Formally Counting Electronic Votes (But Still Only Trusting Paper)

Joseph R. Kiniry
KindSoftware Research Group
Systems Research Group
Complex and Adaptive Systems Laboratory
School of Computer Science and Informatics
University College Dublin
Belfield, Dublin 4, Ireland

E-mail: kiniry@ucd.ie

Abstract

In this extended abstract we summarize our consulting work, scientific research, and activism in the topic of electronic (computer-based) voting. The Dutch and Irish government's activities are our particular focus, as is the Kiezen op Afstand (KOA) system, an experimental platform for electronic voting research with formal methods. We also reflect on the current state of affairs in The Netherlands and Ireland, and discuss next research steps in trustworthy, verified electronic voting systems.

1. Introduction

The Netherlands is known for its forward-thinking and progressive government, laws, and policies and has used computers in voting for over a decade. It is our perception that the current Irish government wants to be a progressive, "modern" government, thus it has recently aggressively attempted to adopt the use of electronic voting machines. Unfortunately, a government's progressiveness, particularly with respect to the adoption of new technology, is sometimes counter to the good of its citizens.

Two research groups have been directly involved in the evaluation and development of voting systems in their respective countries over the past five years. The Security of Systems (SoS) Group at the Radboud University Nijmegen, led by Professor Bart Jacobs, and in which Dr. Joe Kiniry was a key member, has been involved in the evaluation and development of several voting systems in The Netherlands and, recently in the development of a voting system for the Scottish government. The KindSoftware Research Group at University College Dublin, led by Dr. Joe Kiniry, has been involved in the evolution of a Dutch remote voting platform as well as, as a scientist activist, in evaluation, and the fight against the adoption of, voting computers in Ireland.

In this paper, we first summarize our scientific and activist work in computer-based voting. Next, the major technical and sociological flaws of existing system designs, implementations, and protocols are discussed. Finally, we present KOAv2, a new Open Source platform for experimentation in computer-based voting and highlight ongoing work and open problems in trusted, trustworthy electronic voting.

2. Voting Computers in The Netherlands

Electronic voting machines (EVMs), whose primary supplier in The Netherlands is Nedap/Groenendaal, were introduced without controversy in Holland around 1998. They have been widely used in local and national elections ever since. Ninety percent of the votes in The Netherlands are cast on the Nedap/Groenendaal ES3B voting computer [6].

Part of the reason that EVMs were so readily accepted and recently used, for example, for the European general elections in June 2004, is historical. The Netherlands has used digital voting machines (the previous-generation systems with little-to-no software) since the 1980s, and there was very little contrary discussion to such in the media or government for many years. Therefore, Dutch citizens were comfortable with the idea of using technology for voting. This individual and collective comfort-level meant that the

security and reliability issues of the new generation of machines was not raised at the time of their introduction, unlike the furor witnessed during their adoption by other governments in the late 1990s.

But new systems were radically more complex than previous generation machines. And unfortunately, many aspects of these systems were not made public, contrary to the requests of concerned parties in The Netherlands. The internals of such systems are secret and are only (purportedly) exposed to evaluators. Each system (a particular version of hardware and software in combination), under a strict non-disclosure agreement, must be evaluated, according to an unknown set of criteria, before being accepted by the Dutch parliament for use in elections. The main national evaluator is a commercial organization called TNO. Evaluation reports are secret as well.

As attention has focused the world-over on EVMs, particularly beginning around 2002, the Dutch parliament began to reevaluate their use. The SoS Group was involved in some of these evaluations, as discussed in the sequel. The main result of such evaluations was the (very common) suggestion that voter verifiable (paper) trails be used in computer-based voting in The Netherlands.

The Dutch parliament began conducting experiments in 2004 with the "natural" next technological step in the use of technology for voting: remote, Internet-based voting. An usual, the general thought of government officials seemed to be: many complex activities can be accomplished from home these days (e.g., banking, buying airline tickets, etc.), so why not voting? It was generally believed that such a system would increase voter participation because voting is relatively inconvenient for some today. Currently, Dutch citizens must take time off of work to vote because polls are open only during extended business hours (8am to 8pm) for a single day of the work-week, and furthermore each individual must vote in a particular location near their home, which is sometimes far from their workplace.

The question is, given what we know about unreliability and vulnerability of software and networks, do the risks inherent in the introduction of such a system outweigh the benefits?

3. Remote Voting in The Netherlands

European Elections of June 2004 permitted "Remote Voting" or, literally translated, "Kiezen op Afstand" (KOA) via the Internet and telephone. Remote voting was not broadly deployed nationwide. Instead, remote voting was intended only for expatriates, and only after explicit registration. It was thought that such a smaller-scale use (thousands of voters) would provide a useful real-world test for remote voting, with the intention of possibility adopting the technology nationwide in the near future.

Internet-based solution became an option for adoption for a decidedly non-technical reason. The reason, at its core, was that by significantly constraining the remote voting problem, particularly with respect to the registration and voting *process* itself, a secure and reliable system might be constructed. This process, a complex protocol, includes not just the standard elements of protocols in computer science, (computers, networks, databases, cryptography algorithms, etc.), but also people (the voter, government officials, etc.) and organizations (the city hall, political parties, etc.).

The goal in The Netherlands was that such a system must be at least as secure and reliable as the existing remote voting system, which is based upon paper ballots that were physically mailed. But such a justification must be made based upon some kind of risk analysis, coupled with a cost analysis, and an objective balance between risk and cost. Unfortunately, in our experience, such an analysis rarely takes place.

3.1. The Remote Voting Process

The entire KOA voting process is significantly underspecified in public documentation and system specifications. The protocol is summarized here, and each glaring problem with the correctness and/or security of the protocol is highlighted. Our intention is not to disparage this particular protocol, but instead to highlight the fact that designing such protocols, especially when they include human and organizational agents, is extremely difficult and must be carefully specified, verified, and analyzed for risk and cost, prior to any system design and development takes place.

Protocol phase one: Voter registration and authentication. Voter registration is performed using a two-stage process that relies upon *physical* mechanisms for voter registration and authentication.

To register their interest in using remote voting, a citizen must physically visit a designated official government office with official documentation (e.g., a passport) to prove that they are a legitimate voter. At that time the voter chooses a *secret personal access code* (hereafter known as the *voter PIN code*) which is only known to the voter. This voter PIN code is typed into a standard PIN keypad and is stored in a database, associating the voter with his or her voter PIN code.

Theoretically, only the voter knows his voter PIN code. But as the voter PIN code is associated with the voter in some database, and there is no information provided about how that association is stored and protected, this is the first place in the protocol where there is significant underspecification and risk.

The protocol designates that a unique *voter identification code* (hereafter known as the *voter ID code*) is created for

the voter at some point in time, but it is unclear when this step takes place. If it is only done later, when voting sheets are generated and mailed to the voter (see below), then voter privacy is violated.

It is also unclear in the current system design who is responsible for generating and recording voter ID codes. If the voter ID code is generated by the appropriate agency (i.e., the *voting authority* for this particular election) securely when the voter registers, and that voter ID code is used securely in the storage of the voter PIN code, and any association between voter ID code and voter identity is destroyed, then we (perhaps) have the foundation for a secure authentication procedure in this voting process.

Voter ID code:	29835729
Vote date:	24 May 2007
Voting web site:	`http://evote.ie/`
Web site SSL certificate data	
Serial Number:	`00 FC 38 A5 5C ...`
Country	IE
State/Province	Dublin
Organization	Irish Election Authority
Common Name	`evote.ie`
Email Address	`authorityevote.ie`
Not Valid Before	10 January, 2007
Not Valid After	25 May 2007
MD5 Fingerprint	`A4 C7 E9 85 69 ...`

Table 1. Example Election Information Sheet

Some time later, a voter information packet is *physically* mailed to the registered voter. This packet contains general information about the election itself (date, time, etc.), and also customized details that are provided only that specific voter. An simplified example of such voter-customized details included in Table 1. These details include information for voter authentication, including the aforementioned voter ID code, but not his previously chosen voter PIN code.

Candidate	District	Code
Alice Jones	Dublin	123456789
Bob Smith	Dublin	246801234
Charlie O'Connor	Wexford	293548723
Dermot O'Brian	Wexford	395872983

Table 2. Example Candidate List

Also included in this information packet is a list of all candidates running for an office in the current election. On this sheet, a simplified example of which is in Table 2, each candidate is assigned a different number, what is known as a *candidate ID code*. Candidate ID codes are using during voting, rather than candidate names, as discussed later.

The procedure used to generate candidate ID codes is critical in this voting protocol. Candidate ID codes, which are a kind of hash code, are generated by the voting authority. Associated with each candidate is a set of uniquely generated codes. For example, in the aforementioned EU election in 2004, each candidate had ten unique candidate ID codes. Each voter is given only *one* of these codes, and only *that particular code* may be used by the voter during the election. Therefore, election sheets may not be shared between voters, and any observer that sees even a clear-text ballot sent from a client to the voting server gains no information about that ballot's details.

Speculatively, it is possible to derive a perfect hash for a given election, given the voting authority know the number of candidates and the number of registered voters in a given election. Such a unique set of codes guarantees that each and every election information sheet is unique. If only a small, finite number of candidate ID codes are generated then it is possible that two election information sheets are identical.

This is an example of a critical variance point in this protocol. What is the cost of generating more candidate codes? Obviously, using only a single candidate ID code per candidate provides very little security. Intuitively, increasing the number of candidate ID codes seems to increase the security of the voting stage of the protocol, but how is this proposition verified? And furthermore, how many codes is enough?

KOA is an excellent example of a scenario we see far too often in the computing industry, and particularly in electronic voting systems: that of ad hoc systems design. Electronic voting is widely acknowledged as being a critical software system. In classical critical systems design and development industries like aerospace, transportation, banking, etc., systems are described at varying levels of detail using precise, formal notations. Likewise, protocols are carefully defined and verified using specialty tools like model checkers.

In the KOA system, on the other hand, the voting protocol design seems very ad hoc—there is no known explicit security and privacy requirements specification, there is no risk analysis for the protocol, and critical design choices, like using ten candidate ID codes for an EU election, are seemingly pulled out of the air. There is no justification for the design choices made, and there is no analysis that the protocol used is correct or secure. Please refer to Figure 1 while reading the following summary of the KOA voting protocol.

To vote, a registered voter connects to the web server

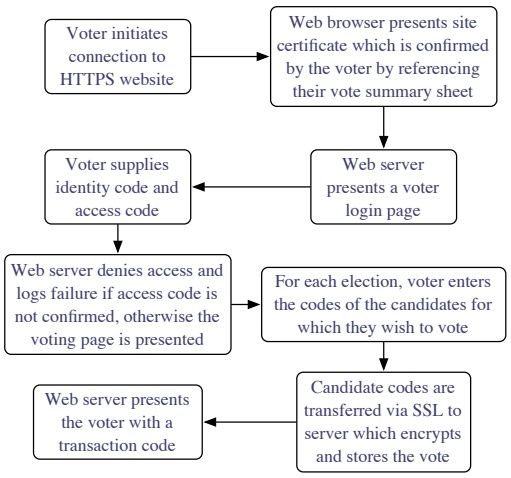

Figure 1. The KOA Vote Process

designated by a URL printed on the election information sheet (recall Table 1). Communication with the voting web site is secured with SSL. To ensure that the voter is communicating with the correct server, the metadata associated with the SSL certificate for the voting site (e.g., information about the site owner, the voting authority, the certificate issuer, certificate serial number, dates of validity, key fingerprints, etc.) is printed on the election information sheet as well. Thus, when the voter initially connects to the voting website, the voter's web browser challenges him to visually inspect the site's certificate, checking that its data conforms to that which is printed on the election information sheet.

After confirming that they have connected to the correct website, the voter authenticates with the web site with their voter ID code and voter PIN code. They then step through a series of simple web pages, typing in candidate codes as appropriate for their candidate choices. The optionally system responds by showing the voter the actual names and parties of the candidates in question to confirm the accuracy of their choice. This highlights another serious flaw in the current system: the data sent *to* the server is obfuscated by virtue of the fact that candidate ID codes are used, but any confirmation of a voter's choices is sent (in plain-text or not) over the SSL channel, thereby potentially leaking a voters ballot to a determined hacker (via a network sniff, man-in-the-middle attack, or otherwise).

When a voter is done, a *election transaction code* is provided. The election transaction code is posted to a public web-based bulletin board system so that the voter can later check that the voter's choices were included properly in the final tally for the election. Unfortunately, given the current system design and because a verifiable voting scheme was not used, simply because a transaction code is printed to the website does not guarantee that particular ballot was counted.

All votes are, in the end, stored in a doubly-encrypted fashion—each vote is encrypted by a symmetric key, per voter, as well as the public key of the voting authority. Only the appropriate voting officials have the corresponding private key and know its passphrase, and thus only they can initiate the ballot counting process. This highlights another ad hoc flaw in the system: cryptography experts that have reviewed KOA find no purpose for the symmetric per-voter key.

In the Dutch EU election in 2004 the passphrase for this key was meant to be known only to the sole individual responsible for the election. In fact, there was some fanfare about it being held in a safe during the election. When the time came though to count the ballots and the passphrase was obtained from the safe, it was discovered that it was the wrong passphrase—no one had activated the election public key in use on the server and the test key that had been used for months had been used in the real election!

Again, this kind of failure highlights the fact that certifying an election procedure is about more than just ensuring that the software system is of high quality or a computer server (or even a key passphrase!) is in a locked room. The entire voting *process* must be well-documented, well-understood, and perfectly executed as complex protocol composed of hardware, software, and people-ware.

Additionally, since failures in various components and their interactions are bound to happen, understanding the implications of such are critical. When is a virtual ballot spoiled? When exactly is a recount necessary? When must the election be invalidated?

3.2. Evaluation of the KOA System

As highlighted, the KOA system, as it stood in 2003, was not rigorously designed, engineered, tested, or deployed. The SoS Group participated in two evaluations of this system, not only to help improve its quality, but also to help the evaluators and activists better understand its properties so as to argue *against* its use.

A lightweight expert panel review. First, Prof. Jacobs was part of an external expert panel that reviewed the system's high-level requirements and design, but performed no review of the software. That panel wrote a report critical of many aspects of the system and, as a result, many of the panel's recommendations were incorporated in the later versions of the system.

Three of the key recommendations were: 1) The system must not be designed, implemented, tested, and administered by the same company. 2) The source code of the system must be made available to any parties interested in

reviewing the code. 3) The system must anonymously log all raw data in the voting process so that an independent recount is possible by any interested third party.

After this (extremely lightweight and dissatisfatory, at least to the SoS Group members) evaluation took place, the group did not expect to be further engaged with the process, even though we were very interested in learning more and challenging the system's introduction.

A secret white hat network security review. Our next involvement in the system came about because Prof. Jacobs was notified that a secret/private demonstration of the KOA system was to take place for the benefit of interested parties in government. Example election information sheets were produced and the KOA system was configured to run a mock election. The SoS Group obtained one of these sheets as well as the name of the machine that was running the example voting server.

During the week of the test our group, led by the author, conducted a gentle network analysis of the ISP hosting the server, the subnet on which the server resided, and all adjacent subnets. Additionally, we analyzed all systems on these subnets, determining which operating system (type and version) they were running, which network services they had enabled, and their likely purpose.

On the final day of the test we attacked the site, executing a small distributed denial-of-service attack. This brought the site to a crawl and (obviously, purposefully) alerted the authorities to our activities. We were told that the Dutch Secret Service was called and various parties were quite agitated until they determined the attack was coming from Nijmegen, thus likely from white hats in Prof. Jacobs's group.

Afterwards, the team responsible for the KOA system were quite curious to hear our analysis. We made dozens of recommendations in a report to the Ministry and nearly all of these recommendations were adopted. In particular, we highlighted the fact that test and development machines must not be on the same subnet as the deployed server, all servers must have only the voting service running and no other (e.g., no mail server or secure shell server should be running on the webserver during the election), and all systems must have the most up-to-date, and preferably the most secure, operating systems installed.

A public white hat network security review. Next, the Ministry requested that the SoS Group audit the remote voting system and attempt to break into, or otherwise disrupt, a public trial vote that took place in late 2003. For a one week period, much like we did previously, the group analyzed the remote voting system's network, servers, communication, web site, etc., just as a black-hat hacker would do during the actual vote.

This analysis and experience helped the system designers and administrators learn how to react to a real-world attack [16]. It also pointed out subtle network and physical attacks possible by determined hackers. In particular, the SoS Group was, perhaps unsurprisingly, able to effect a denial-of-service attack on the service with little effort. As a result, the remote voting system was "tightened down" even further and protocols, both technical and procedural, were put in place to deal with denial-of-service attacks.

We must emphasize that, at this point in time, the source code of the system and any of its detailed specifications tests had still not been made available, even to the SoS Group under an NDA.

Formally counting votes. The SoS Group also has become directly involved in the implementation of a small but critical part of the remote voting system. Due to recommendation 1) above, the Ministry decided to open up a bid for a third party construction of the ballot counting subsystem. It is thought that if one or more third parties design and implement the tally software in isolation, then the likelihood of fraud in counting ballots is vanishingly small. The SoS Group bid on this project and won it by virtue of our radical proposal: the ballot counting system would be formally verified.

In early 2004, three members of the SoS Group (Dr. Engelbert Hubbers, Dr. Martijn Oostdijk, and the author) designed, implemented, and verified a ballot counting subsystem. Leveraging our past experience with verification of Java, the vote counting system was constructed in Java and specified with the Java Modeling Language (JML) [17, 18]. The system was extensively tested, as thousands of unit tests were generated with a tool called JMLunit [3]. Additionally, key components were verified using ESC/Java2, a static verification tool co-developed by the SoS Group [1, 8]. As a result, we have very high confidence in the correctness and accuracy of the vote counting system [11, 15].

4. Voting Computers in Ireland

Several years ago the Irish government purchased hundreds of Nedap voting computers for tens of millions Euro. These machines needed some customization to work with the Irish voting system, because, while the Dutch voting system is list-based, the Irish system is a proportional representation, single-transferable vote (PR-STV) system.

Additionally, software needed to be developed to perform election management, including election setup (identifying parties, candidates, districts, etc.), configure each Nedap machine for an election, and perform an analysis of election results, including the final ballot tally and election report. This software was developed by a small, partly Irish

company called Powervote, founded exclusively for this development.

Over the past three years, multiple reviews of the ES3B machines, the Powervote software, and a risk analysis were conducted by the Commission on Electronic Voting (CEV), producing several reports [25, 26].

In short, the end results of these extensive analyses is that the CEV: 1) recommended the use of the Nedap hardware for future use in Ireland, subject to a number of recommendations, foremost of which is the introduction of a voter-verifiable (paper) audit trail [VV(P)AT], and 2) could not recommend the use of the election management software due to serious problems with its quality and correctness.

Unfortunately, the Powervote software has never been made public, nor has any of the Nedap hardware or software. Recently, activists analyzed and hacked the Nedap machines in question, raising additional questions not addressed in detail in the CEV reports [9].

It is important to note that, even were the full voting system's source code released to the public, as has been done in The Netherlands, it is not only the *election software* that must be verified. It is **much** more important to verify the *election procedure* and run a *verifiable* election.

Additionally, the citizens participating in an election must have *trust* in the election procedure, which includes a trust in all of the subcomponents and subsystems of an election: the election officials, observers, electronic and non-electronic equipment, security (e.g., police in Ireland, independent security personnel in The Netherlands), media, etc.

In other words, *Open Source Software* is not enough: an *Open Voting Process* is mandatory.

5. Where Ireland and The Netherlands Meet

What brings Ireland and The Netherlands together is more than just their their choice of voting computers, as both are heavily invested in Nedap hardware. Additionally, for example, they are both relatively small countries with respect to geography and population. Thus, one would hope that the two governments learn from each others' experiences, and perhaps even work together to better understand and solve the challenges of elections, with or without the use of electronic voting equipment.

But there are also a number of critical differences between the two countries and their citizens, differences that significantly impact on the adoption of electronic voting. Furthermore, these differences make it unlikely that they will work together to solve their joint problems in electronic voting.

For example, citizens' trust in government varies dramatically. Dutch citizens seem to trust their government and its officials significantly more than Irish citizens trust their officials. Thus, a risk analysis in The Netherlands will be biased by this perception difference. Likewise, a formal analysis of the election process which incorporates human agents (particularly election officials and voters) must take into account this difference.

Second, The Netherlands is very technically modern society. Electronic cash is widely used, one can pay for buses and trains with "Chip and PIN" smart-cards, all government services and banks are available online, the public transportation system is very efficient and reliable, etc. Ireland is quite to the contrary: credit cards are not even widely used, let alone any kind of electronic cash-based system, one may only pay for local public transport with cash, very few government services are available online, etc. Thus, in some sense, The Netherlands is more able to rationally adopt technological solutions to the public's problems, elections or not.

Nedap's corporate behavior has, thus far, also varied greatly between the two countries. To date, Nedap officials have been quite conciliatory to Irish officials (see the responses from Nedap and Powervote in the Appendices of the CEV reports), whereas they have recently been very threatening to Dutch officials (see the Dutch electronic voting activists website [6]).

Additionally, the two governments differ dramatically in their openness. For example, only through the hard work and great cost of Irish citizen activists, primarily through Freedom of Information requests, has any information at all about electronic voting in Ireland become public [10].

Even though the Dutch government is more open and seemingly more rational than the Irish government, poor decisions are still made. For example, recently, on the demand of the Dutch government, a VVPAT solution was added to Nedap machines and demonstrated in The Netherlands. Unfortunately, the precise solution that they have chosen to implement, a scroll-based ballot printout being a glass window, is widely regarded as perhaps *the* worst realization of VVPAT.

So the situation is: 1) nearly all electronic voting systems adopted by governments are proprietary, 2) most computer scientists and mathematicians are not aware of the multitude of interesting problems (algorithmic, computational, mathematical, etc.) inherent in electronic voting systems, and 3) we need a simple, concrete way to educate scientists, government officials, and normal citizens about the problems, challenges, and good solutions in electronic voting.

Thus, in response to these needs, our research group has adopted and extended the (incomplete, ill-documented, non-working) KOA electronic voting system for the purpose of providing a concrete, high-quality technical foundation for, quite broadly, international collaborative research in electronic voting systems.

6. A Platform for Electronic Voting Research

The KOA system is described in detail in several papers and theses [4, 7, 14, 20, 27], thus we will not attempt to describe the system in full detail here. The system, including all system description and requirements [19], source code, formal specifications, open problem descriptions, etc. is available via our research group's website [13]. This platform is meant to provide a foundation for tackling many of the problems elucidated in critical studies like the SERVE report [12].

As mentioned previously, the KOA system was used for a remote voting experiment, voting over the telephone and Internet, in 2004. But the software architecture is not specific to *remote* voting; it is a perfectly acceptable foundation for a kiosk-based or polling place-based electronic voting system.

The KOA system released by the Dutch government was seriously incomplete. It was a partial snapshot of (only) the source code of the system—the system did not compile, approximately 10% of the system was missing due to intellectual property claims, it was tied to a proprietary foundation (a commercial Enterprise Java technology), and it contained no high-level documentation and very few tests.

Our group, working with others, have resolved these issues. We have reverse engineered the missing pieces, rewritten the system to work on an entirely Open Source foundation (primarily Enterprise Java and database technologies), written new subsystems (e.g., a formally specified and verified implementation of the Irish PR-STV voting system), and translated to English much of the documentation we eventually received from the Dutch government.

The main subsystems missing at this point in time are:

- *VVPAT subsystem*

  We have not built a subsystem for experimenting with different forms and processes of voter-verifiable (paper) audit trails. Of particular interest are concrete, scientific HCI experiments with different VVPAT solutions.

- *non-web-based voting interface*

  The primary interface to the system is via a web browser, which is a very simple, but perhaps to too heavyweight and overly complex front-end. Election setup, voting, and election reporting are conducted via a web browser interacting with the server software, but the formally specified vote counting system is implemented with a Java Swing-based interface.

- *formal specification of the full voting process*

  Only specific subsystems, particularly the Dutch and Irish tally subsystems, are formally specified at this time. A variety of other subsystems must be specified to completely understand and reason about the full system. It is not even clear how to couch such specifications, which formal foundation to use, or which aspects of the problem to focus upon first.

- *analysis of the remote voting protocol*

  The protocol used in the Dutch elections, as realized in the current system, has not been formally specified or critically analyzed in any fashion. The creation of new distributed systems protocols, particularly with the use of cryptography, is incredibly difficult. New protocols must have formal specifications be formally verified before they are implemented.

- *risk and cost analysis of the system*

  No risk or cost analysis, or even a proposed model or framework for such, exists for the KOA system. There are many excellent collaboration opportunities for this kind of work, perhaps between a systems- or verification-centric computer scientist, a cryptographer, and an economist.

- *trustworthy voting system*

  The currently specified and verified voting systems realized in KOA are not *trustworthy* voting systems. Incorporating such a system, like those of Chaum, Ryan, and Schneider (e.g., *Prêt à Voter*) [2, 21, 22, 23, 24], especially if it is formally specified and verified, is an excellent research opportunity.

As is clear, there are many open scientific and engineering opportunities in electronic voting that may be concretely realized and critically analyzed in the KOA system.

7. . . . But Still Only Trusting Paper

Regardless of the existence of this high-quality experimental foundation for electronic voting research, there is still far too many serious open problems to identify and solve before electronic voting is adopted for any serious local or national election. As mentioned before, *Open Source Software* is not enough, an *Open Voting Process* is mandatory.

In the end, physical (perhaps paper) ballots **are** the ballots. Digital ballots are only a "shadow" of the physical ballot. It is extremely difficult to ensure that digital artifacts are not copied, manipulated, and destroyed—exactly the kinds of manipulations we do not wish ballots to permit. Even advanced so-called "Digital Rights Management" (DRM) systems that the entertainment industries have promulgated are no match for a determined adversary interested in stealing a song, let alone a determined group of political activists interested in stealing an election.

Only after completing the open tasks summarized in the previous section, and experimenting with a variety of forms of physical ballots and verifiable voting systems, can we begin to make a rational decision about the introduction of electronic voting in a major election.

8. Conclusions

In our experience, Dutch government representatives responsible for electronic voting systems, remote or not, have a "nothing to hide" attitude and are open to new ideas and external review. They also have a healthy distance toward the supplier of the system; even after spending significant funds on building and testing the system, because SoS Group and others found non-trivial flaws in the system's design and implementation, the project was immediately suspended.

The situation is Ireland is dramatically different. Even after the detailed involvement over several years of dozens of experts, generating hundreds of pages of high-quality reports and reflections on the system, the current government officials, particularly the Taoiseach (Prime Minister), Mr. Bertie Ahern, and the Minister currently responsible for the electronic voting hardware, Mr. Martin Cullen, still claim that the system is fit for use and Ireland's lack of adoption of electronic voting machines is, in fact, the opposition's fault.

In fact, only in the first week of May 2007, four years after machines were purchased, did the Minister finally admit that the government might have made a mistake in purchasing equipment before consulting with experts:

> I suppose with hindsight, yes, I might have dealt with it differently ...
>
> I suppose if I was to go back, I would have said let's do another testing phase in the next election rather than moving it into full use immediately, but there isn't a person yet who hasn't made a mistake on things, and if that's a mistake, I put my hands up.
>
> -Mr. Martin Cullen, on the Irish government's adoption process for electronic voting

Unfortunately, the Taoiseach still refuses to acknowledge the truth of the current situation. As reported in late April, 2007 in the national media [5]:

> The Taoiseach has told the Dáil he was embarrassed by the fact the we did not have an electronic voting system when he watched the French election results come in within two hours.
>
> He said he apologised to the people of Meath where he had been visiting at the time, and said in a technologically advanced country we were going back to the 'peann luaidhe'.
>
> Mr Ahern said that with 'a bit of luck' our election would be finished within five days.
>
> The Taoiseach blamed the Opposition for the electronic voting machines lying unused and said if they had not played politics with the issue we would not be the laughing stock of Europe.
>
> He said if they had taken a mature attitude to a voting system which had worked, we would have electronic voting and it was a disgrace.
>
> Labour Party leader, Pat Rabbitte criticised the 62m Euro overspend on electronic voting machines. He said this Government had presided over huge waste and the Comptroller and Auditor General had estimated that the cost of storing these machines was about 1m Euro a year.

Regardless of these statements, the Irish government is not using computers in the upcoming election on the 24th of May. But this situation may change, regardless the continued effort of activists, scientists, and activist scientists. Governments, like any other large group, do not necessarily exhibit rational behavior.

It is in this atmosphere that our work on the KOA system continues. This system represents a high-quality foundation for performing experiments in electronic voting. It is hoped that, by elucidating the complex conflicting challenges inherent in voting, more computer scientists and mathematicians will become involved in electronic voting research.

But, while we have a high degree of confidence in this experimental system, particularly with regards to our portions written and verified by our groups with formal methods, there are still open issues that must be addressed prior to the adoption of electronic voting in any serious venue.

Foremost among these issues is our contention that all of the software used in an electronic voting system must be made Open Source. KOA represents a nearly unique situation because, unlike many similar electronic and remote voting systems developed for governments, the Dutch government owned the program code of the KOA system. Thus, there were no corporate claims of ownership or secrecy on the program code. It was only due to this ownership situation, coupled with the Dutch government's honest desire for openness, that nearly the entire KOA system was released in July of 2004 under the GPLv2 Open Source license.

Surprisingly, until very recently, mainstream press has only barely picked up on this major event and further developments on KOA. The primary Dutch website dedicated to Open Source software has covered the story well. To learn more about the original release of the KOA system, see the Ministry's own web pages and *OSOSS.nl*'s continued cov-

erage. To learn more about the new KOAv2 release, see our research group's website [13].

References

[1] P. Chalin, J. R. Kiniry, G. T. Leavens, and E. Poll. Beyond assertions: Advanced specification and verification with JML and ESC/Java2. In *Proceedings of Formal Methods for Components and Objects (FMCO) 2005*, Lecture Notes in Computer Science, 2006.

[2] D. Chaum, P. Ryan, and S. A. Schneider. A practical, voter-verifiable election scheme. Technical Report CS-TR: 880, School of Computing Science, Newcastle University, Dec. 2004.

[3] Y. Cheon and G. T. Leavens. A simple and practical approach to unit testing: The JML and JUnit way. In B. Magnusson, editor, *ECOOP 2002*, volume 2374 of *Lecture Notes in Computer Science*, pages 231–255. Springer–Verlag, June 2002.

[4] D. Cochran. Secure internet voting in Ireland using the Open Source Kiezen op Afstand (KOA) remote voting system. Master's thesis, University College Dublin, Mar. 2006.

[5] Lack of e-voting a disgrace, says Ahern. `http://www.rte.ie/news/2007/0425/voting.html` accessed in May, 2007, Apr. 2007.

[6] We don't trust voting computers. `http://wijvertrouwenstemcomputersniet.nl/Nedap-en`, 2006–.

[7] F. Fairmichael. Full verification of the KOA tally system, May 2005.

[8] C. Flanagan, K. Leino, M. Lillibridge, G. Nelson, J. B. Saxe, and R. Stata. Extended static checking for Java. In *ACM SIGPLAN 2002 Conference on Programming Language Design and Implementation (PLDI'2002)*, pages 234–245, 2002.

[9] R. Gonggrijp, W.-J. Hengeveld, A. Bogk, D. Engling, H. Mehnert, F. Rieger, P. Scheffers, and B. Wels. Nedap/Groenendaal ES3B voting computer: a security analysis. Available via the "We don't trust voting computers" website at `http://wijvertrouwenstemcomputersniet.nl/Nedap-en`, Oct. 2006.

[10] Irish citizens for trustworthy evoting. `http://evoting.cs.may.ie/`.

[11] B. Jacobs. Counting votes with formal methods. In C.Rattray, S. Majaraj, and C. Shankland, editors, *Algebraic Methodology and Software Technology*, volume 3116 of *Lecture Notes in Computer Science*, pages 241–257. Springer–Verlag, 2004.

[12] D. Jefferson, A. D. Rubin, B. Simons, and D. Wagner. A security analysis of the secure electronic registration and voting experiment (SERVE). `http://www.serveusa.gov/`, Jan. 2004.

[13] KindSoftware research group homepage. `http://secure.ucd.ie/`, 2004–.

[14] J. Kiniry, A. Morkan, D. Cochran, F. Fairmichael, P. Chalin, M. Oostdijk, and E. Hubbers. The KOA remote voting system: A summary of work to date. In *Proceedings of Trustworthy Global Computing*, 2006.

[15] J. R. Kiniry and D. R. Cok. ESC/Java2: Uniting ESC/Java and JML: Progress and issues in building and using ESC/Java2 and a report on a case study involving the use of ESC/Java2 to verify portions of an Internet voting tally system. In *Construction and Analysis of Safe, Secure and Interoperable Smart Devices: International Workshop, CASSIS 2004*, volume 3362 of *Lecture Notes in Computer Science*. Springer–Verlag, Jan. 2005.

[16] J. R. Kiniry, M. Oostdijk, and E. Hubbers. Beveiligingsanalyse van `Internetstembureau.nl`. Presented to the Dutch parliament in 2004., 2004.

[17] G. T. Leavens, A. L. Baker, and C. Ruby. *Behavioral Specifications of Business and Systems*, chapter JML: A Notation for Detailed Design, pages 175–188. Kluwer Academic Publishing, 1999.

[18] G. T. Leavens, A. L. Baker, and C. Ruby. Preliminary design of JML: A behavioral interface specification language for Java. Technical Report 98-06i, Department of Computer Science, Iowa State University, Feb. 2000. Available via `http://www.cs.iastate.edu/~leavens/JML.html`.

[19] LogicaCMG. Kiezen op Afstand: Hertellen Stemmen. Functional specifications, 2004.

[20] A. Morkan. KOA evaluation, demonstration installation and implementation, May 2005.

[21] B. Randell and P. Ryan. Voting technologies and trust. *IEEE Security & Privacy*, 4(5):50–56, 2006.

[22] P. Y. A. Ryan. Prêt à with paillier encryption. Technical Report CS-TR: 1014, School of Computing Science, Newcastle University, Apr. 2007.

[23] P. Y. A. Ryan and T. Peacock. Prêt à: A system perspective. Technical Report CS-TR: 929, School of Computing Science, Newcastle University, Sept. 2005.

[24] P. Y. A. Ryan and S. A. Schneider. Prêt à with re-encryption mixes. Technical Report CS-TR: 956, School of Computing Science, Newcastle University, Apr. 2006.

[25] The Commission on Electronic Voting. First report of the commission on electronic voting on the secrecy, accuracy and testing of the chosen electronic voting system. Technical report, The Commission on Electronic Voting, Dec. 2004.

[26] The Commission on Electronic Voting. Second report of the commission on electronic voting on the secrecy, accuracy and testing of the chosen electronic voting system. Technical report, The Commission on Electronic Voting, July 2006.

[27] P. Tierney. Implementing the irish voting system, May 2007. Final Year Undergraduate Project Thesis.