certification of internet voting may be much more difficult. Not all the equipment, such as operating systems and browsers, will be controlled by the election board. There is increasing disparity, rather than standardization, in the use of these products.

4 Neff’s cryptographic protocol

Thus far, there is no practical, provably secure e-voting scheme that can be trusted in a large-scale, binding, governmental election. The problem with electronic, especially internet voting systems is that the voting public has no good way to tell whether votes were recorded or counted correctly because they have no complete trust in the voting machines. But there are protocols which offer the promise of verifiable voting without needing to trust the voting machines. One of them is the cryptographic protocol by Andrew Neff. That scheme is an e-voting protocol that makes excellent use of cryptography in order to achieve a level of transparency and verifiability, while still meeting many of the practical concerns that most remote e-voting systems lack. The main idea of that protocol is that it allows voters to verify that their votes have been accurately recorded, everyone can verify that the tallying procedure is correct, preserving privacy and coercion resistance in the process. As a result the reliance of the voter on the software and hardware of the computers used for voting is limited.

4.1 Features

Due to the widespread controversy over the trustworthiness of the current voting schemes, Andrew Neff’s protocol is focused on creating a level of transparency in elections. If a vote is cast by use of a machine, either in a polling station or by the Internet, the vote is processed and stored, and there is no straightforward method for checking that the votes are stored without any kind of alteration. The voter may distrust the security mechanism of mixing and counting the votes. Therefore, one of the important features of a protocol should be to provide the voter a way to verify that her vote has been accurately counted and included in the final result independently from the problems on hardware and software level. That is one of the central ideas in Andrew Neff’s protocol. In that scheme the voter is involved in the verification process and what distinguishable is that at the end either the voter becomes convinced that the voting machine works how it is expected or he can detect cheating and frauds. That property that each vote is cast-as-intended assures coercion resistance. Moreover the receipt that proves that the vote is correct-
ly included in the tally is unreadable outside of the voting booth. That’s why the voter is not able to sell his vote and can not be threatened to vote in a particular way. So, problems like vote buying and vote selling are limited to a great extent. This is a clear advantage over any scheme that either relies on lengthy manual recounts for election verification, or one that does not meaningfully verify results at all (like many current DRE-based schemes).

4.2 Overview of the election procedure

Andrew Neff has proposed a cryptographic voting protocol for use in DRE (Direct Recording Electronic) voting machines. The election process involves the following steps:

At the voting machine, an authenticated voter selects his desired candidate and submits it as his vote. Then the DRE constructs a verifiable choice (VC). That VC is the encrypted electronic ballot that represents the voter’s choice (cf. Fig. 1). It is a \((n \times l)\) matrix of ballot mark pairs (BMPs). The parameter \(n\) corresponds to the rows, one row per candidate and \(l\) is a security parameter.

As a next step the voter gets a receipt from the DRE. That receipt contains the encrypted vote and the pair \((BSN, \text{hash}(VC))\), where BSN is a unique ballot sequence number of the voter’s ballot that is useful in the verification process. With the help of the receipt the problem of untrusted voting machine can be solved. Neff solves that problem and it is remarkable that the proof which the voter gets for his choice is not like an ordinary receipt because it convinces only the voter for the correctness of the process and at the same time does not contain any information that can be used outside the voting booth. For that reason Neff’s protocol remains receipt free.

DRE performs an interactive zero-knowledge protocol with the voter. The idea is to convince the voter that the row that corresponds to his choice will be interpreted as a vote for the “right”candidate during tallying. Therefore for each BMP which appear on the selected row DRE provides a pledge bit. As a next step DRE prints all the pledges on the receipt. The voter receives the pledges and sends his challenges. With these challenges the voter asks the DRE to provide a proof that the ciphertext that he has chosen for his challenge indeed decrypts to the value of the pledge that the DRE has provided for that BMP. That protocol is repeated for all the \(l\) BMPs on the chosen row.
The DRE constructs a fake proof for all the other candidates. It is necessary because otherwise the information on the receipt would reveal which row is selected. If there is no proof for the unchosen rows the receipt would contain pledges and challenges only for the selected row. The fake proof has a different order from the proof for the “right” row. At the first step challenges are printed and after that the pledges.

The DRE then constructs an opened verifiable choice (OVC) according to the voter challenges and submits it to the bulletin board. OVC is formed from the VC and it is quite similar to it. The difference is that every BMP in the row of the chosen candidate in the OVC is half-opened according to the challenges of the voter.

Later, the voter can check that the OVC that is printed on the receipt does appear on the bulletin board and matches the hash values printed before, and that the OVC contains valid openings of all the values which are pledged in the locations indicated by the challenges printed on the receipt.

After the election is closed, the trustees apply a universally verifiable mix net to the collection of posted ballots.

4.3 Neff’s scheme

I present Neff’s scheme following the several steps explained above and divide the description of the scheme into the following main sections: receipt construction, performance of the interactive zero knowledge protocol and receipt verification.

The VC that appears on the receipt contain BMPs. Each BMP is a pair of plain texts \(b_1, b_2\) in \(\{0,1\}\) which are encrypted with the help of a random parameter (Section 2.3). Thus each BMP is a pair \([b_1 \ b_2]\), an encryption of \((b_1, b_2)\). The BMPs in the selected line must be of the form \([0 \ 0]\) or \([1 \ 1]\). The right and the left position of one BMP in the selected line are encryption of the same message but because of the randomness used the encryption doesn’t look the same. In this way the receipt could not reveal which row is selected. In contrast, the format of the plaintexts in the BMPs in the unchosen rows must be of the form \([0 \ 1]\) or \([1 \ 0]\).
The receipt should help the voter to verify the way he voted. Almost all the existing protocols require complex computation on the part of the voter (infeasible for an unaided human). Thus, they require the voter to trust that the computer actually casting the ballot on her behalf is accurately reflecting her intentions. Neff’s protocol proposed verifiable receipt-free voting schemes that overcome this problem. It involves voters in the process of checking the accuracy of the recorded votes such as that voter can detect cheating and errors without needing to trust the computer. The idea is to convince the voter that the row that corresponds to his choice does indeed contain a set of BMPs of the form $b \overline{b}$. The DRE performs an interactive zero-knowledge protocol with the voter to prove that the encrypted ballot corresponds to the correct candidate. As a result each vote is either counted-as-cast or the fraud is detected. The proof works as follows: For each BMP which appear on the chosen row, DRE provides a pledge bit. That pledge has the same value like the bit $b$ used in the particular BMP. For example, if this BMP has been correctly formed as $b \overline{b}$ the DRE can always convince the voter by using the value $b$ as a pledge. Then the DRE asks the voter for his challenges. That challenge is a bit string where the $k$-th bit equal to 0 means open the left element of the $k$-th BMP and 1 means the right one. Open the left element means that the voter wants a proof that the left element of the particular BMP corresponds to the pledge that is provided for that BMP (cf. Fig. 2). In other words, If DRE is cheating and this BMP contains $0 \overline{1}$ or $1 \overline{0}$ the voter has a chance $\frac{1}{2}$ of detecting this. By repeating the protocol for each of the $l$ BMPs in row $i$, the probability that a malformed row escapes detection is reduced to $\frac{1}{2^l}$.

![Fig.2 Prove that selected line $i$ has correct design](image-url)
Neff’s protocol makes ingenious use of zero-knowledge arguments. The DRE output a zero-knowledge proof for every other unchosen candidate. They are indistinguishable from the “real” ones and differ only in the order they are conducted.

The rows corresponding to the other candidates contain BMPs of the form $\begin{bmatrix}0 & 1\end{bmatrix}$ or $\begin{bmatrix}1 & 0\end{bmatrix}$. In order to avoid inaccuracy during the verification, it is necessary that the challenge to each BMP match the pledge of the DRE. The DRE should know whether the left or the right part of the BMP to select as a pledge. That’s why for each unchosen row the DRE selects an $l$-bit challenge uniformly at random and after that commits the pledges and opens the BMPs according to the challenges (cf. Fig. 3).

Even if the voter does not trust his own computer, it is enough that someone with a good implementation of the verification algorithm performs the check. The voter only has to make sure that he gave the challenge for the real candidate after the DRE was committed, and that the challenge printed on the receipt matches what he gave. Everything else can be publicly checked outside the voting booth. Since no one can tell from the receipt in which order the commitments and challenges were made, they cannot be convinced which of the proofs is the real one. That’s why the receipt cannot be used to prove to a third party how the voter voted and in that way the protocol remains receipt-free.
Fig. 3 *Fake proof for unselected lines j*

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>l</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
</tr>
</tbody>
</table>

DRE (Challenge):  
right  right  left  right

DRE (Commitment):  
0  1  0  ...  0

DRE (opens):  
1  0  1  0  ...  1  0

As a next step the DRE constructs an opened verifiable choice (OVC) according to the voter challenge and submits it to the bulletin board. OVC is formed from the VC and is quite similar to it. The difference is that every BMP in the row of the chosen candidate in the OVC is half-opened according to the challenges of the voter. That means that one of the two bits that are contained in one BMP is opened. The opened encryption of the bit $b$ contains both the pledge bit and the random parameter $w$ used for the encryption of the $b$ which appear on the VC. That random parameter makes the encryption of the bit $b$ unique and without it there is no way to get the same encryption which appears on the VC although the encryption function is the same. Remark that the OVC reveals no information about the choice of the voter. It contains only information that should match the hash (VC) on the receipt and should contain valid openings of all the values pledged to in the locations indicated by the challenges. Otherwise there would be errors or cheating that the voter can detect.

After the ballot of the voter has been recorded on the public bulletin board, the voter may use his receipt to verify his vote was cast as intended and is accurately represented in the election results. The receipt contains all the information described (BSN, hash (VC), pledges, challenges) which are needed for the verification. Everyone is able to encrypt the pledge bit with the help of the random parameter that is contained on the OVC and match the result with the cyphertext that appears on the VC. So every kind of inaccuracy can be detected.

Voters are also given the option of taking either a detailed or basic receipt. The detailed receipt contains all the information described (BSN, hash (VC), pledges, challenges), but a basic receipt contains only the pair (BSN, hash (VC)), which are printed by the DRE immediately after the voter has made his vote. This decision is made separately for each race on a ballot.
There are problems that arise in the way the protocol is conducted. If the challenges for the unchosen rows are selected randomly by the DRE the following problem arises. A vote buyer could tell the voter in advance to vote for candidate \( i \) and use some fixed value for the challenge, and the voter could later prove how he voted by presenting a receipt with this prespecified value appearing as the \( i \)-th challenge. To achieve coercion-resistance voter can specify additional challenges for unselected lines and overwrite the values that are already selected by the DRE. In this way the voter can pretend that he voted for Candidate A by setting the coercer’s challenge as he is forced to and to use another challenge to cast his vote. In this way he can convince the coercer that he has used his value but at the same time his vote is cast-as-intended.

The receipt contain the challenges for all \( n \) rows and each challenge contain \( l \) bits according to the fact that there are \( l \) BMPs on each row. It can happen that the challenge of the voter is the same as another challenge on the receipt. In that way the voter cannot be sure which row is counted for the “right one”. Therefore it is necessary that all \( l \)-bit challenges are different from each other.

### 4.4 Possible Attacks

Attacks launched by malicious DREs and attacks launched by malicious tallying software can be detectable but irrecoverable. In a ballot deletion attack, a malicious DRE could erase voters’ ballots or submits random bits in their place. Election officials and voters can detect this attack after the close of polls, but there is little they can do at that point. Since the electronic copy serves as the only record of the election, it is impossible to recover the legitimate ballots voted on that DRE. Neff’s scheme use ballot sequence numbers (BSNs) to uniquely identify ballots. BSNs enable voters to find and verify their ballots on the public bulletin board, and by keeping track of the set of valid BSNs, election officials can track and audit ballots. For example a DRE submits multiple ballots with the same BSN. Election officials will be able to detect this attack after the ballots reach the bulletin board, but recovery is difficult. Suppose a DRE submits 50 valid ballots (i.e., from actual voters) and 50 additional ballots, using the same BSN for all the ballots. The talliers are not able to distinguish the invalid ballots from the valid ones. Another example is when a malicious DRE “steals” BSNs from the set of BSNs it would normally assign to legitimate voters’ ballots. For a particular voter, the DRE might submit a vote of its own choosing for the BSN it is supposed to use, and on the voter’s receipt print a different (invalid) BSN. Since the voter will not find his ballot on the bulletin board, this attack can be detected, but recovery is nontrivial because the voters’ legitimate ballots
A social engineering attack, an adversary attempts to fool the other side into voluntarily revealing a secret. In that protocol the DRE fools the voter into revealing his future actions. Suppose a DRE triggers a reboot after it learns a voter’s random challenge $c_i$ for his chosen row, and then restarts the protocol, feigning an error. If the voter chooses the same challenge $c_i$ again, the DRE can undetectably forge a ballot for a different candidate by constructing row $i$ as an unchosen row consistent with that challenge $c_i$. If the voter happens to choose a different challenge, the DRE can escape detection by rebooting again and then behaving normally. Note that in all these attacks, non-malicious hardware or software failures could cause the same problems. This may make it hard to distinguish purposeful attacks from unintentional failures.

Subliminal channels, also known as covert communication channels, arise in electronic ballots when there are multiple valid representations of a voter’s choices. If the DRE can choose which representation to submit to the bulletin board, then the choice of the representation can serve as a subliminal channel. A subliminal channel in an encrypted ballot carrying the voter’s choices and identifying information about the voter threatens voter privacy and enables vote coercion. For example, a DRE could embed in each encrypted ballot the time when the ballot was cast and who the voter chose for president. Then, a malicious observer present in the polling place could record when each person voted and later correlate that with the data stored in the subliminal channel to recover each person’s vote. Alternatively, if a malicious poll worker learns a voter’s BSN, he can learn how a person voted since each encrypted ballot includes the BSN in plaintext. Detecting such attacks can be quite difficult: without specific knowledge of how to decode the subliminal channel, the encrypted ballots may look completely normal.

5 Obstacles to Adoption

In this section, I explore the obstacles which would be necessary to overcome prior to the implementation of Neff’s scheme to a real, binding, large-scale governmental election. Although the technical aspects of an electronic voting scheme are central to its worth, a scheme will never be adopted unless it is viewed as trustworthy. Not only do the election boards need to trust it, but, arguably more importantly, the electorate itself must believe in a scheme’s trustworthiness.
Neff’s scheme relies on the participants detecting any reordering of the messages in the protocol. Previous studies have shown that non-cryptographers have a limited understanding of cryptography and how to use it[13]. In that voting protocol, humans are not just asked to use cryptography, but to become an active participant in a cryptographic protocol. By internet voting the humans are not assisted by other authorities and minor deviation from the protocol may not be noticed. Participating in an interactive cryptographic protocol is tricky and error-prone, and humans may not notice if the DRE makes subtle deviations from the protocol which dramatically affect security. There was a field study conducted and published by Bederson and Herronsonin 2002 [14] that sought to capture the electorate’s reaction to touch-screen style DRE-machines. The results indicated that the reaction to the Diebold Accu-Vote TS machines during these trials was that “most of the voters responded favorably to it”. The most relevant question asked in the questionnaire was whether the voters “trusted that the system recorded the vote they intended to cast”. Of the responses, 85% reported trust in the system, while 7% reported moderate trust and the remaining 8% indicated they did not trust or only somewhat trusted the system. The results lead us to believe that generally, voters are open to DRE-machine technology. But it is important to consider whether the scheme is too technically complex for voters to trust it. The security of Neff’s scheme relies on the DRE not knowing how the voter will make future decisions, the interactions between the DRE and voter happening in a particular order, and the voter carefully monitoring the DRE’s output. This seems to be the major obstacle for its public acceptance.

The affordability of any scheme is certainly a concern. Luckily, the bulk of the costs of Neff’s scheme lie in the DRE-machines themselves. Therefore in many cases the cost of adopting Neff’s scheme would consist of relatively small add-on costs, including special purpose printers. There is also the cost of designing, creating, and maintaining election web sites. There are multiple other recurring costs that are not seen as additional costs, as they are no different than the costs of other current election schemes. A few examples of these reoccurring costs include those of the personnel, polling stations, and certification of the equipment associated with an election.

6 Conclusion

After careful analysis of numerous aspects surrounding the voting problem, I conclude that it will be difficult for Neff’s scheme, in its original proposed form, to be adopted in elections. The social, legislative, and economic factors
surrounding election decisions limit the scheme’s likelihood of implementation. However, the scheme is innovative and its central ideas are notably promising. It has shifted the focus of poll station e-voting research towards a new direction: that of provable security through cryptography. It presents a new research challenge by placing human voters directly within an interactive cryptographic protocol. Protocol designers have previously assumed participants are infallible computer agents, but voting protocols must cope with human error and ignorance. As with the development of any new technology, refinement of an original idea is necessary in order to improve upon its weaknesses.

Neff’s scheme is of great importance, and its simpler variants seem to be headed toward acceptance for real elections. The scheme already meets a strong definition of security. Consequently, if a variant is able to sufficiently address the non-technical concerns, in particular the issue of complexity, it will likely be considered for adoption in a real, binding, large-scale governmental election. The other vulnerabilities mentioned above are fundamental in the architecture of the Internet and of the PC hardware and software that is ubiquitous today. They cannot all be eliminated for the foreseeable future without some unforeseen radical breakthrough. It is quite possible that they will not be eliminated without a wholesale redesign and replacement of much of the hardware and software security systems that are part of, or connected to, today’s Internet.
Literatur


[15]