The Bell™

Privacy, Security and Technology in Internet Voting

FEBRUARY 2001 | Published Online Monthly | Vol. 2 No. 2

www.thebell.net | ISSN 1530-048X

Mission bells were used in colonial California for telling time, announcing events, and for passing on news from one city to another. Our symbol is the classic outline of a mission bell because THE BELL newsletter serves similar purposes.

Web Page

Visit The Bell's web page at http://www.thebell.net – more information, more up-to-date.

Call for Papers

Join the dialogue and submit your paper to THE BELL. See page 2. All papers are peer–reviewed. Submissions accepted at any time.

Free Subscription

THE BELL is FREE of charge for Internet distribution in PDF format, and is also available in hard copy. For information, see the back cover.

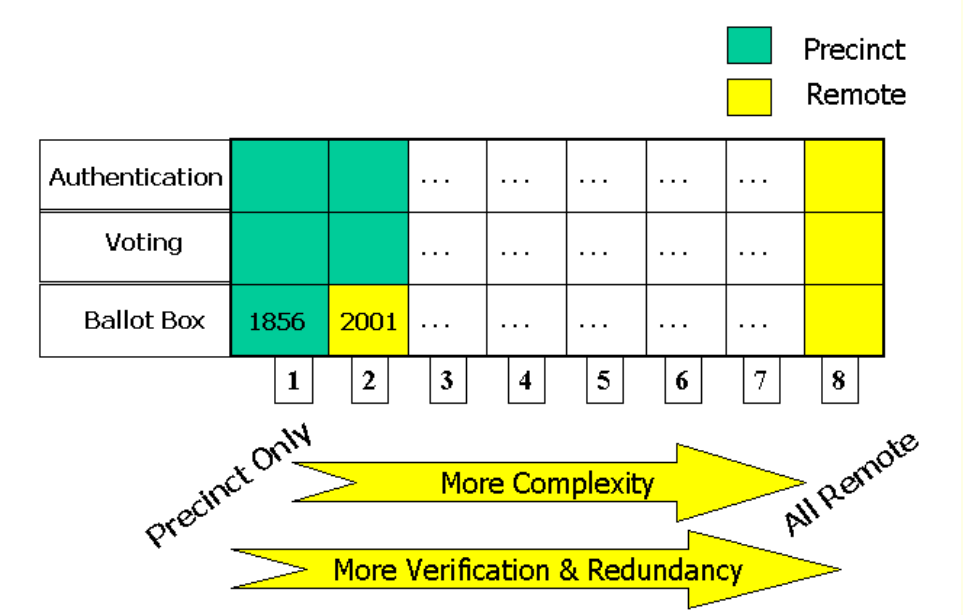
Read Also...

READ NEXT MONTH:

Trust Paradigms
CA Assembly Bill 55
CA Voting Reform
Information & Reliability

Voting System Requirements

This issue is dedicated to discussing and presenting a set of 16 voting system requirements that support fail-safe privacy, verifiable security and tamper-proof ballots, published for the first time as a draft in November 2000. This set of requirements is technologically neutral, can be applied to paper, electronic and network (Internet) voting, and exceeds the current requirements for paper-based ballots and electronic voting DRE (Direct Recording Electronic) machines. The requirements are based on the principles of Information Theory and of trust as qualified reliance on information, favoring multiple, independent channels of information over one purportedly "strong" channel. However, adding multiple channels can also decrease reliance if the design principles laid out in these requirements are not followed. It is shown that adding a physical ballot copy (e.g., in paper) to current electronic voting systems so that the voter actually casts two ballots, is ineffective and opens new opportunities for frauds and attacks. Practical implementations of the 16 requirements are also discussed in terms of a timetable for implementation of Internet voting and taking into consideration system complexity:



A precinct-based network voting system that complies with the 16 requirements can be readily deployed, with local authentication, local voting stations and remote (as well as local) ballot boxes, *with* several advantages over conventional voting systems and *without* compromising privacy or security.

THE BELL™ Newsletter
ISSN 1530-048X

Editor: Eva Waskell
editor@thebell.net

Website: www.thebell.net

Address: 1001 D Street, San Rafael, CA 94901-2800
Phone: (415) 482-9300
Fax: (415) 482-9400

Privacy: We will not forward to third parties any personal, address or credit information supplied to us by you. Any other information we may receive is treated as public and non-confidential.

Submissions: Contributions are welcome. Please see instructions at www.thebell.net/editor/.

Rights: Contents are copyright © THE BELL, 2000. "THE BELL", "SAFEVOTE" and "INTERNET DECISION MAKING" are trademarks of Safevote, Inc. All rights reserved. Permission is hereby granted for reproduction in whole for internal or non-profit use, provided that credit is given to THE BELL and to the authors of the reproduced materials. All other reproduction without the prior written consent of Safevote, Inc. is prohibited. This notice does not supercede the rights of the authors whose copyrighted materials are used by permission.

Disclaimer: The information provided in this newsletter is believed and intended to be correct and useful; however, Safevote, THE BELL, the editor, the contributors and the newsletter staff assume no liability for damages arising out of the publication or use of any material contained herein and cannot assume responsibility for the consequences of errors contained in the articles, or misapplications of the information provided.

Editorial Board: THE BELL Editorial Board has an open mandate to provide the newsletter with independent, external advisory review of both the materials to be published and the editorial line. Editorial Board members have no affiliation with THE BELL or its publisher.

From the Editor

Dear Reader:

One of the main objectives of The Bell is to provide for open dialogue and exchange of high quality, non-partisan information. In this issue we invite election officials, experts and other interested parties to join a dialogue regarding a set of 16 strict voting system requirements.

How did these requirements originate? Around 1997, Ed Gerck had seen a list of requirements for secure voting in the cryptography literature and thought that they were incomplete and inconsistent. So, he asked the question "What are the *most* desirable properties of secure voting?" as he edited and added items in the original list, sending it back and forth to Internet cryptography workgroups for comments. In September 1999, Gerck collected his findings and drafted a version that became the core 16 requirements. All along the way, questions and criticisms from people who reviewed the text were incorporated. The requirements changed somewhat as this clarification process progressed and more background information was added, but the core elements including the need for fail-safe voter privacy, voter verification, qualified reliance on information and relying on multiple channels of information over one purportedly "strong" channel remained.

The November 2000 issue of The Bell published the 16 requirements as applied to Internet voting systems. But it soon became clear that technology independent requirements were needed for voting systems in general. During the fall of 2000, the voting system requirements evolved further in the IVTA tech work group, with input from election officials, the general public, voters, public tests and other online discussion groups. To this day, the document is updated frequently, as questions or criticisms are woven into its fabric.

In this issue of The Bell, we are making a public call for comments. Election officials, worldwide, are the ones who know the nitty gritty details of an election and make elections work in spite of the faults of current voting systems. The objective of the 16 voting system requirements discussed in this issue is to make this knowledge available as a list of instructions able to be followed by a human clerk who works obediently with paper and pencil, for as long as is necessary, but without insight or ingenuity. In other words, as software to be run in computers. And where humans provide for oversight and auditing, not chad removal.

Eva Waskell
Editor

THE BELL EDITORIAL BOARD

Eva Waskell (coordinator)
Editor, The Bell
Communications Director, Safevote, Inc.
California, US

Tony Bartoletti
Information Operations, Warfare and Assurance Center
Lawrence Livermore National Laboratory
California, US

Professor Netiva Caftori
Computer Science Department, Northeastern Illinois University
Member, National Board of Computer Professionals for Social Responsibility
Illinois, US

Dr. Gordon Cook
Editor and Publisher
The COOK Report on Internet
New Jersey, US

Ed Gerck, Ph.D.
CEO and CTO, Safevote, Inc.
Chairman of the Board, Internet Voting Technology Alliance
California, US

Jason Kitcat
Founding Partner, Swing Digital Ltd.
Co-ordinator of the FREE e-democracy project
Brighton, UK

Professor Hans Klein
Public Policy Department, Georgia Institute of Technology
Chairman of the Board, Computer Professionals for Social Responsibility
Georgia, US

Voting System Requirements

by Safevote[*]

This document, published as a first draft in November 2000, presents a set of 16 voting system requirements that support fail-safe privacy, verifiable security and tamper-proof ballots. This set of requirements is technologically neutral, can be applied to paper, electronic and network (Internet) voting, and exceeds the current requirements for paper-based ballots and electronic voting DRE (Direct Recording Electronic) machines. The requirements are based on the principles of Information Theory and of trust as qualified reliance on information, favoring multiple, independent channels of information over one purportedly "strong" channel. However, adding multiple channels can also decrease reliance if the design principles laid out in these requirements are not followed.It is shown that adding a physical ballot copy (e.g., in paper) to current electronic voting systems so that the voter actually casts two ballots, is ineffective and opens new opportunities for frauds and attacks. A timetable and practical implementations are also discussed.

DISCLAIMER: This paper does not intend to cover all the details of the technologies reported, or all the variants thereof. Its coverage is limited to provide support and references to the work in progress on voting systems and to unify references, concepts and terminology. No political, country-oriented or other criticism is to be construed from this paper, which respects all the apparently divergent efforts found today on the subjects treated. Individuals or organizations are cited as part of the fact-finding work needed for this paper and their citation constitutes neither a favorable nor an unfavorable recommendation or endorsement.

This information is hereby licensed by Safevote, Inc. to be used freely for commercial and non-commercial purposes, which use must visibly acknowledge this license and Safevote, Inc. as the licensor. This information is not in the public domain.

Background

As defined by Alan Turing some fifty years ago, a mathematical method is *effective* if it can be set out as a list of instructions able to be followed by a human clerk who works obediently with paper and pencil, for as long as is necessary, but without insight or ingenuity. Together with Alonzo Church, Turing argued that every *effective* mathematical method can be carried out by a sufficiently powerful computer (represented by the universal Turing machine).

These Voting System Requirements were born out of the desire to create products that would allow modern computer-based technology to truly emulate the secure desirable properties valued in centuries of public voting. In other words, can we use a perfect clerk in elections – one who works obediently with paper and pencil, for as long as is necessary, but without insight or ingenuity?

Indeed, if perfect clerks would conduct an election using paper-ballots, this would provide the best model we have for a public election. Such an election would be, for example: **anonymous** (avoiding collusion, coercion), **secret** (all cast votes are unknown until the election ends) and yet **correct** (all votes are counted) and **honest** (no one can vote twice or change the vote of another), oftentimes also **complete** (all voters must either vote or justify absence). In such a system, if we know the voter (e.g., in voter registration) we cannot know the vote and if we know the vote (e.g., in tallying) we cannot know the voter. After an election, all votes and all voters are publicly known – but their connection is both unprovable and unknown.

But real-life clerks are not perfect. Neither are computers. So we need to introduce the concept of qualified reliance on information in terms of providing proofs (e.g., proof of voting, proof of correctness) that can be objectively evaluated and not just subjectively accepted or taken at face value.

To discover and rate such proofs, the Requirements employ the idea that one should favor multiple, independent communication channels over one "strong" channel – which idea was successfully used by the Moguls in India some 500 years ago in the context of combating corruption [1], and was mathematically described by Claude Shannon some 50 years ago in the context of combating noise when he introduced his Information Theory [2], a well-known general theory of communication processes.

---

[*] Copyright © Safevote, Inc. and THE BELL, 2001. See copyright notice on p. 2, The Bell. Original Article: *Safevote (Ed Gerck, Editor), Internet Voting Requirements, The Bell, Vol. 1 No. 7, p. 3, November 2000, ISSN 1530-048X,* available at www.thebell.net/ archives/thebell1.7.pdf. Current version archived at www.thebell.net/papers/vote-req.pdf. Send comments at any time to vote-req@safevote.com

Thus, for example, how could a voting system prove that the vote received at the ballot box is the same vote seen and cast by a voter? This question is not easier to answer if the voter is close to the ballot box, or far away. Distance plays no role, contrary to what one might think at first. The essential problem is that the voter is not at the ballot box, hence the voter has no way of knowing if what was sent through that communication channel (which may be very short) was what was received.

To solve this question in electronic voting some advocate printing a paper copy of the ballot, which the voter can see and verify that it is identical to the ballot she intended to cast, and then sending the paper copy to ballot box A while an electronic copy of that same ballot is sent to ballot box B. The idea is that ballot box B could be tallied quickly while ballot box A would be used as a physical proof for a manual recount. Such a suggestion is oftentimes advanced as the *sine qua non* solution to voting reliability in electronic voting.

But what makes the introduction of a paper ballot special is not the fact that it is paper instead of bits. It is the fact that the voter is actually casting his vote twice. We now have two independent channels of information for the ballot, one from the terminal as source B, the other one from the printer as source A. So we have N = 2.

In other words, this design provides for two outputs: ballot A and ballot B. However, in the event of a discrepancy between the two, no resolution is possible from within the system. Technology provides no answer in this case. Thus, between two conflicting outputs, **each of which is the result of a trusted system**, the decision of which "is correct" must be made **independently of the system**, by policy.

This means that the paper/electronic system does not work when it is needed most – i.e., when the system reveals that it is not a "perfect clerk."

The situation can thus be summarized:

- If the system would always be similar to a perfect clerk then N = 1 (one channel) would suffice, whether paper or electronic. But if we use a system with N = 1, we cannot define any level of reliance on the final result except that which was assigned a priori.

- If we add one independent channel (e.g., the paper ballot) to a system that already provides one channel (e.g., electronic ballot), this creates a system with N = 2. However, this additional channel makes the system indeterminate and still incapable of, by itself, defining any level of reliance on the final result except that which was assigned *a priori* (e.g., paper is more trustworthy).

Clearly, before considering other well-meant suggestions (which might be similarly ill-fated), what is necessary is to seek a logically provable solution to reliability problems caused by imperfect communication systems.

Such a solution needs to consider not only **machine-machine** communication channels but also **human-machine** communication channels because the voter can act as a source and as verifier in more than one part of the system. Further, **human-human** communication channels must be considered because we do not want machines to have the potential to run amok, unchecked.

Information Theory [2] can be used to describe such communication channels and, as previously noted, the concept of qualified reliance on information can be introduced as a formal definition of trust [3] in order to rate such channels in terms of providing proofs.

As a result, the only provable solution to increase reliability in communications (e.g., the communication between the voter as a sender and the ballot box as a receiver) turns out to be to increase the number/capacity of independent channels until the probability of error is as close to zero as desired (direct application of Shannon's Tenth Theorem in Information Theory [2]). To be complete, the solution considers not only machine-machine communication channels but also human-machine and human-human. Thus, if an electronic system is able to provide N proofs (human and machine based), these N proofs for some value of N larger than two will become more reliable than one so-called "physical proof" – even if this one proof is engraved in gold or printed on paper.

On the other hand, by using two ballot boxes A and B, what can we say if they show different vote totals? How would we know which one is correct? There is no way to know. And, of course, anything printed on paper can be faked, whether money or votes. The same goes for bits. Even if policy says otherwise in either case.

Therefore, adding a paper ballot copy to current electronic voting systems is ineffective because in the event of two conflicting outputs from each trusted system, the decision of which one "is correct" must be made outside the system and a priori. It also presents opportunities for fraud (e.g.,someone can change and/or delete some paper ballots after the election in order to cast doubt on the integrity of the entire election) and attacks (e.g., a group of voters might agree beforehand to callout a "discrepancy" after they vote and thereby disrupt an election, which is similar to a "denial of service" attack).

Thus, we need a real-world voting system – one that is not based on perfect parts (N = 1) or produces an undefined result in the case of a single error (N = 2).

In order to provide for qualified reliance on information, such a voting system needs to obey two requirements:

- its communication channels must include three types of channels: human-machine (ballot channels), machine-machine (transmission channels) and human-human (audit channels), **and**

- it must have more than one independent channel for each type of channel. These channels can include any media such as electronic records, paper, microfilm, CD-ROM, etc.

The principle is that **multiple independent channels can be used to correct errors in one channel, but one channel cannot correct errors in itself.**

In plain English, the greater the number of independent channels for the verification of a result, the greater trust the result may have.

However, suppose the terminal where the voter enters his choices will change them to something else and then send this information over N different channels, what difference does it make if it is N = 1, 2 or 500?

None. In such a case N would still be 1 **for the ballot channel**. The 2 or 500 channels are not independent **for the ballot** because they all originate as copies from that single stored corrupted ballot. So, it does not make a difference in terms of ballot reliance. This would, however, make a difference in terms of communication reliance, in which there are now different **transmission channels**, 2 or 500 channels for which each channel could behave as a correction channel for another – meaning in this case that the ballot box would more probably receive the right ballot (even though corrupted) for N = 500 than for N = 1.

What is needed is thus a requirement to include several truly independent ballot, transmission and audit channels – **whether or not electronic transactions are used** – and use these channels to rate the reliance on each node of an end-to-end balloting system, even during the election and in real time. There should be several ways to implement this requirement and channels could be added also in time and context, not just in space. Channels can also transport information by reference, not just information by value.

What is also needed is a way to allow the voter to verify results, for example the presence/absence of her ballot at the ballot box and whether her ballot at the ballot box is a valid one. This is useful because sufficient indirect verification does produce trust. **"Trust but verify"** is in our collective wisdom and it is definitely applicable here. It is important to note that even if just a fraction of the voters (e.g., 5%) do verify the results, the capability of verification is already a deterrent to fraud because a fraudster has no way of knowing who will verify, or not.

Another characteristic of a good voting system is that the only person whom you prove the vote to is the voter. If the proof can be shown to someone else, then the vote can be coerced or sold. Therefore, when using multiple channels of information, they either have to be deniable by the voter or else temporary so that the voter cannot be threatened or hurt as a result of the vote.

Regarding the use of paper, it is important to note that the reason to distrust a paper/electronic voting system with N = 2 is **not** based on a distrust of paper. Paper is just another communication channel. The reason is that adding paper does not solve the problem and makes the problem indeterminate. The reason is thus that we need N > 2. Certainly, paper can be one of the channels, if desired, because the channel make-up is irrelevant. But a cost-benefit analysis might result in the use of non-paper channels.

Another question that must be addressed is thus the possibility of all-electronic voting systems. Should we trust them and why?

Nowadays, all-electronic systems and computers are used to fly commercial and military jets. And yet, no one in the public is afraid that a terrorist will introduce a virus in the system and down all commercial jets worldwide, or all U.S. military jets. Why? Because there is a **designed** redundancy at many levels in the system. For example, there are three independent laser inertial navigation sensors and any decision on the plane's position depends on the agreement of at least two, which decision is further verified by a GPS system, as well as flight time and speed calculations.

Thus, voting systems –like any other system – derive their trustworthiness from the fact that they work consistently, both conceptually and perceptually. However, in the absence of an easy conceptual understanding of the system (e.g., a laser inertial navigation sensor) that the average user could grasp, a sufficiently coherent perceptual understanding (e.g., it works) is enough to eventually build trust in the system.

Trust may also be denied by the design itself, because disasters may occur at any time if the principles of communication reliance (i.e., trust itself) are not taken into account. Imagine a plane that would be flown with just two navigation sensors, one compass-based and the other electronic –we would then have an idea of the disastrous consequences of using a paper/electronic voting system with N = 2, even though a physical channel is used (compass, paper).

Thus, the deciding factor in trusting a system is not whether it includes one or even two sources of information that can be touched or seen in physical form (e.g., a paper in your hands, a paper behind a screen, a compass needle behind a screen).

A factor that mitigates against an all-electronic voting system is the fact that although paper and electronic records are both vulnerable to subversion, it is a lot easier to change what is in an electronic record than it is to change what is on paper.

Thus, electronic records need to be bound to other references in a manner that is demonstrably inaccessible to an attacker, both through physical access controls and through cryptographic protocols.

Moreover, there really needs to be a step-by-step description of the voting process, so that when someone asks, "What if the intruder succeeds in breaking into the system to change X?" this can be clearly answered, for example, by:

- To change X would cause a subsequent binding failure, thus it would be detectable except with parallel access to Y and Z, which are independently inaccessible, or
- Knowledge of an alternate (and attacker-desirable) value for X is insurmountably difficult to achieve, and the effort could not be leveraged to any other X.

Put most plainly, people know that ordinary voting systems can be subverted by someone who could bribe enough individuals to collude, but the physical fact of several tons of paper ballots still represents somewhat of an obstacle to an "easy subversion" in the eyes of many.

In contrast, people are well aware that electronically one can modify a million records with as little as a few keystrokes. This is the "fear" that needs to be addressed in an all-electronic system – that such a subversion can be so massive and rapid, executed from the safety of a remote laptop, etc. that it would be unavoidable.

Of course, one alternative to reduce fear would be education. To educate voters regarding the very nature of distributed cryptographic assurances and at a level where the concepts are not clothed in excessive abstractions.

But cryptography is not by itself the critical issue, nor the silver bullet. And no amount of education will stop attackers, while it may aid them.

Instead, voting systems can use the concept of multiple independent communication channels to make it **as impossible as desired** to tamper with the electronic ballot both before and after it is cast.

Here, the question is not how many copies of paper or bits one has, but how many independent channels the attacker needs to subvert versus how many independent correction channels one has available during such an attack. Of course, if the attacker is able to subvert the correction channels while attacking the other channels, then they would not be independent.

Therefore, **the same mechanism that protects casting the ballot must also be used to protect presenting the ballot**. And this needs to be given as a set of Requirements that work together in an end-to-end design. The make-up of each channel's carrier (e.g., paper, bits, electrons) is by itself irrelevant.

These Requirements are therefore general principles, valid for any physical implementation of a "ballot" – whether as print marks on paper, pits on a CD-ROM surface, electrons hitting a video screen (electronic ballot), modulated electromagnetic waves, bits in a network protocol or any other form of information transfer to and from the voter. They also apply to any form of voting, including majority voting and single transferable votes. The Requirements may be wholly applied or just a subset may be used.

To achieve these goals, the Requirements should be able to handle voting rules of any type and could apply to voting systems anywhere in the world. However, the main objective here is for the Requirements to be as complete and independent from one another as possible, without sacrificing consistency. It is understood that "completeness" is an elusive goal that might never be reached when we consider the diversity of election needs [4], while "consistency" is a necessary feature for the Requirements to work together in a particular election. In short, this was the reason to stop the Requirements with # 16. Increasing the number of Requirements could risk decreasing their consistency, in general. Of course, other Requirements may be added, or deleted, as needed.

Some of the words used in the Requirements may have different (and equally valid) meanings in other contexts (e.g., "voter privacy"). Therefore, the Requirements also include the operational definitions for the main words used. Three words are, however, used without a definition even though they could also be misunderstood. These words are "trust" [3], "manifold" and "meshwork" [5], as defined in the references.

1. Privacy Considerations

One principle followed in the development of these Requirements is that they MUST NOT reduce the privacy, security and integrity features of paper ballot voting systems operating under best conditions – even if such features are not legally required. The justification is that if we have learned to rely upon such features when using paper ballots, then they should be maintained when using other types of ballots (e.g., electronic).

In this regard, we consider the most important requirement of a voting system to be **voter privacy**. To clearly define this concept, we need to discern not only different types of voter privacy but also different "strengths."

The different types of voter privacy arise out of the need to know who the voter is during voter registration and in voter lists after voting in spite of the fact that we MUST NOT know who the voter is when the vote is known. These

apparently conflicting requirements can be met by defining voter privacy in voting systems as "the inability to link a voter to a vote."

Thus, even though both the voter (before and after the election) and the vote (after the election) MUST be well-known, **the connection between voter and vote MUST be unknown at all times** (even a long time after the election).

Voter privacy (i.e., the inability to link a voter to a vote) is the most important property of a voting system because once it is compromised, coercion and collusion cannot be avoided and therefore no other requirement can be assured.

The different strengths of voter privacy arise out of the need to protect voter privacy against different attacks or faults. We note that a perfectly implemented paper-ballot voting system is able to protect **voter privacy** even in the event of a court order that would mandate otherwise. Thus, these Requirements consider that **voter privacy** needs to be likewise protected –i.e., to the same degree.

To make these issues clearer, we first distinguish four degrees of voter privacy strength (policy, computational, information-theoretic, and fail-safe) as a function of what needs to be broken in order to compromise voter privacy in each case. Next, we rank voter privacy, ranging from lowest to highest strength, as a function of the strength of what we need to rely upon in each case.

Policy privacy. Exemplified by election systems that depend on election officials and/or separated machines in order to protect voter privacy. In one such case, the voter's identity and actual vote selection are separated at the voter's computer and sent to two different servers, which information is then prevented by policy to be rejoined. This measure guarantees that neither the election server computers nor a third-party are able to determine the way in which a voter voted, but does not prevent the election host's administration from doing so. Intrusion detection and log files overseen by a supervising service cannot help against collusion with the supervising service. Policy privacy also cannot prevent attackers from penetrating both systems and rejoining the information. It also cannot prevent a court order that would mandate rejoining the information in the servers.

Computational privacy. Exemplified by election systems that rely upon a threshold of collusion in blind signatures, mix-servers or homomorphic encryption, such that not less than N people working together can compromise voter privacy. However, such systems rely not only on the flawless implementation of mathematical formulas but also on the absence of a compromise to the computational platform (e.g., a virus that would record all N keys for later use, or a non-intrusive electromagnetic eavesdropping device that would record all N keys for latter reuse without ever penetrating the platform). It also cannot protect voter privacy in the case of a court order that mandates revealing all keys or secrets used in the system.

Information-theoretic privacy. Exemplified by election systems in which there is no reliance on cryptography in order to protect privacy (e.g., reliance on RSA encryption). It defines a privacy strength that cannot be broken by computation, even with unbounded time and resources, in contrast to systems that would only provide for "computational privacy"(i.e., privacy which could be broken by computation, given time and resources) [6]. Information-theoretic systems include, for example, systems where: (a) parties share keys in advance and use one-time pads (which is impractical and in any case subject to collusion attacks where keys are revealed); (b) parties share physically protected channels (this fails against collusion attacks where the channel is compromised without detection); (c) parties share information (via secret-sharing techniques) and they are assumed not to pool it together (again this fails against collusion attacks). Information-theoretic privacy also cannot protect voter privacy in the case of a court order that mandates revealing all keys and secrets used in the system.

Fail-safe privacy. Defined here for election systems where voter privacy cannot be compromised even if everything fails including software and hardware, everyone colludes and there is a court order that mandates revealing all keys and secrets used in the system. Current paper ballot voting systems provide for fail-safe voter privacy. Another example of a system with fail-safe voter privacy is provided by Safevote's implementation of the Requirements using a meshwork system (Section 8) [7]. By not revealing the private voter information to the voting system, in any form, and yet providing an anonymous control structure with the DVCs (Digital Vote Certificates) and the EBs (Electronic Ballots), **Safevote's meshwork system relies upon data which can be mathematically verified but whose private relations are unknowable by design [7].**

The table below summarizes the four cases discussed above, ranking them in increasing degree of privacy strength.

Privacy Strength	Relies Upon	Attacks
Policy	correctly following a communication protocol defined by policy	1. circumvent policy 2. collusion 3. court order
Computational	mathematical expressions and their correct implementation in software/hardware	1. break the code 2. obtain the keys 3. collusion 4. court order
Information-theoretic	unprovable (non-computable) relations	1. obtain secrets 2. collusion 3. court order
Fail-safe	unknowable relations	none

2. Summary of Requirements

A voting system, whether using paper, electronic recording or networks such as the Internet, needs thus to satisfy various requirements, which are summarized in 16 main points. Implementations and examples are discussed in other sections.

1. ***Fail-safe voter privacy.*** Definition: "Voter privacy is the inability to link a voter to a vote." Voter privacy MUST be fail-safe – i.e., it MUST be assured even if everything fails, everyone colludes and there is a court order to reveal all election data. Voter privacy MUST be preserved even after the election ends, for a time long enough to preserve backward and forward **election integrity** (e.g., to prevent future coercion due to a past vote, which possibility might be used to influence a vote before it is cast).

2. ***Collusion-free vote secrecy.*** Definition: "Vote secrecy is the inability to know what the vote is." Vote secrecy MUST be assured even if all ballots and decryption keys are made known by collusion, attacks or faults (i.e., vote secrecy MUST NOT depend only on communication protocol and cryptographic assumptions, or on a threshold of collusion for the keyholders).

3. ***Verifiable election integrity.*** Definition: "Election integrity is the inability of any number of parties to influence the outcome of an election except by properly voting." The system MUST provide for verifiability of election integrity for all votes cast. For any voter the system MUST also provide for direct verifiability that there is one and only one valid ballot cast by the voter at the ballot box.

4. ***Fail-safe privacy in verification.*** If all encrypted ballots are verified, even with court order and/or with very large computational resources, the voter's name for each ballot MUST NOT be revealed.

5. ***Physical recounting and auditing.*** MUST provide for reliability in auditing and vote recounting, with an error rate as low as desired or, less strictly, with an error rate comparable or better than conventional voting systems [8]. The auditing and vote proofs MUST be capable of being physically stored, recalled and compared off-line and in real-time during the election, without compromising election integrity or voter privacy, and allowing effective human verification as defined by election rules.

6. ***100% accuracy.*** Every vote or absence of vote (blank vote) MUST be correctly counted, with zero error [8].

7. ***Represent blank votes.*** MUST allow voters to change choices from 'vote' to 'blank vote' and vice-versa, at will, for any race and number of times, before casting the ballot.

8. ***Prevent overvotes.*** As defined by election rules. MUST provide automatic "radio button" action for single-vote races. If overvoting is detected in multiple-vote races, MUST warn the voter that a vote has to be cleared if changing choices is desired. This warning MUST be made known only to the voter, without public disclosure.

9. ***Provide for null ballots.*** As defined by election rules, MAY allow voters to null races or even the entire ballot as an option (e.g., to counter coercion; to protest against lack of voting options). Overvoting, otherwise prevented by Requirement #8, MAY be used as a mechanism to provide for null ballots.

10. ***Allow undervotes.*** As defined by election rules, the voter MAY receive a warning of undervoting. However, such a warning MUST NOT be public and MUST NOT prevent undervoting.

11. ***Authenticated ballot styles.*** The ballot style and ballot rotation to be used by each voter MUST be authenticated and MUST be provided without any other control structure but that given by the voter authentication process itself.

12. ***Manifold of links.*** MUST use a manifold [5] of redundant links *and* keys to securely define, authenticate and control ballots. MUST avoid single points of failure – even if improbable. If networks are used, MUST forestall Denial-of-Service (DoS) and other attacks with an error rate comparable or better than conventional voting systems [8].

13. ***Off-line secure control structure.*** MUST provide for an off-line secure end-to-end control structure for ballots. MAY use digital certificates under a single authority. Ballot control MUST be data-independent, representation-independent and language-independent.

14. ***Technology independent.*** MUST allow ballots and their control to be used off-line and/or in dial-up and/or in networks such as the Internet, with standard PCs or hand-held devices used to implement their components in hardware or in software, alone or in combination for each part.

15. ***Authenticated user-defined presentation.*** MUST enable the ballots to dynamically support multiple languages, font sizes and layouts, so that voters could choose the language and display format they would be most comfortable with when voting as allowed by law and required by voters with disabilities, without any compromise or change to the overall system, from an authenticated list of choices defined by election rules.

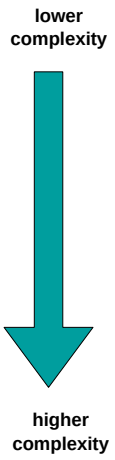
16. ***Open review, open code.*** Allow all source code to be publicly known and verified (open source code, open peer review). The availability and security of the system must not rely on keeping its code or rules secret (which cannot be guaranteed), or in limiting access to only a few people (who may collude or commit a confidence breach voluntarily or involuntarily), or in preventing an attacker from observing any number of ballots and protocol messages (which cannot be guaranteed). The system SHOULD have zero-knowledge properties (i.e., observation of system messages do not reveal any information about the system). Only keys MUST be considered secret.

3. Timetable and Implementation

This set of 16 strict requirements can be used with various forms of precinct-based voting, including network (Internet) voting. They can also be used to experiment with various forms of remote voting, including voting from home.

As further discussed in Section 5, three components are essential to voting systems during an election: voter authentication, the voting station and the ballot box. If we consider that each of these three components may be either local (i.e., at the precinct) or remote (i.e., elsewhere, at home), then voting systems in general can be classified in eight types, as a function of component location (where "network" can be a dial-up connection or also the Internet):

TYPE	Authentication	Voting	Ballot Box
Precinct	local	local	local
Precinct-based Network	local	local	remote
...	...		
...	...		
...	...		
...	...		
...	...		
Network	remote	remote	remote



While complexity increases as shown above, verification and redundancy may also increase. Thus, higher complexity does not necessarily imply higher risk. For example, one can use redundancy to increase reliability [2] and verification to increase trust [3], thereby reducing risk. The entire set of eight types cannot be uniquely ordered in terms of complexity, except for the three types named in the table. The Requirements can be applied to all eight types of voting systems, from precinct voting to voting from one's home, with varying degrees of risk as a function of implementation.

In the table above, precinct voting (with all components local) provides for a 100% controlled environment and potentially least risk. **A precinct-based network voting system is the next step in terms of risk**, where only the ballot box is remote. Precinct-based network voting uses networks (dial-up, Internet) to connect the remote stations to each another and to the local stations. The voter authentication stations as well as the voting stations are situated in a precinct and the ballot boxes are remote (but some ballot boxes with ballot copies MUST also be local as defined in the Requirements).

Precinct-based network voting also provides controlled training opportunities for all involved and was recently demonstrated in official public tests by Safevote [9].

It is important to note that the security problems reflected in the increased potential risk as we deploy remote components do not disappear with encryption or even with a secure protocol such as the Secure Socket Layer (SSL). If the voter is led to connect to a spoofing site, which appears to be what he wants, he may have a secure connection to a fraudster and that will not make it safer. Thus, voter authentication must also address the question of preventing spoofing.

Further, digital certification by a Certification Authority (CA) according to X.509 or PKIX protocol is not enough to combat the spoofing problem. A CA Certificate may give the voter the impression of a correct document, as in a spoofing situation, that was accepted by the voter's browser as in a standard Certificate format, with "exact" but false information about the CA, its name and e-mail. Spoofing can also affect addresses in the IP (Internet Protocol) layer, not just names in the DNS (Domain Name Service) layer, which is much harder to be detected by a voter.

However, we note that security considerations regarding the perils of spoofing, virus and Trojan horses do NOT apply to well- implemented precinct-based network voting systems.

In terms of a timetable to implement Internet voting for public elections, these Requirements can be readily applied to precinct-based network voting systems – for example, as already demonstrated by Safevote in 2000 [9].

Certainly, as will be discussed elsewhere, there are ways to avoid spoofing (both DNS-based as well as IP-based) in the other six types of Internet voting so that the probability of spoofing can be reduced to a value as close to zero as desired. However, given the lingering political questions inherent in public sector remote Internet voting (e.g., the digital divide) and the need to move gradually for a host of reasons (training, legal requirements, procurement procedures, system certification, tests, trust, etc.), such discussion is premature. Practical examples of the effective prevention of spoofing will most probably first be seen in banking applications and in private sector Internet voting.

Since part of a voting system may thus use networks (dialup, Internet) to interconnect the stations involved, there is a great concern about security and privacy. The distinguishing quality of the Internet is that no one can control both ends of a connection, neither sending nor receiving data. Opposite points may be under opposing controls. In particular, this makes it impossible to "measure" or authenticate the persons or processes at the remote end of an interaction, or even in-between, either sending or receiving. It is difficult to trust that which we cannot control.

However, ***the Internet can support reliable and secure transactions, and does so regularly, as long as all endpoints of the transactions are under the control of a single authority*** – even if multiple keys are used. People are generally unaware of this quality because it is not the

"standard mode of operation" employed by the public in web browsing or sending an email.

Therefore, the reader is advised that as long as the endpoints fully control the cryptographic key agreement and node addressing schemes used, the "Internet as a transfer medium" is indeed extremely reliable in accurately delivering opaque blobs of encrypted and certified data.

Implementation of the 16 Requirements can be done in several different ways, for each type of voting system. Having studied many alternatives, Safevote defined a **basic architecture** called "Multi-Party Protocol ™" as a particular design choice, which this document presents as a reference to help explain the requirements. This architecture, however, is not the only possible choice.

Implementors of these requirements are free to use other design choices. The particular implementation provided here is thus merely a reference, an example. Any other implementation may also imperfectly or partially represent a Requirement. We may see many different implementations of the same set of Requirements, which designs may all be useful for one reason or another.

In Safevote's design, Denial-of-Service (DoS) attacks, as well as other network-based attacks such as Large Packet Ping, Buffer Overrun, TCP SYN Flood, IP Spoofing, TCP Sequence Number, IP Fragmentation and Network Penetration can be forestalled by using the stealth, moving target technology described in http://www.safevote.com/tech.htm.

The same technology can be applied to prevent attacks that target the DNS protocol, not the just the IP protocol. Features such as the Digital Vote Certificates (DVC) and the Electronic Ballots (EB), also developed by Safevote, ensure voter privacy, vote secrecy and election integrity within the scope of legal requirements for public elections. Other choices, such as the use of touch screen technology and an open PC-based architecture, help provide for usability and low cost.

The design used by Safevote is proprietary (U.S. patents pending 60/225996, 60/226042, 60/226158, 60/231600, 60/231681 and others) and can be licensed by other companies for commercial applications. The design, including source code, is also available with zero cost for non-commercial applications.

4. Precinct-based Voting

This document focuses on the requirements for precinct-based voting where voting stations are set up in polling places. This also includes Direct Recording Electronic (DRE) voting machines. Network (e.g., Internet) voting systems based on suitable PC-based hardware architectures may have a fraction of the cost of DREs while adding oversight and remote redundancy – functions that DREs do not have. A DRE operates alone, supervised only by itself and there is thus no clear reason to trust the vote count. DREs fail to prove that the counted vote is really what the voter voted. Current DREs could be ideal "con machines" for voters – a DRE has no witness to its acts but itself. Lack of network connectivity is therefore not a positive factor for DREs, contrary to what one may think at first sight.

The benefits of precinct-based Internet voting systems following these requirements include 100% accuracy, large and clear ballot layout, tamper-proof ballots before and after casting the vote, physically redundant secure ballot copies, zero paper ballot costs, more voting places for the voter (because the voter is not limited to voting at one precinct close to home), reliability tabulating results, voter verification that her/his/ ballot is being counted, real-time auditing, effective human-verified recounting, and easy integration with state-wide or national tabulation. Some, but not all, of these benefits can be provided by current DRE voting machines. These benefits cannot be provided by paper ballots, whether using punch cards or optical scan systems.

Even though the voting stations are not remotely located in precinct-based network voting, the use of digital certificates as a substitute for human-based control, "electronic ballots" in lieu of the traditional paper ballots, and remote ballot boxes in these Requirements demands careful analysis. This document will now discuss the various components of an election system necessary to provide the level of privacy and security mandated by public elections, as defined by the Requirements.

5. Remote Ballot Boxes

Voting systems comprise five main components: (1) a **registration service** for verifying, registering and providing authentication means (e.g., credentials) to legitimate voters; (2) **voter authentication** stations, which may be provided together with a voting station, with the task of determining a voter's authorization to vote based on the voter credentials supplied by the registration service; (3) **voting stations** where the voter makes choices on a ballot; (4) a device called the **ballot box** where the ballot is collected; and (5) a **tallying service** that counts the votes and announces the results. Additionally, voting systems have other components such as auditing, ballot generation, ballot management, recounting and storage.

Of the five main components listed above, only (2), (3) and (4) are used **during** an election. The first component is used before the election begins while the last component is used after the election closes. A registration system is always used even if it is a "same day registration service" or simply a "no questions asked" system. This is not a semantic question, whether we should call "registration" that which does not register, but simply a recognition of the fact that there must be a registration policy (even if the policy is absence of policy)

and that a service thus exists which enables the voter to vote based on such policy – for example, by giving the voter a credential or a ballot.

Privacy, security and integrity of voting systems depend critically on these parts securely working together in combination, not just in isolation. However, these parts are not usually located at the same place and do not work together at the same time. The problem is to guarantee that their isolated actions do correspond to a system-wide policy, at all times and at all places, even though such actions may be remotely located to each other and done at different times.

Central to this problem is the ballot itself. Ballots need to be controlled from inception to end when they are generated, viewed, approved, controlled, distributed, voted, collected, tallied, audited, recounted and stored. Oftentimes ballots need to be defined per voter (e.g., when ballot rotation is used to reduce positional bias) or per class of voters (e.g., due to geo-political, regional and language differences).

Also, law usually demands that whenever ballots are handled at least two election officials or authorized poll workers must be present. At the end of an election, law demands that all unused ballots must be accounted for. Various control procedures are used – e.g., with methods that provide only one-way access to the ballot box so that votes cannot be deleted or changed once they are cast. Further, ballots are usually confined within the physical limits of a poll site, under the supervision of poll workers. When transported to a tallying place, ballots are also physically protected.

Controlling the ballot is a well-defined task for paper ballots which can be physically designed as they will appear to a voter, physically printed, physically transported, physically counted and so on. The foregoing aspects may be protected by physical tamperproof seals, and may always involve the presence of at least two authorized officials. In order to compromise a paper ballot, several people need to be involved and collude in an entire mesh of events.

However, in Direct Recording Electronic (DRE) devices used in electronic voting, there is usually no paper to be printed and none to use for control. After the ballot is cast, a DRE may record anything as the ballot, not necessarily that which the voter voted. There are no unused ballots to be counted at the end of the election. The very definition of a "ballot" comes into question. What was the ballot image seen and used by the voter?

This problem is well-known in conventional systems for electronic voting. Virtually all DRE systems on the U.S. market have been certified according to the Federal Election Commission (FEC) Voting Systems Standards by accredited bodies. (For a complete list of certified systems, see www.electioncenter.org/about/nased.html). These standards define how ballot images are to be stored in such systems, and certified, before the elections. These standards are also specific about the storage of individual voter ballot images after voting, for example, as given in section 2.3.2 of the Standards, under "Accuracy and Integrity." However, in DREs there is no way of knowing whether that ballot image is really the ballot image seen and cast by the voter.

In conventional voting systems, whether paper-based or electronic, ballots may thus be provided by means of printed paper, non-volatile computer memory, CD-ROM, magnetic media, or any other way of storing and presenting the ballots, together with means to store the ballots cast until the votes are tallied. Conventional voting systems include several ways to protect the ballot and its voted contents from tampering or accidental errors.

However, dial-up and/or Internet networks can be used together with computers in order to define some aspects of ballot usage locally and also remotely, for example and as given in this document, by storing encrypted cast ballots at the local machine and also at servers which are not located at the poll site (and which might serve one or more poll sites). This may include local ballot printing (at the precinct) and also remote printing, in **paper**, **microfilm** or **other media**.

Thus, in the case of electronic voting using the Internet or any other network (hereafter called network voting or Internet voting), ***the definition of an "electronic ballot" needs to consider electronic voting systems where the ballot boxes, the voting stations, the voter registration service, the tallying stations and/or any other component of the voting system may be remotely situated.***

Driving this evolution from paper-based voting to network voting are economic, political and social factors. For example, considerable cost reduction can be achieved by sharing costs in a client-server system with electronic data transfer, while providing increased voter participation due to greater availability of voting places for a voter, reduced physical transportation and physical security needs for transportation of ballot boxes, and reduced time for tabulating results.

Assuring the integrity of the election becomes difficult, however, when the ballot is handled remotely, outside the poll site. Locating ballot handling outside the poll site is at odds with the notion of protecting the ballot as conventionally done in poll sites. Locating the voting stations outside poll sites (remote voting) makes it difficult, for example, to verify whether the ballot is being correctly presented at the voter's home screen for example.

Further, using the Internet for voting, voters may trivially and unavoidably change the window size where the ballot is displayed. Also, voters may independently change its colors, font size or other features. The operating system, which may vary from voter to voter in remote voting, may also display the

ballot image differently than what was intended. Other aspects that come into play when ballots are handled outside controlled environments are, for example: to protect the identity of the voter, to prevent the voter from voting twice, to prevent attacks that might deny access to a remote ballot box or from a remote voting station, to assure vote secrecy and voter privacy, to provide for auditing and vote recounting.

Automation and networking also increase the possibility of widespread fraud, viruses, Trojan horses, hacker attacks, or simple bugs in software which can unfairly bias the results of an election. Both vote accuracy as well as reliability may sharply decrease when either one or both the ballot box and the voting station are located outside the same poll site.

Thus, several difficulties arise to potentially undermine the integrity of elections in network voting, many of which are posed by questions related to the very definition of a ballot and how it is controlled. Methods used in conventional systems to assure integrity of elections by physically controlling the number and style of ballots, whether in paper or provided by a machine at the poll site, cannot be applied. Unsupervised handling of ballots is also unavoidable. Viruses and Trojan horses, for example, may easily change what a voter sees on a screen in spite of the correct information being sent by a central server that dispenses valid ballots, or in spite of what a voter may type on the keyboard, even when using an encrypted communication channel between the machine used by the voter and the server.

6. The Wallet Approach And Verifiability

In conventional systems, one solution to this problem is the "wallet"approach used in e-commerce. This solution can be recognized in some Internet voting systems involved in public tests. In this type of solution, a communications system is defined in which a particular Object (a "control structure," for example, a Java applet or a browser plug-in – also called a "wallet" in e-commerce protocols such as SET) is installed or transferred from a server (ballot server) to a client (voting station). That control structure is a special piece of software which encapsulates ***both*** (1) data fully defining the intended server-client communications relationship, including the ballot; and (2) methods for using the data to effect that relationship, including reading and verifying a voter's digital certificate, displaying the ballot, reading the voter's choices and casting the ballot. After it is transferred to the client, the Object as a control structure runs on the client side to control communications between client and server. According to this solution, once the client computer receives such an Object, it needs only to access the Object's methods. The Object handles all communications details including cryptographic or other modules, which modules may be embedded in the Object itself or referenced by means of digital signatures to certified modules outside the Object.

The problems with the "wallet" approach are well-known in the art in e-commerce and include, for example: long downloads of hundreds of kilobytes or even some megabytes of data; unreliable behavior because the client stations may not adequately support the resources required by the "wallet" (e.g., memory, browser version); the need for frequent version changes due to bugs or discovered attacks leading to a repeated need to download ever newer versions of the "wallet" (with all the time, certification and cost penalties involved); the need to rely on the user to install the "wallet" without interruptions even in the event of varying access speed; and heavy traffic load on the "wallet" server. In fact, the shortcomings are so severe that the "wallet" approach has already been abandoned in e-commerce.

In voting, there are further shortcomings to the "wallet" approach. Contrary to e-commerce, voting demands anonymity. Also, voting cannot be protected by insurance against fraud such that a certain level of fraud can be accepted as "the cost of doing business." Thus, there must be official verification and public trust that the "wallet" does not include malicious code or covert channels that would either bias the vote or reveal the voter's identity – which verification and trust are time consuming to achieve and yet must be repeated for each new version of the "wallet." Possibly, this process will actually incur growing distrust as problems are discovered, as has been experienced in e-commerce applications of similar technology.

Also, Trojan horses or computer viruses at the client side may easily subvert the "control structure" and render useless all efforts to control the ballot, how it is presented and how it is voted. Again, in e-commerce this may be compensated by insurance, which is not the case in network voting.

Further, although the order of candidates may be rotated and/or incremented in the ballot to prevent bias, anyone who sees the first ballot and gets access to the source code, to the downloaded binary image or to the file image of the code will be in a position to learn the assignment table between candidates and accumulators that carry the vote, in total or in part. Thus, a virus or Trojan horse attack could be designed even with ballot rotation and could be used to bias the outcome of the election.

These forms of attack become greatly facilitated when the source code is open (as oftentimes and herein required for security purposes), when the system uses the Internet for any of its parts, when the software is available for study in compiled form in electronic voting machines or servers used for demonstrations prior to the election, when the attacker may gain access to the software by any means, or when the election extends over several days –as many as twenty-nine days of voting are being considered for Internet voting, for example.

Another problem in conventional voting systems, especially important in network voting, is to provide for verifiability, i.e. to be able to verify whether a particular voter voted – but without revealing how or too precisely when/where that voter voted.

Verifiability may be performed by auditors (in auditing) or by anyone, including the voters themselves in what is called "universal verifiability." Methods that provide for verifiability while an election is in progress are useful to help deter fraud, minimize costs after a fraud is discovered, contain a discovered fraud, and even allow an election to proceed normally after a fraud attempt is discovered. However, methods that provide for verifiability usually rely on a listing of the voter's real name, which is not only ambiguous (i.e., for common voter names such as "John Smith") but also introduces the question whether that voter's name does correspond to one and only one vote. Still other methods try to solve the question of identical voter names by assigning "voter codes" to each voter, which raises the question whether such assignment has not been compromised by a hacker attack or unwittingly, by a "bug".

Another flawed method is one that lists the voters' real names (or linked voter codes) together with their respective encrypted ballots and intends to provide voter anonymity by tallying the ballots without decrypting them (called homomorphic encryption). But even if the encryption is perfectly secure, such method cannot protect voter privacy in the case of a court order that would ask to decrypt the ballots one by one. This method's privacy assurance relies on one weak link – that individual ballot encryption will never be broken. Besides a court order, unauthorized decryption (e.g., by an attacker, by collusion, by design faults, by lost or stolen passwords) or unintended decryption (e.g., using an area in memory as the accumulator, which area is watched for increments as each vote is tallied, by a virus or Trojan horse) is hard to prevent and impossible to disprove in conventional systems, especially because voted ballot information needs to be kept for a long time (due to legal, auditing and recounting needs).

7. Voting Protocols

Other types of cryptographic voting protocols have been proposed, the suitability of each type varying with the conditions under which it is to be applied. However, the questions introduced by remote ballot boxes and *electronic ballot* management, which need to be handled both *before* voting (e.g., ballot certification, ballot distribution to voters, ballot style authentication) and *after* voting (e.g., auditing of tallied ballots, auditing of ballot images), are not discussed in such protocols – which protocols are thus useful only for a limited theoretical discussion of voting, without a more comprehensive theoretical modeling or even a practical network voting system in mind. These voting protocols include schemes using mix-nets, homomorphic encryption, and blind signatures.

Clearly, for electronic voting on networks where the ballot box and/or the voting stations are remotely situated, these issues need to be revisited with a fresh perspective. Problems facing remote ballot control cannot be overlooked.

Voter authentication, including ballot style authentication and ballot rotation, must also be addressed. One fundamental aspect of voter authentication is that while a voter Registration Service has the voter information that can authenticate the voter, the voter does not need to be authenticated by the Registration Service when voting. The voter needs to be authenticated where the ballots are given or received. This creates a problem similar to the fundamental problem in communication systems [2], because usually the ballots are given and received at a different location than that where the voter registration information is verified and stored.

Generally, one recognizes two levels of authentication: "simple authentication", using a password or a PIN (Personal Identification Number) as a verification of claimed identity; and "strong authentication", involving digital certificates formed by using cryptographic techniques. Simple authentication (PIN, passwords) does not meet the Requirements presented in this document.

8. The Meshwork System

What is needed first is to change paradigms and avoid the "Fort Knox Syndrome" model exemplified by the dictum "make it stronger!" so widely seen in conventional security designs, including the "wallet" design. The "Fort Knox Syndrome" model fails to solve the problem of how to provide secure network voting because in this model the entire chain can still be compromised by failure of one weak link – even if that link is made stronger. And the addition of any link, even if very strong, would not make the system less vulnerable, and might make the system more vulnerable because the security of the system would still depend on the weakest link (which might be the newest link). Further, such solutions are actually based on ***the impossible assumption that "no part will fail at any time"*** –because if a critical part fails, the system fails.

The design followed by Safevote uses multiple links arranged in time and space, which links build a special manifold of closed control loops (***a meshwork*** [5]) under the principle that every action needs both a trusted introducer and a trusted witness, thus spawning two non-deterministic trust chains for every action. ***Closed control loops allow trusted properties in the system to rely on self-trust, which is the only trust that is always verifiable.*** By making the issuer of a *Digital Vote Certificate* (*DVC*, see http://www.safevote.com/aboutus.htm) also become the

final verifier of the same *DVC*, an off-line end-to-end security system can be built that is always verifiable.

The meshwork system closes the "loop of trust" and forces selected endpoints of the transactions to be under the control of a single authority (who may use split keys) – e.g., the election officials.

Further, by introducing suitable redundancy and information-theoretic [6] "one-way walls," one can use these principles to design systems that can be as secure as desired, i.e. with an error rate as close to zero as desired [8] for a set of properties. For example, one can design a ***collusion-free system*** by adding enough parts so that even if M parts fail at the same time (e.g., all key-holders collude), the system will still provide the expected functionality. One can also design a ***fail-safe system*** where M parts can fail at any time, even if M is the entire number of parts.

In terms of risk models, the closed-loop meshwork system used by Safevote implements a multi-risk design – exemplified by the following simple equation that formally represents risk in first order analysis:

A = B * C

where A = average amount lost, B = probability of failure and C = value at stake.

Clearly, one can equally well call "risk" either A or B or C. Each choice leads, however, to a different risk model. If risk = A, this means that one is talking about the security of the system as a whole, for a certain period of time, in terms of average loss. If risk = B this means that one is talking about an individual transaction in terms of probability of loss of one event for a significant ensemble of events. If risk = C this means that one is talking about risk as total loss over a period of time long enough so that one can observe repeated failure events until losses have eroded all that was at stake (being nothing else left, the loss ends there, hence the risk).

In the closed-loop meshwork system used by Safevote, the risk model that considers risk = C is called "fail-safe" for any risk beyond C, because C is all there is ever to be lost in the system even if every link fails.

Thus, rather than seeking "infinite protection" or "absolute proof" by means of one link (which is clearly impossible and mitigates against the "wallet" solution as well as the "printed vote" solution to electronic voting), **a secure network voting system could provide for a measure of protection as large as desired by using an open-ended(i.e., non-deterministic, adaptable) closed loop meshwork of links, each link individually affording some "finite" protection and collectively contributing to higher-orders of integrity in closed loops of trust.**

However, it is important to note that adding channels (even physical channels) can also decrease reliance if adequate design principles are not followed.

9. Implementation Example

According to the specifications herein and restricted to controlled voting stations in precinct-based Internet voting, such a system can be built today, providing for DoS protection, voter anonymity, vote secrecy and election integrity in Internet voting, as exemplified in tests conducted by Safevote – such as the public test in Contra Costa County [9].

COMMENTS

This proposal evolved during public discussions at the Internet Voting Technology Alliance (IVTA) beginning in September-November 2000, and also included input from election officials, the general public, voters and public tests.

During the course of this work, it was important to provide also a reference implementation of the Requirements. Any list of requirements that is not given together with running code, at least as one reference implementation, may be actually impossible to implement and thus cannot be used to define any standard. Standards cannot be figments of our imagination but realities which must already be implemented and working with all operational factors taken into account.

These Requirements were edited by Ed Gerck, including comments and references from Tony Bartoletti, Thomas Blood, Netiva Caftori, Gordon Cook, Hal Dasinger, Hugh Denton, Rosario Gennaro, Jason Kitcat, Brook Lakew, Elaine Maurer, Don Mitchel, Erik Nilsson, Michael Norden, Marcelo Pettengill, Roy Saltman, Bernard Soriano, Gene Spafford, Einar Stefferud, Arnold Urken, Eva Waskell, Thom Wysong, the IVTA tech WG (http://www.mail-archive.com/tech@ivta.org/), the CPSR-activists list, several cryptography lists, contributions from comments collected at Safevote's website and from articles published in The Bell (www.thebell.net).

These Requirements are intended for voting systems. While there may be some overlap, different sets of requirements apply for other election components, such as voter registration systems and election reporting. For example, voter registration systems need to positively identify the voter and allow voter names to be public information.

REFERENCES

[1]*"... one of the earliest references to the security design I mentioned can be found some five hundred years ago in the Hindu governments of the Mogul period, who are known to have used at least three parallel reporting channels to survey their*

provinces with some degree of reliability, notwithstanding the additional efforts." Ed Gerck, in an interview by Eva Waskell, "California Internet Voting." The Bell, Vol. 1, No.6, ISSN 1530-048X, October 2000. Available online at http://www.thebell.net

[2] Shannon, C., "A Mathematical Theory of Communication." Bell Syst. Tech. J., vol. 27, pp. 379-423, July 1948. Available online at http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html. Shannon begins this pioneering paper on information theory by observing that "the fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point." He then proceeds to so thoroughly establish the foundations of information theory that his framework and terminology have remained standard practice. In 1949, Shannon published an innovative approach to cryptography, based on his previous Information Theory paper, entitled Communication Theory of Secrecy Systems. This work is now generally credited with transforming cryptography from an art to a science. Shannon's Tenth Theorem states (cf. Krippendorf and other current wording): "With the addition of a correction channel equal to or exceeding in capacity the amount of noise in the original channel, it is possible to so encode the correction data sent over this channel that all but an arbitrarily small fraction of the errors contributing to the noise are corrected. This is not possible if the capacity of the correction channel is less than the noise."

[3] *"When we want to understand what trust is, in terms of a communication process, we understand that trust has nothing to do with feelings or emotions. Trust is that which is essential to communication, but cannot be transferred in the same channel. We always need a parallel channel. So the question is having redundancy. When we look at the trust issue in voting, it is thus simply not possible to rely on one thing, or two things – even if that thing is paper. We need to rely on more than two so we can decide which one is correct. In this sense, the whole question of whether the Internet is trusted or not is simply not defined. The Internet is a communication medium and whatever we do in terms of trust, it is something that must run on parallel channels."* Ed Gerck, testimony before the California Assembly Elections & Reapportionment Committee on January 17, 2001, in Sacramento. Assemblyman John Longville (D), Chair. For an application of this model of trust to digital certificates, see "Trust Points" from http://www.mcg.org.br/trustdef.txt excerpted in "Digital Certificates: Applied Internet Security" by J. Feghhi, J. Feghhi and P. Williams, Addison-Wesley, ISBN 0-20-130980-7, p. 194-195, 1998.

[4] This is similar to the situation found in Goedel's incompleteness theorem. The Requirements form a logical system of some complexity and thus we do not expect such a system to be both complete and consistent.

[5] "Manifold" means a whole that unites or consists of many diverse elements and connections, without requiring these elements and connections to depend upon one another in any way. "Meshwork" is used to denote a manifold in the context of the Multi-Party protocol designed by Safevote to implement the Requirements. A meshwork builds a meta-space in relationship to a space. In other words, a meshwork describes relationships about a space, not the space itself.

[6] *"We say that information-theoretic privacy is achieved when the ballots are indistinguishable independent of any cryptographic assumption; otherwise we will say that computational privacy is achieved."* In Ronald Cramer, Rosario Gennaro, Berry Schoenmakers, "A Secure and Optimally Efficient Multi-Authority Election Scheme," Proc. of EUROCRYPT '97. Available online at http://www.research.ibm.com/security/election.ps

[7] E. Gerck, "Fail-Safe Voter Privacy", The Bell, Vol.1, No.8, p. 6, 2000. ISSN 1530-048X. Available online at http://www.thebell.net/archives/thebell1.8.pdf

[8] "Accuracy" and "reliability" are used here in standard engineering terminology, even though these different concepts are usually confused in non-technical circles. Lack of accuracy and/or reliability introduces different types of errors:

(i) Reliability affects a number of events in time and/or space, for example, errors in transfers between memory registers. We know from Shannon's Tenth Theorem [2] that reliability can be increased so that the probability of such an error is reduced to a value as close to zero as desired. This is a capability assertion. It does not tell us how to do it, just that it is possible. This is the realm of Requirements #12 and also #5, where one can specify an error rate as low as desired or, less strictly, an error rate "comparable or better than conventional voting systems".

(ii) Accuracy affects the spread of one event, for example whether a vote exists. Here, Requirement #6 calls for 100% accuracy. The Requirement is that no "voter-intent" or "chad" or "scanning" issue should exist – which is feasible if, for example, each voting action is immediately converted to a standard digital form that the voter verifies for that event. Accuracy error can be set to zero because 100% accuracy is attainable in properly designed digital systems that (e.g., by including the voter) have no digitization error.

For an illustration of the above definitions of accuracy and reliability, see the four diagrams in http://www.mcg.org.br/coherence.txt

[9] "Contra Costa Final Report" by Safevote, Inc. Available upon request. Summary available at www.safevote.com

THE BELL c/o Safevote, Inc.
1001 D Street
San Rafael, CA 94901-2800

FIRST-CLASS MAIL
U.S. POSTAGE PAID
SAN RAFAEL, CA
PERMIT NO. 896

DATED MATERIAL
Please Expedite

FIRST-CLASS MAIL

Voting System Requirements

See p. 3

To enter your FREE monthly subscription, visit the website www.thebell.net or use the form below.

cut here **M A I L O R D E R F O R M** **cut here**

Enter your one year monthly subscription to THE BELL: visit the website www.thebell.net or fill out the form below

Privacy Notice: We will not forward to third parties any personal, address or credit information supplied to us by you.

NAME/TITLE_____

COMPANY_____

ADDRESS_____

E-MAIL_____

[ ] PDF 12-Month Subscription – FREE

[ ] Printed 12-Month Subscription – $ 30.00 SUBJECT TO AVAILABILITY

[ ] Year 2000 Public Sector U.S. Market Intelligence Study, 200+ pages – $ 850.00 SUBJECT TO AVAILABILITY

[ ] Hard-copy Six Issues of THE BELL, 96 pages, from May to October/2000 - $ 15.00 SUBJECT TO AVAILABILITY

INSTRUCTIONS: Mail completed order form to the address below. Pay by CHECK or MONEY ORDER, payable to Safevote, Inc. Allow two weeks for processing.

THE BELL c/o Safevote, Inc.
1001 D Street
San Rafael, CA 94901-2800