Sensus: A Security-Conscious Electronic Polling System for the Internet

Lorrie Faith Cranor*
Public Policy Research
AT&T Labs Research
lorrie@research.att.com

Ron K. Cytron
Department of Computer Science
Washington University in St. Louis
cytron@cs.wustl.edu

Abstract

We present the design and implementation of Sensus, a practical, secure and private system for polling (conducting surveys and elections) over computer networks. Expanding on the work of Fujioka, Okamoto, and Ohta, Sensus uses blind signatures to ensure that only registered voters can vote and that each registered voter only votes once, while at the same time maintaining voters' privacy. Sensus allows voters to verify independently that their votes were counted correctly, and anonymously challenge the results should their votes be miscounted. We outline seven desirable properties of voting systems and show that Sensus satisfies these properties well, in some cases better than traditional voting systems.

1. Introduction

Democratic governments and organizations must have mechanisms for polling their members. Traditionally, elections have served as the official mechanisms for people to express their views to their governments, while surveys have augmented elections as unofficial — but nonetheless valuable — measures of public opinion. In both surveys and elections, privacy and security are usually desired, but not always simultaneously achievable at a reasonable cost. Mechanisms that ensure the security and privacy of an election can be time-consuming and expensive for election administrators, and inconvenient for voters. Conducting secure and private elections can become even more difficult when voters are geographically distributed.

Due to the rapid growth of computer networks and advances in cryptographic techniques, electronic polling is now a viable alternative for many non-governmental elections and surveys, and it is likely to become viable soon for governmental elections as well. Electronic polling over the Internet can be convenient for voters with easy access to networked computers, even if the voters are geographically distributed. In addition, electronic surveys and elections can be inexpensive to administer. However, if not carefully designed, electronic polling systems can be easily compromised, thus corrupting results or violating voters' privacy.

Following the work of Fujioka, Okamoto, and Ohta [10], we have designed a security-conscious electronic polling system called *Sensus* that can be used to conduct surveys and elections over the Internet. Sensus was designed primarily as a replacement for postal mail balloting systems; however, it is flexible enough to suit a variety of other polling applications, including those not feasible using traditional polling systems [8]. We have demonstrated that our implementation can be used to conduct small-scale elections. Furthermore, we believe our implementation could accommodate large-scale elections with minor modifications.

While there has been much theoretical discussion of secure and private electronic voting systems, most of the systems actually used for Internet polling ignore security and/or privacy issues. For example, the Reform Party gave its members the option of voting electronically via the World Wide Web (rather than voting by telephone or postal mail) to select the party's 1996 Presidential candidate. However, regardless of the voting method used, voters' names were affixed to their ballots, eliminating any possibility of a secret ballot. Other Internet polls lack provisions to ensure that each voter votes only once. With polls becoming increasingly common on the Web, we believe it is important for people to be aware that security and privacy considerations need not be ignored. If these considerations are not addressed early, lay people are likely to view all secure Internet applications with skepticism. Our work has focussed on developing a *practical* security-conscious electronic polling system design that can be implemented and used for actual surveys and elections.

In this paper we present the Sensus design and implementation. In Section 2 we present our design goals, including a list of desirable properties for election systems. In Section 3 we detail the Sensus polling protocol, describing the role of each system component and comparing Sensus with other polling protocols. In Section 4 we evaluate Sensus and analyze the degree to which it satisfies the properties outlined in Section 2, and in Section 5 we present our conclusions.

* Work was performed while this author was affiliated with the Department of Engineering and Policy, Washington University in St. Louis.

2. Sensus Design Goals

While a wide variety of voting systems and protocols exist, the basic procedure for conducting a democratic election[1] is fairly standard. This procedure generally involves four tasks:

Registration The registration task involves compiling a list of people eligible to vote.

Validation The validation task involves checking the credentials of those attempting to vote and only allowing those who are eligible and who have not already voted to proceed.

Collection The collection task involves collecting the voted ballots.

Tallying The tallying task involves counting the votes.

To have confidence in the election results, people must believe that these tasks are performed properly. However, there are numerous opportunities for corruption during the performance of each of these tasks. For example:

- Election authorities may cheat by knowingly allowing ineligible voters to register, registered voters to cast more than one vote, or ballots to be systematically miscounted or destroyed.
- Ineligible voters may register (often under the name of someone who is deceased) or eligible voters may register under multiple names.
- Registered voters (eligible and otherwise) may be impersonated at the polls.
- Ballot boxes, ballots, and vote counting machines may be compromised.

Traditionally, election fraud has been prevented through the use of physical security measures, audit trails, and observers representative of all parties involved. But the prevention of election fraud is made more difficult by the frequent requirement that votes remain private. Observers may not observe a ballot until after it has been placed in a ballot box, and audit trails must not provide the ability to link a ballot back to the voter who cast it. Even so, these security measures generally work well enough that the possibility of widespread fraud is small and people have confidence that election results are accurate.

[1] In the Sensus system, there is no difference between an election and a survey. Therefore, throughout this paper the terms *poll*, *election*, and *survey* will be used interchangeably. Likewise, the term *ballot* will refer to both survey forms and ballots and the term *voter* will refer to both voters and survey respondents.

When designing an electronic polling system, it is essential to consider ways in which the four tasks mentioned above can be performed electronically without sacrificing voter privacy or introducing opportunities for fraud. In addition, it is useful to consider all desirable polling system properties, including those not always achievable in traditional systems.

Our design goals are based on our survey of the literature on traditional and proposed electronic polling systems. We reviewed several sets of "ideal" election system characteristics found in the literature [1, 10, 15, 18, 20] and developed a set of four "core properties" that are likely to be desirable in almost any election system:

Accuracy A system is accurate if (1) it is not possible for a vote to be altered, (2) it is not possible for a validated vote to be eliminated from the final tally, and (3) it is not possible for an invalid vote to be counted in the final tally.

In the most accurate systems the final vote tally must be perfect, either because no inaccuracies can be introduced or because all inaccuracies introduced can be detected and corrected. Partially accurate systems can detect but not necessarily correct inaccuracies. Accuracy can be measured in terms of the margin of error, the probability of error, or the number of points at which error can be introduced.

Invulnerability A system is invulnerable if (1) it permits only eligible voters to vote, and (2) it ensures that each eligible voter can vote only once.

Privacy A system is private if (1) neither election authorities nor anyone else can link any ballot to the voter who cast it, and (2) no voter can prove that he or she voted in a particular way.

The second privacy factor is important for the prevention of vote buying and extortion. Voters can only sell their votes if they are able to prove to the buyer that they actually voted according to the buyer's wishes. Likewise, those who use extortion[2] to force voters to vote in a particular way cannot succeed unless they can demand that voters prove that they voted as requested.

Verifiability Verifiability. A system is verifiable if anyone can independently verify that all votes have been counted correctly.

[2] Benaloh and Tuinstra [1] explain:

> There are reports that in some small Italian villages, the voting system employed allows voters to list their votes in any order. Political bosses are said to assign different permutations of their preferred candidates to each voter. If a particular permutation fails to appear when the votes are counted, a boss can assume that the voter to which that permutation was assigned did not vote "properly", and reprisals can be taken.

A weaker definition of verifiability used by some authors [10, 18] allows that a system is verifiable if it allows voters to verify their own votes and correct any mistakes they might find without sacrificing privacy. Less verifiable systems might allow mistakes to be pointed out – but not corrected – or might allow verification of the process by party representatives but not by individual voters. Traditional voting systems generally only allow for minimal verification by party representatives.

In addition, we developed three extra properties that an electronic polling system should possess. Two of these properties are important for ensuring a high voter turnout, something that is often desired but not always achieved.

Convenience A system is convenient if it allows voters to cast their votes quickly, in one session, and with minimal equipment or special skills.

Flexibility A system is flexible if it allows a variety of ballot question formats including open ended questions (this is important for write-in candidates and some survey questions).

Mobility A system is mobile if there are no restrictions (other than logistical ones) on the location from which a voter can cast a vote.

One of the reasons people are interested in electronic voting systems is that they can be mobile. Voter participation might increase if people could easily cast votes from computers in their homes, offices, schools, and libraries. Of course, for governmental elections it would be essential to retain centralized polling places for people who would not otherwise have access to computers.

The mobility property itself is a major contributor to some of the problems associated with designing a secure and private electronic voting system. By allowing voters to cast their votes from virtually anywhere, we dramatically expand the universe of ineligible people who may attempt to vote. We also limit our abilities to prevent voters from proving how they voted, as there are no longer private voting booths that can prevent vote buyers from observing vote sellers as they cast their votes.

We designed Sensus to possess all of the above properties (to varying degrees), with one exception. Sensus does not address the second part of the privacy property. Unless voters are required to cast their votes from inside a solitary voting booth, voters will be able to prove how they voted by allowing another party to observe them while they are casting their votes. We do not believe this problem can be addressed without sacrificing mobility or convenience.

In addition, like most distributed cryptographic systems, Sensus does not address problems related to ballots being intercepted or delayed while in transit. The design of the Sensus system assumes that voters have a reliable mechanism for delivering messages to the election authorities in a timely manner.

3. Sensus Polling Protocol

Sensus has been designed as an easily adaptable modular system. The polling protocol requires the existence of validator, tallier, and pollster modules. Additional modules may augment the Sensus system.

The responsibilities of the essential modules include all of the tasks described in Section 2, with the exception of the registration task (which may be performed by human officials or by an optional registrar module). The validator is responsible for the validation task, and the tallier is responsible for the tallying and collection tasks. The pollster acts as a voter's agent, performing all cryptographic and data transfer functions on a voter's behalf.

The Sensus protocol is based closely on a scheme proposed by Fujioka, Okamoto, and Ohta [10] that uses blind signatures[3] to provide security while protecting voters' privacy. The Sensus protocol requires the voter to prepare a voted ballot, encrypt it with a secret key, and blind it. The voter then signs the ballot and sends it to the validator. The validator verifies that the signature belongs to a registered voter who has not yet voted. If the ballot is valid, the validator signs the ballot and returns it to the voter. The voter removes the blinding encryption layer, revealing an encrypted ballot signed by the validator. The voter then sends the resultant signed encrypted ballot to the tallier. The tallier checks the signature on the encrypted ballot. If the ballot is valid, the tallier places it on a list of valid ballots to be published after all voters vote. The tallier then signs the encrypted ballot and returns it to the voter as a receipt. Upon receiving the receipt, the voter sends the tallier the ballot decryption key. The tallier uses the key to decrypt the ballot and add the vote to the tally.

3.1. Sensus Modules

As mentioned above, the validator, tallier, and pollster modules are essential for conducting a Sensus election. In addition, the registrar, ballot-authoring, and other such modules can automate election tasks, saving time or reducing the

[3] The Sensus polling protocol uses *blind signatures* to preserve privacy and democracy simultaneously. First introduced by Chaum [4], blind signatures allow a document to be signed without revealing its contents. The effect is analogous 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.

chance of human error. In this section we will detail the use of these modules to implement the Sensus protocol and describe our module implementations briefly. A diagram illustrating the transactions that must take place between the pollster, validator, and tallier modules is shown in Figure 1.

We have implemented basic registrar, pollster, validator, and tallier modules in C and Perl on a Unix system. Our implementation uses the RSAREF encryption library, which is distributed free of charge by RSA Data Security, Inc. It also uses the Webget Perl script by Jeffrey Friedl. Our implementation requires the registrar, validator, and tallier modules to be run on a machine with a Web server that supports CGI scripts.

For maximum security and privacy, the validator and tallier modules should be run on separate machines and the pollster module should not be run on a machine that houses any of the other modules.

Registrar. The registrar is responsible for registering voters prior to an election or poll. The registrar must take a list of people eligible to register (population list) and a list of people who have applied to register and whose identities have been verified (verified list), and produce a list of registered voters. Registered voters will generally be listed by their names or identification codes, a public encryption key, and optionally, an e-mail address.

As with traditional voting systems, the main difficulty in implementing a registrar lies in verifying the identity of applicants, a task that may be impossible without a face-to-face meeting. For this reason, some election administrators may choose not to automate the registration process. However, for most purposes, an automated registration process can produce sufficiently accurate results.

Our registrar implementation requires that each voter be sent a voter identification number (which need not be secret) and a secret token T prior to the registration process. For example, university students might be given these numbers when appearing in an administrative office to have their identification card photos taken, members of a professional society might be sent these numbers in the postal mail after joining the society, or voters in governmental elections might be given these numbers when they obtain or renew their drivers licenses.

Eligible voters generate public/private key pairs and register to vote by sending the registrar their voter identification number, T, and public key. The registrar verifies that the applicants have submitted the correct tokens and adds their identification numbers and public keys to the registered voter list. The registered voter list also contains a validation field for each voter which is set to 0 before each election and changed to 1 by the validator after a voter's ballot is validated.

Pollster. The pollster acts as a voter's agent, presenting human readable ballots to a voter, collecting the voter's responses to ballot questions, performing cryptographic functions on the voter's behalf, obtaining necessary validations and receipts, and delivering ballots to the ballot box. The pollster is the only component of the Sensus system that voters must trust completely; voters concerned about the privacy of their ballots may want to install personal copies of the pollster on trusted machines. Pollsters may be implemented with a variety of user interfaces. Some pollsters may have the ability to display multimedia ballots with a graphical user interface; others will use a simple text-based interface. The pollster may also be used to assist voters in verifying that their votes were counted correctly or in contesting an election.

Our pollster implementation has a simple text user interface. It has the ability to display unvoted ballots described using our ballot description language, *BLT*. BLT was designed for maximum flexibility in generating ballots. It allows for ballot elements to be presented in random order or conditionally (although conditional presentation is not yet implemented). However, the Lisp-like format of BLT makes it difficult for humans to use it to compose Sensus ballots. This problem can be resolved through the use of a ballot authoring tool.[4]

Now that World Wide Web browsers with HTML forms support have become widespread, HTML may prove to be a better ballot specification language for most purposes.

Validator. The validator is responsible for checking voter registration and ensuring that only one vote is cast by each registered voter. The validator creates a blinded validation certificate by signing a blinded ballot. The voter then unblinds the validation certificate and submits it to the tallier with his or her ballot. The validator will issue no more than one validation certificate to each registered voter.

Our validator uses the registered voter list to obtain each voter's public key and check the signatures on their ballots. The validator changes the contents of the validation field from 0 to 1 after validating a ballot. With this method no record is kept of the order in which ballots are validated.

Tallier. The tallier is responsible for collecting the voted ballots and tallying the results of the election or survey. Voters first submit encrypted ballots, signed by the validator to the tallier. The tallier checks the authenticity of the validation and verifies that the encrypted ballot is unique among the encrypted ballots received thus far. If the ballot is valid and unique, the tallier issues a signed receipt to the voter.

[4] We have implemented a prototype ballot authoring tool called Ballot Palette. This module was implemented using C and the XView tool kit. It provides a graphical representation of a ballot and allows users to edit ballots by dragging icons with a mouse. Ballot Palette then produces BLT code for the created ballot.

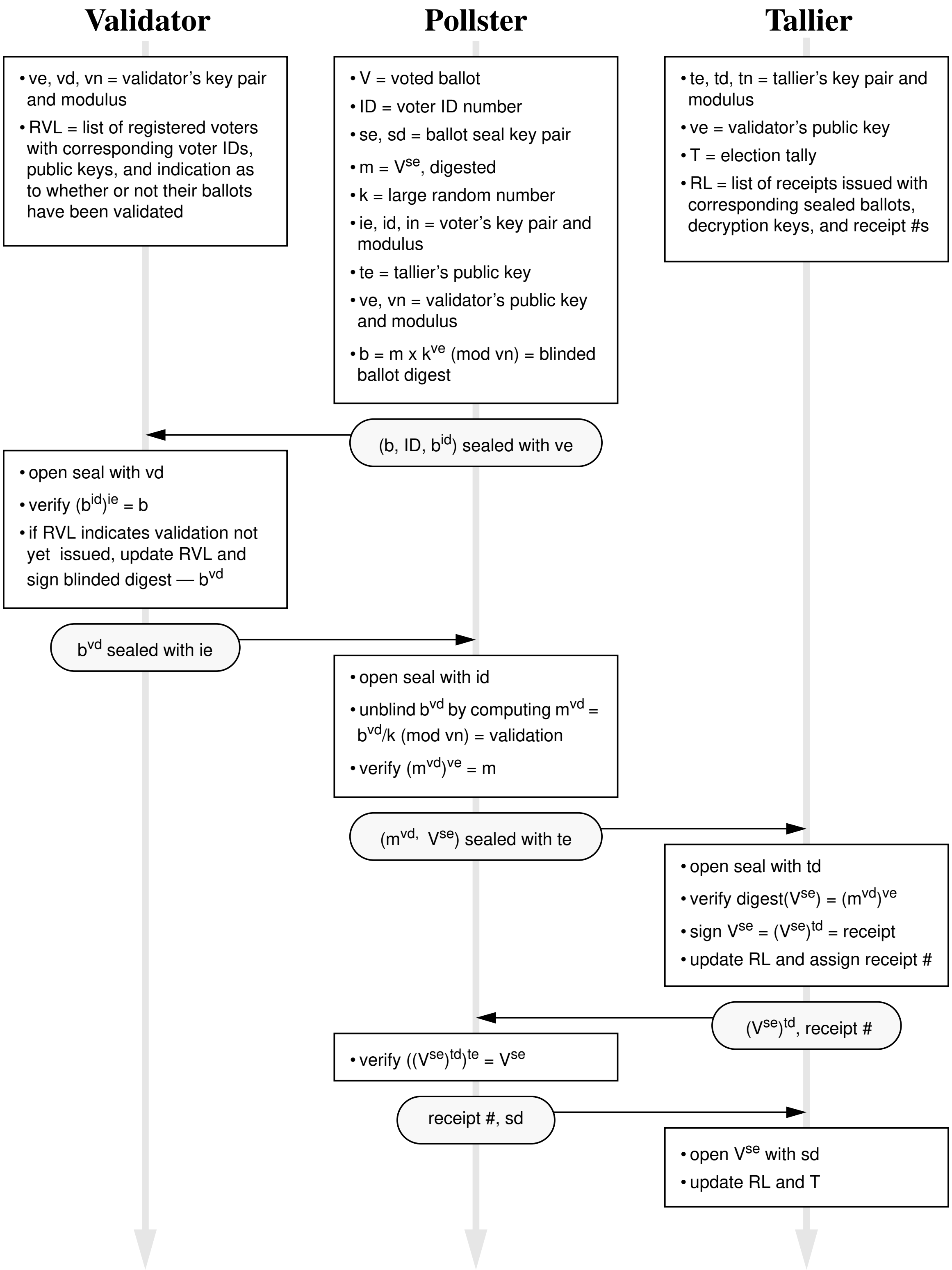

Figure 1: Sensus Protocol Overview

The voter then submits the ballot decryption key. The tallier uses the key to decrypt the ballot. After the election, the tallier publishes a list of encrypted ballots, decryption keys, and decrypted ballots, allowing for independent verification of election results.

Our tallier computes a 16-byte digest of each encrypted ballot received and uses it to index the encrypted ballots and receipts. A hash table could be added for greater efficiency in looking up encrypted ballots. This modification is probably necessary to accommodate large-scale elections.

Note that there is a very small chance that two or more voters may submit identical encrypted ballots. If this were to occur, only one of these ballots would get counted.

3.2. Other Polling Protocols

A variety of cryptographic election protocols have been proposed over the past 15 years that have been designed to minimize fraud and maximize privacy. In addition, some have been designed with additional goals in mind, such as making it impossible for voters to prove that they voted in a particular way. Many of the proposed protocols are not practical to implement for a large number of geographically distributed voters; however, they remain of theoretical interest. In this section we provide a brief analysis of several other cryptographic election protocols as well as traditional election protocols, and compare them with Sensus.

Traditional Election Systems. Most traditional election systems are far from ideal. They tend to rely on a number of trusted parties who have the ability to conspire to change the outcome of the election or reveal the way particular voters voted. These systems generally work because most of the trusted parties are either trustworthy or have little trust in each other, and thus no conspiracy takes place.

The voting systems used for national elections in the United States are generally designed to satisfy all of the core properties to some degree. However, there are few official criteria which voting systems throughout the U.S. are required to satisfy [14]. In most systems there are opportunities for votes to be changed, lost, or incorrectly recorded during the counting process. Although inaccurate tallies may be the result of fraud, all documented inaccuracies in computerized vote tallying have been the result of problems with or misuse of the voting equipment or software. For example, a 1984 Carroll County, Maryland school board election was incorrectly tallied by a computerized tallying system after an election administrator accidently installed the wrong utility program for reading ballot cards [19]. The use of absentee ballots gives national elections the mobility property, allowing voters to cast their votes from almost anywhere they want. However, absentee ballot systems tend to reduce privacy further and increase the opportunity for ballots to be changed or lost. Despite these procedural shortcomings, it would be difficult for a national election to be thrown because of the large number of precincts and the diversity of voting systems used [12]. In addition, vote buying is probably rare because it is nearly impossible for a voter to prove how he or she voted after leaving the polling booth. (However, vote buying can occur easily when absentee ballots are used.)

Most traditional election systems can be verified only by party representatives or trusted third parties. It is generally not possible for voters to verify that individual votes were counted correctly. In addition, while the verification process can often detect procedural problems and large discrepancies between the final tally and the number of voters who visited the polls, it usually cannot correct inaccuracies.

The systems used for national elections are usually also used for local elections in the United States. However, when used for local elections these systems are more likely to be abused because a relatively small number of precincts contribute to the final vote tally.

Large professional, social, and special interest organizations tend to hold their elections through vote-by-mail balloting systems. These systems allow voters to cast their votes from virtually any location, however, they often sacrifice accuracy and privacy. This method usually works because organizations that use this system tend not to hold highly controversial elections. In addition, they often hire a disinterested party to run their elections (generally at considerable expense). With appropriate precautions, vote-by-mail can be suitable for even the most controversial elections. Probably the largest non-governmental organization to use vote-by-mail balloting to date is the Teamsters. In 1988 the Teamsters sent mail ballots to 1.5 million members. According to Teamsters election officers, the only problems encountered were a few attempts to vote multiple times or intimidate voters. Nonetheless, many people are still skeptical about the security of vote-by-mail balloting. The California and Kansas Supreme Courts have both ruled on cases involving vote-by-mail balloting. In both cases the courts refused to strike down laws allowing vote-by-mail balloting, despite the Kansas court acknowledging that "vote by mail increases the potential for compromise of secrecy and opportunity for fraud" [13].

Many states also use vote-by-mail balloting for some elections, especially in small precincts. Generally voters are asked to submit their ballots in double envelopes to protect their privacy. The state of Oregon has been using vote-by-mail for some local and special elections since 1981. In December 1995 Oregon held a vote-by-mail primary election to nominate candidates for a U.S. Senate vacancy, thus becoming the first state to conduct a mail-only primary election to nominate candidates for a federal office. Oregon held the general election for the vacant Senate position by mail in January 1996, and the state's Presidential preference primary by mail in March 1996. A telephone survey conducted by the Oregon Survey Research Laboratory following the January election found that 76.5 percent of the 1225 respondents "favored vote-by-mail elections over polling place elections" [21]. The Oregon Secretary of State's Office credits vote-by-mail with increasing voter participation and reducing the cost of conducting elections in Oregon [16]. While surveys and media editorials indicate there is still some public mistrust of vote-by-mail balloting, we have been unable to find reports of any significant problems that have occurred as a result of vote-by-mail balloting.

Cryptographic Polling Protocols. Chaum published the first cryptographic voting protocol in a 1981 paper on anonymous electronic mail and digital pseudonyms [3]. This protocol uses public key cryptography and relies on rosters of digital pseudonyms to conceal the identity of voters. However, the protocol does not guarantee that the identity of voters cannot be traced. Chaum later proposed a protocol which unconditionally conceals the identity of voters [5]. However, elections conducted with this protocol can be disrupted by a single voter. Although Chaum's protocol can detect such disruptions, it cannot recover from them without restarting the entire election [11].

In 1985 Cohen (a.k.a. Benaloh) and Fischer published a description of a secure election scheme in which it is very difficult for dishonest voters to disrupt the election [7]. However, the scheme does not protect the privacy of individuals from the election authority. Cohen later presented extensions to this scheme which distribute the power of government and offer more privacy protection [6]. Benaloh claims this scheme is "reasonably practical" and cites political problems as a greater hindrance to implementation than technical problems. However, he also acknowledges that knowledge of college-level mathematics is required for voters to independently verify election results [2]. In addition, because of the scheme's large communication complexity, casting a vote may take an unacceptably long time [17].

In 1994 Benaloh and Tuinstra proposed a set of verifiable secret-ballot election protocols that do not allow voters to prove the contents of their votes [1]. Unlike the other cryptographic protocols discussed here, these protocols require voters to vote inside a voting booth. The authors maintain that the simplest of their protocols does not require computations on the part of the voter that are outside "the range of normal human ability." However, the more complex protocols that have fewer requirements for trusting election authorities would require the voter to bring a personal computing device into the voting booth. Even the Full Scale Receipt-Free Protocol does not guarantee that voters cannot be coerced, unless one or more election authorities are trustworthy. Although not a practical solution for Internet voting, the receipt-free nature of this system is significant because it prevents voters from participating in vote buying schemes.

A number of other cryptographic voting schemes have been proposed that require interactions between voters. These schemes, including [9], may be useful in a board room setting, but are not suitable for most large-scale elections.

The more practical cryptographic schemes do not require any interaction between voters or use of specialized equipment. However, none of these schemes prevent vote buying. One of the more simplistic of these schemes requires two election authorities: a validator and a tallier. In this scheme, shown in Figure 2, voters encrypt their ballots with the tallier's public key, sign them, and forward them to the validator. The validator strips off the voters' signatures, checks to make sure the ballots were submitted by registered voters who had not yet voted, and forwards the ballots to the tallier. The tallier decrypts the ballots and records the votes. This scheme prevents non-registered voters from voting and registered voters from voting multiple times. However, it only protects voters' privacy if the tallier and validator do not collude. In addition, it does not provide a mechanism for voters to use to verify that their votes were counted correctly.

In the *Two Agency Protocol* developed by Nurmi, Salomaa, and Santean [15], the responsibilities of validating registered voters and computing and publishing the results of the election are divided between two agencies, as in the simplistic scheme. In this protocol the validator distributes a secret identification tag to each voter just prior to the election. The validator then sends the tallier a list of all identification tags, with no record of the corresponding voters. Each voter sends the tallier his or her identification tag and an encrypted file containing a copy of the tag and the voted ballot. At this point the tallier can make sure the identification tag is valid, but has no way of examining the contents of the ballot. The tallier publishes the encrypted file, and the voter responds by sending the tallier the key necessary to decrypt it. When the election is over, the tallier publishes a list of all voted ballots and the corresponding encrypted files. At this point the voters can confirm that their votes were counted properly. Any voter who finds an error can protest by submitting the encrypted file and decryption key again. Because the encrypted file was published earlier, the tallier cannot deny having received it.

The Two Agency Protocol is verifiable by individual voters (unlike the simple protocol discussed earlier), however, it still has several problems. Most importantly, it does not protect voters' privacy if the tallier and validator collude. Thus, the authors state that if the two agencies are going to work together, there might as well be just one agency [18].

The *One Agency Protocol* is identical to the Two Agency Protocol, except for the tag distribution procedure. In the One Agency Protocol, tags are distributed by the tallier

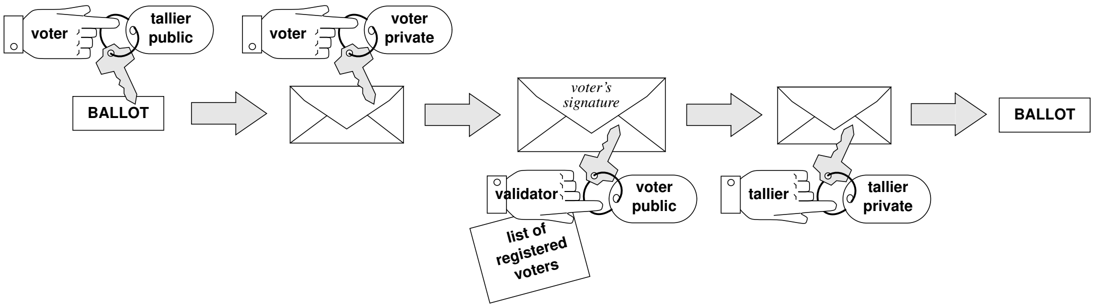

Figure 2: A Simplistic Voting Protocol

(there is no validator) using an ANDOS (all-or-nothing disclosure of secrets) protocol for secret selling of secrets. This solves the collusion problem; however, the ANDOS protocol is quite computationally complex and does not scale well. Another problem with both the One and Two Agency Protocols is that the tallier may cast votes for all voters who have been assigned a tag but do not exercise their right to vote [18, 15].

When Chaum first introduced the concept of blind signatures in 1982, he suggested that blind signatures could be used for secret ballot elections. Ten years later, Fujioka, Okamoto, and Ohta developed a practical voting scheme that uses blind signatures to solve the collusion problem inherent in protocols like the Two Agency Protocol without significantly increasing the overall complexity of the protocol [10]. The Sensus protocol is based closely on the Fujioka, Okamoto, and Ohta scheme. The main difference between these schemes emerges after the voter has submitted the encrypted ballot to the tallier. In the Sensus protocol, the tallier responds by sending a receipt to the voter. The voter may submit the decryption key immediately after receiving this receipt, completing the entire voting process in one session (verification must still wait until the election is over). In the Fujioka, Okamoto, and Ohta protocol the tallier responds by placing the encrypted ballot on a list that is published after all voters vote. Thus, a voter cannot submit his or her decryption key until after the voting phase of the election is over. As a result, votes cannot be cast in a single session.

The Fujioka, Okamoto, and Ohta protocol satisfies most of the core properties well; however, it fails to correct one of the problems inherent in the One and Two Agency protocols: the election administrators (in this case the validator) can cast votes for abstaining voters. These invalid votes can be detected by the abstaining voters themselves or by an auditor who checks the signatures on all the validation requests submitted. However, there is no way to identify the invalid ballots and remove them from the tally. If voters who wish to abstain submit blank ballots, then this problem can be avoided. Another problem with the Fujioka, Okamoto, and Ohta protocol is that it requires the voter to participate in a complex set of transactions in order to cast a vote. However, we resolve this problem in the Sensus system by including a pollster module that allows the voter to perform these transactions with ease.

4. Evaluation

We outlined seven desirable properties of polling systems in Section 2. In this section we will evaluate Sensus' ability to satisfy these properties.

While evaluating the security and privacy aspects of the Sensus system, we make a few assumptions.

- We assume that a vote cannot be linked to a particular voter by tracing the packets in which the vote is delivered to the tallier back to the sender. Thus, we assume all communication between voter and election authorities occurs over an anonymous channel. This is not necessarily the case using the current Sensus implementation; however, an anonymous channel could be secured through the use of a chain of World Wide Web forwarding servers.

- We assume that the voter is using a computer system in which it is not possible for clear text messages to be intercepted. Thus we assume no parts of the voter's computer system can be snooped through physical or electronic means. The voter's privacy while casting the vote can only be violated if the voter allows someone to look over his or her shoulder. This is, of course, not the case if the voter is using a multi-user system where other users have root privileges.

- We assume that messages from voters will not arrive at the validator and tallier in the same order, allowing the validator and tallier to collude to link ballots with the voters who cast them. This assumption is valid given a voter population large enough that multiple voters are likely to attempt to vote at approximately the same time. In addition, voters concerned about this type of collusion need not submit their ballots to the tallier immediately after obtaining validation certificates from the validator. While it is possible for the validator to delay the processing of a validation request until the previous voter has submitted a ballot to the tallier (thus ensuring that the ballots arrive at the validator and tallier in the same order), such behavior would likely be detected by voters who experience long delays.

- We assume that all encryption algorithms used are sufficiently strong that encrypted messages cannot be decrypted without the proper keys. Thus, security in the current implementation is based on the strength of RSA.

Accuracy. Sensus satisfies all three parts of the accuracy property well. While it is possible to alter, eliminate, or add votes, any such behavior is detectable. Voters whose votes have been altered or eliminated from the final tally can discover the problem by examining the tallier's published list. These voters can submit their receipts anonymously along with their ballots and decryption keys to protest the election results and have them corrected. The only party that can add invalid votes to the final tally is the tallier. These can be detected by any party who checks the authenticity of the validation certificates for all ballots; the final tally can then be corrected.

Invulnerability. Sensus satisfies the invulnerability property completely when all registered voters submit ballots. However, if abstaining voters do not submit ballots, it is possible for the validator to submit and validate ballots in their names. By checking the signatures on all of the validation requests, an auditor may detect that this has occurred and determine the number of inappropriately validated ballots. However, it is not possible to correct the final election tally.

Privacy. Sensus satisfies the first part of the privacy property well, but does not satisfy the second part at all. Given the above assumptions, it is not possible for any party to link a ballot to the voter who cast it. However, Sensus does nothing to prevent a voter from proving that he or she voted in a particular way.

Verifiability. Sensus satisfies the weak version of the verifiability property completely. Voters can verify that their votes were counted correctly and correct any mistakes they might find without sacrificing their privacy. However, it is not possible for any interested party to verify that all votes were counted correctly.

Convenience. Sensus satisfies the convenience property well, allowing voters to cast their votes quickly, in one session, and with minimal equipment, or special skills. In mock elections voters were able to mark their ballots, perform all necessary cryptographic functions, and complete all Sensus transactions within a few minutes – an amount of time mock election participants found acceptable. A more efficient implementation of modular arithmetic and other functions would likely reduce this time to one or two minutes. Although, the computer science students and professors who participated in mock elections found Sensus easy to use, incorporating the Sensus user interface with a World Wide Web browser would make Sensus more accessible to less computer literate people.

Although Sensus allows voters to cast their votes in one session, voters may choose to cast their votes in two sessions to ensure that their ballots cannot be linked back to them if the validator and tallier collude. To make this two-session voting more convenient, the pollster could be programmed to automatically begin the second session after a random delay.

Flexibility. Sensus satisfies the flexibility property well, as it was implemented using a ballot specification language that allows a variety of question and answer formats, including free text.

Mobility. Sensus satisfies the mobility property well, as it can be used from any computer connected to the Internet. The Sensus program is currently only implemented on a Unix system, but it should be easily portable to other platforms.

5. Conclusions

We have presented seven desirable properties of polling systems, as well as the design and implementation of Sensus, a practical electronic polling system that satisfies these properties. Our system protects voter privacy, even when election authorities collude. It also allows voters to verify that their votes have been counted correctly, and anonymously challenge the results should their votes be miscounted. Sensus will not accept ballots from those not registered to vote, nor will it accept more than one ballot from each registered voter. Invalid ballots can only be introduced into the final tally by the validating authority if some voters do not submit ballots.

Mock elections conducted with Sensus indicate that the system is convenient for voters to use. Voters can generally mark a short ballot and complete all cryptographic functions and transactions with election authorities within a few minutes. Integration of the user interface with a World Wide Web browser would make Sensus even easier to use.

It is important that security and privacy considerations be taken into account when designing electronic voting systems. In addition to the usual security concerns that should be considered when designing any secure computer system, voting systems have unique concerns that come about as a result of our desire to maintain voter privacy. Although neither Sensus nor the other voting protocols described here satisfy all of our desirable properties completely, some satisfy them well enough so as to be as good as or better than the traditional voting systems they may replace.

Properly designed electronic voting systems should be suitable for a variety of polling applications, including large-scale elections. Although electronic governmental elections may be a long way off, professional and social organizations have already started to conduct surveys and elections electronically. While most of these elections currently ignore privacy concerns, advances in email and Web browser software that can easily interface with cryptography software should pave the way for secure and private electronic elections in the near future.

Acknowledgments

The authors wish to thank David Chaum for discussions and advice on this work.

References

[1] Benaloh, J., and Tuinstra, D. Receipt-free secret-ballot elections. In *Proccedings of the Twenty-sixth Annual ACM Symposium on the Theory of Computing* (May 23–25, 1994), pp. 544–553.

[2] Benaloh, J. D. C. *Verifiable Secret-Ballot Elections*. PhD thesis, Yale University, December 1987.

[3] Chaum, D. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM 24*, 2 (1981), 84–88.

[4] Chaum, D. Blind signatures for untraceable payments. In *Blind Signatures for Untraceable Payments* (New York, 1982), D. Chaum, R. Rivest, and A. Sherman, Eds., Plenum Press, pp. 199–203.

[5] Chaum, D. Elections with unconditionally-secret ballots and disruption equivalent to breaking RSA. In *Advances in Cryptology - EUROCRYPT '88* (Berlin, 1988), C. G. Gunther, Ed., vol. 330 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 177–182.

[6] Cohen, J. D. Improving privacy in cryptographic elections. Tech. Rep. YALEU/DCS/TR-454, Yale University, February 1986.

[7] Cohen, J. D., and Fischer, M. J. A robust and verifiable cryptographically secure election scheme (extended abstract). Tech. Rep. YALEU/DCS/TR-454, Yale University, July 1985. Also appeared in 1985 Foundations of Computer Science conference proceedings.

[8] Cranor, L. F. Can declared strategy voting be an effective instrument for group decision-making? Tech. Rep. WUCS-95-04, Washington University Department of Computer Science, St. Louis, February 1995.

[9] Demillo, R., and Merritt, M. Protocols for data security. *Computer* (February 1983), 39–51.

[10] Fujioka, A., Okamoto, T., and Ohta, K. A practical secret voting scheme for large scale elections. In *Advances in Cryptology - AUSCRYPT '92* (Berlin, 1993), J. Seberry and Y. Zheng, Eds., vol. 718 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 244–251.

[11] Iversen, K. R. A cryptographic scheme for computerized general elections. In *Advances in Cyptology - CRYPTO '91* (Berlin, 1992), vol. 576 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 405–419.

[12] Kimberling, W. Secure against what? an approach to computer security. *The FEC Journal of Election Administration 13* (1986), 11–14. Published by the National Clearinghouse on Election Administration, Federal Election Commission, Washington, DC.

[13] Mutch, R. E. Voting by mail. *State Legislatures* (December 1992).

[14] Neumann, P. G. Security criteria for electronic voting. In *Proceedings of the 16th National Computer Security Conference* (1993), pp. 478–481. Baltimore, Maryland, September 20–23.

[15] Nurmi, H., Salomaa, A., and Santean, L. Secret ballot elections in computer networks. *Computers & Security 36*, 10 (1991), 553–560.

[16] Oregon Secretary of State Executive Office. Oregon election information. Available on the Internet at http://www.sos.state.or.us/executive/or.elec.info/orelinfo.htm, November 21 1995.

[17] Sako, K., and Kilian, J. Secure voting using partially compatible homomorphisms. In *Advances in Cryptology, Crypto'94* (1994), Lecture Notes in Computer Science, Springer-Verlag.

[18] Salomaa, A. Verifying and recasting secret ballots in computer networks. In *New Results and New Trends in Computer Science* (Berlin, 1991), H. Maurer, Ed., vol. 555 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 283–289.

[19] Saltman, R. G. Accuracy, integrity and security in computerized vote-tallying. *Communications of the ACM 31*, 10 (1988), 1184–1191, 1218.

[20] Schneier, B. *Applied Cryptography*. John Wiley & Sons, New York, 1994.

[21] Southwell, P. L. Preliminary report survey of vote-by-mail election in the state of oregon. Available on the Internet at http://darkwing.uoregon.edu/∼osrl/vbm/vbm.html, January 30 1996.