A Modular Voting Architecture ("Frogs")

by
Shuki Bruck (CalTech, bruck@paradise.caltech.edu)
David Jefferson (Compaq, jefferson@pa.dec.com)
Ronald L. Rivest (MIT, rivest@mit.edu)
August 18, 2001

Abstract:
We present a "modular voting architecture" in which "vote generation" is performed separately from "vote casting."

Introduction
A key security question about any voting system is: "Why should the voter have confidence that his vote is actually counted as he believes it was cast?"

With paper-based systems (e.g. op-scan) the voter directly creates the final ballot, and can in principle verify it before casting it. Although the scanning may introduce errors, the ballot itself remains as a physical audit trail that is the official record of his vote.

With electronic DRE systems, there is normally a layer of electronic mechanism between the voter and the official record of his vote. This mechanism both creates the record and displays it back to the voter. The voter is at the mercy of this mechanism; should it lie to him, the voter is cheated. The voter's situation is much like that of a blind person who must rely on someone else to vote for him.

In principle, we should be able to build trustworthy electronic voting systems. In practice, this is surprisingly difficult. It seems much harder than building trustworthy electronic commerce systems.

One reason trustworthy voting systems are hard to build is that it must be impossible for a voter to prove to someone else how he voted. This prevents a voter from being coerced or paid off by someone else. Therefore a voter can not receive a receipt for his vote, nor can election officials publish the names of all the voters associated with their ballots. Votes must be either anonymous or unreadable (encrypted) once cast. This makes voting more challenging than electronic commerce, where receipts and well-labeled audit trails are the norm.

The "crucial step" of voting is thus the instant when the voter casts his vote. At that instant the vote ceases to be connected with and under the control of the voter, and becomes an anonymous vote in a collection of similar ballots. With an electronic system, how does the voter know that his vote isn't changed by malicious software at this instant? Unfortunately, he can't---if the voter could identify and confirm his vote in the pool of "anonymous" votes, then he could prove how he voted to someone else.

The "vote-casting" mechanism must perform the following operations in a trustworthy and reliable manner:

1. Allowing the voter to see (privately) what vote is about to be cast on his behalf,
2. Having the voter affirm that these are indeed his choices,
3. Making an indelible record of the cast vote
4. Marking the record with appropriate authentication information (in a way that doesn't compromise voter privacy)
5. Removing as necessary any indication of the voter's identity.

(This list presumes that the voter's access to the vote-casting equipment is controlled; if not then there is an additional step of checking the voter's authorization to vote and determining the appropriate ballot style.)

It is challenging to make reliable and trustworthy equipment. The difficulty increases non-linearly with the complexity and number of tasks it must perform. Complexity is the enemy of reliability and of security. Security engineers have repeatedly learned the hard way that only path to success is to "keep it simple". Very simple. Very very simple.

Accordingly, the design proposed here identifies the essential functionality of the vote-casting mechanism, and proposes a voting system in which the critical vote-casting component is separate from other components, such as vote generation. In this way the vote-casting component can be kept as simple as possible.

We thus have a "modular" architecture, where vote-generation and vote-casting are separate modules or components. This may feel familiar to the voter who is already used to filling out his op-scan ballot at one station and having it scanned at another.

Many advantages arise from such a modular voting system architecture. One of the most important is that having clean standardized interfaces between modules allows voting systems to be composed of components made by different manufacturers.

The AMVA ("Frog") Voting System Architecture

We now give some details of the proposed architecture. The architecture could be instantiated in a number of possible ways.

We propose that ballots retain a discrete physical form, rather than recording votes only in the guts of some electronic system. Each ballot is recorded on an object we call a "frog". (This is not an acronym; it was chosen to be a neutral term with convenient clip-art for slides.)

We imagine as a preferred implementation a small card (say the size of a business card) containing a memory chip with a few thousand bytes of nonvolatile memory, costing perhaps 20 cents to manufacture. The frog supplies read/write memory, and also has a "lock" or

"freeze" capability that freezes the contents so that they may no longer be changed. (The card can also not be unlocked once locked; imagine blowing a small fuse controlling the "write" circuitry on the card.) This is a "dumb" memory card with a lock feature, not a smart card containing a processor.

The frog is thus a replacement for an op-scan paper ballot or a punch card. It is nonetheless electronic and digital, so that it can be read reliably. (No "chad"!) The system uses *blank* frogs---there are *no* printing costs as for paper ballots. Frogs unused in one election can be used in the next. Frogs are small, so storage costs are minimal. At the end of the election, frozen frogs are retained as an audit trail and for recounts.

The data format inside the frog is that of a simple text file (example given later).

There are three distinct steps a voter goes through to vote:

1. *Signing in.* Voter obtains an initialized frog.
2. *Vote generation.* Voter fills the frog with his choices.
3. *Vote casting.* Voter confirms his choices, freeze the frog, and deposits the frog in the "ballot box."

This is similar to what happens with op-scan voting.

We now review each step in a little more detail.

Signing in

Here the voter identifies himself to an election official, who checks that he is registered to vote. The election official obtains a blank frog, and initializes it. The initialization writes the following information on the frog: the election, the precinct, the date of the election, the ballot style (which determines which races the voter is eligible to vote for), any rotation (candidate ordering) parameters, the ID of the election official, and the language to use. The identity of the voter is not recorded.

Initializing a frog is done with a small device operated by the election official, who must have an appropriate key to operate the device. (There are many possible variations on this theme, including having a large number of frogs pre-initialized.)

Initializing a frog is similar to having a ballot "printed on demand".

Our scheme is compatible with the use of state-wide registration databases, allowing voters to voter in any precinct in the state. (However, the voter may lose some privacy in doing so. Even though his vote may be published with other votes from his home precinct, his vote is signed by the vote-casting equipment where he voted.)

Vote generation

The voter takes his initialized frog to a vote-generation station. Here the voter makes his choices and has them recorded on the frog. The voter is given generous feedback, and may change his or her vote easily.

The emphasis at the vote-generation station is on ease-of-use for the voter. There may be a touch-screen interface. Blind voters would have audio available. The vote-generation may have the look and feel of the next generation of DRE equipment, very well done.

But the vote generation station does not have the security responsibility that has been segregated into the vote-casting component. Thus the vote-generation equipment may be sophisticated (read "complex" i.e. "not simple"). Certification of the vote-generation equipment is much less crucial than for the vote-casting equipment, the main concerns being a reasonable degree of relability, usability, and conformation to frog recording standards, with no need for high security. We thus imagine that vote-generation equipment may evolve rapidly.

The vote-generation equipment checks that the voter has made a selection for each race (or explicitly decided to make no such selection).

The vote generation equipment writes the voter's choices onto the frog, in a format that is a public standard.

Vote-casting

The voter removes the frog from the vote-generation equipment and places it in the vote-casting equipment. This may be physically adjacent to the vote-generation equipment, but is unconnected to it.

We imagine that the vote-casting equipment may be purchased by the states or counties involved and used for many years. It utilizes a standard interface, and performs a very simple set of functions; these functions stay the same for many years. On the other hand, the vote-generation equipment may be leased and may change noticeably from year to year, as improvements in user interfaces are introduced.

The vote-casting equipment works as follows:

1. At the beginning of election day, the vote-casting equipment is initialized by inserting one or more cryptographic signature keys, each on a separate smart card.
2. It accepts a frog into its "frog reader".
3. It reads all of the data on the frog, and displays it without any omissions, alterations, or interpretations. (It allows the voter to scroll up and down if necessary to see everything.)

4. It allows the voter to push a button named "Cast my vote" or "Don't cast my vote," depending on whether the contents of the frog accurately represent his final vote choices or not.
5. If the voter pushes the "Don't cast" button, the frog is ejected from the frog reader, unaltered. The voter may then re-use the vote-generation equipment to revise his selections as he wishes, and then return to the vote-casting equipment.
6. If the voter pushes the "Cast" button, then the following steps are taken:
    a. One or more cryptographic signatures are added to the end of the data file on the frog. (More discussion of this later.)
    b. The frog is "frozen" so that it may no longer be altered.
    c. The frog is dropped into the "frozen frog bin" containing all of the cast votes.
    d. An electronic copy of the entire contents of the frog is transmitted exactly out a standard serial port to one or more "vote storage units".
7. At the end of the day, the cryptographic keys are removed from the vote-casting equipment, and power is turned off.

Discussion

Vote-generation

The vote generation equipment should be exceptionally flexible and user-friendly. (This also makes it complex.) It should accommodate voters with disabilities, provide plenty of feedback, and gently remind the voter if he has not voted in some races.

It should understand the header information in the frog that specifies the ballot style, rotation parameters, language, etc., and work appropriately. It should be able to handle "write-in" votes.

It should itself provide an opportunity for the voter to review his vote, although this review is backed-up by the second and final review possible at the vote-casting station.

The vote-generation equipment should allow the voter to insert a partially or completely filled-out (but unfrozen) frog, and to modify any or all of the selections therein indicated. (For example, we are not so terribly opposed to having political parties distribute pre-initialized frogs that have "suggested" selections made already, although this viewpoint is probably too controversial.) It could be required that a voter spend a minimum amount of time at the vote-generation station, in order to prevent him from being coerced to move quickly and observably through that station, so quickly that he wouldn't have time to change the pre-initialized ballot. Perhaps the equipment should insist that the voter review and confirm each of his selections individually.

Vote-storage units

The vote-storage units are small units devoted to storage and reliability. They may consist of a small processor and a memory unit (e.g. flash RAM). They store the votes in random order in their memory. At the end of the day, the votes they contain can be examined for tallying.

In a given precinct there may be several such vote-storage units, for redundancy. It is conceivable that different political parties could provide such units, so they each get their own copy of the votes cast at the end of election day, since we consider the votes themselves, without voter identification, to be public. (These units only need to be certified that they don't record the time of day or other information that might help identify the voter who cast a particular vote.) The vote-casting unit could broadcast the votes to each of the storage units simultaneously.

Witness cards

The vote-casting equipment has slots capable of holding up to six (say) smart-cards or the equivalent (e.g. a PCMCIA card). Each such card contains a cryptographic key and is capable of producing digital signatures.

The idea is that each such card is an "electronic witness" to the voting process. Each card can add its digital signature to the end of the ballot before it becomes an official record. The signature means that "I saw this ballot validated and deposited by a qualified voter during the election."

Because the cards are inserted at the beginning of the election, and removed at the end of the election, it becomes impossible for someone to use the vote-casting equipment by itself off-hours to "forge" ballots. Ballots can only obtain the requisite set of signatures by being submitted during election day, when all of the witness cards are in place.

The reason for allowing multiple witness cards is as follows: In addition to the election official, the political parties and perhaps some neutral organizations (e.g. the League of Women Voters) can each have their own electronic witness. (This could be arranged merely as a courtesy, or could be formalized so that sufficiently many such signatures were required before a vote would be considered valid. In any case, a sufficient number of signatures by witnesses of the election officials would be required.) In this way each organization can assure itself that no votes can be cast without being witnessed by its own card. Since each party brings its witness card at the beginning of election, and removes it at the end, they can assure themselves that no "stuffing of the ballot box" can happen before the election begins, or after it ends.

The witness cards themselves have no memory, other than temporary working memory. A witness card receives a hash of the vote, and signs that with a deterministic signature scheme (no time-dependence is allowed that might allow a card to mark a ballot).

The witness card of the election official should probably be randomized, so that identical ballots don't yield identical signed results. There must be public standards and certification for these signature cards.

State-freeness

We note that the vote-casting equipment here is entirely "state-free". It repeats the same simple set of operations over and over, without remembering anything about what it has done or having to know anything specific about the particular election. It merely reads the frog, has the voter confirm the contents, gets digital signatures from the witness cards, drops the frog in the box, and sends out copies to the storage units. It doesn't need to know the date, the time of day, the election it is in, etc. It is really quite dumb and simple, and thus can be made quite trustworthy.

To emphasize this point, note that there is no distinction made between "test mode" and "real mode" for the vote-casting equipment. We have always wondered why putting a piece of voting equipment into "test mode" and evaluating its behavior should imply anything meaningful whatsoever about its behavior during a real election. Especially when these devices are increasingly software based; it is a trivial programming exercise to have such a device behave honestly when it knows it is in "test mode", but to surreptiously change 3 percent of the votes when it is in "real mode." By eliminating the distinction between test mode and real mode altogether, we eliminate this attack possibility altogether.

The witness cards should also be certified to be stateless.

Simplicity and openness of the vote-casting device

The vote casting device is a "computer", but not a general-purpose one. It, and its software, should be as absolutely simple as possible. It should not be nearly as complex as a standard PC, for example. It needs only a touch screen, a slow processor and bus, minimal working RAM, and only one or two kinds of I/O port (e.g. serial, USB, or PCMCIA); it needs no rotating storage devices, no network card, no sound card (except for units for the handicapped), no advanced graphics, and no clock, no keyboard, and no mouse.

It should have all of its software in ROM, and that software should be as simple as possible—in particular it should not be based on a full-featured operating system, but rather the vote casting app should be built in as an embedded application.

All of the sofware in the vote casting machine (source and object) should be open for public inspection, even if some of it is proprietary. The goal is that the software should be simple enough that authorities can reasonably certify that it has the required security and correctness properties, and also that anyone else who wishes can study the code as well and satisfy himself or herself that it does what it should and no more (or, if not, then point out problems leading to improvement).

Data format on the frog

We propose that the electronic data format for ballots become a national standard; this standard format would be used to record ballots on frogs.

We give an example of a possible format below. The data is stored in the standard UTF-8 character set, as a set of lines that can be displayed to the user. The format has a header that describes the election location, the precinct, the id of the election official that initialized the ballot, the date of the election, the ballot style, language, and the rotation parameters. The body of the ballot specifies the choices the voter has made. Such a representation is both human-readable and machine-readable. The voter can confirm his choices as they are displayed on the vote-casting equipment without interpretation.

```
State of Massachusetts, Middlesex County, Precinct 11
Ballot Initialized by Election Official 10
Election Closes November 7, 2004 at 8pm EST
Ballot: MA/Middlesex/1; English; No rotation

You have chosen:
     U.S. President: Mary Morris
     U.S. Vice President: Alice Applebee
     Middlesex Dog Catcher: Sam Smith (write-in)
     Proposition 1 (Casino): FOR
     Proposition 2 (Taxes): AGAINST
     Proposition 3 (Swimming Pool): FOR
     Proposition 4 (Road Work): NO VOTE
```

We feel that such standardization of the format of electronic ballots is a very important step for the evolution of voting systems, which should be vigorously pursued, independent of whether other aspects of our proposal are taken up.

Freezing frogs

Election officials should randomly test blank frogs from their supply to ensure that the "freezing" feature actually works. (This is something that can't really be tested on an individual basis.)

Publication and tabulation of the votes

We propose that every vote cast should be made publicly available, together with its associated digital signatures from the witness cards. These could be posted on the web or otherwise be made available. The storage required is small: an election with one million voters, each casting a ballot requiring three hundred bytes to represent, is less than half a gigabyte. (The list of votes would fit on a single CD.)

The tabulation of the votes can then be done simultaneously by any and all interested parties. Any party who has a properly witnessed ballot (signed by enough parties) can double-check that this ballot is on the published list. (For example, the Democratic party storage unit might have received a vote that was somehow dropped from the election official's storage unit.) If not, it can submit this ballot to be included in the list.

If any properly witnessed vote is "irregular" in any way (e.g. a candidates name is misspelled) then the appropriate election official would have to make a decision as to how to count that vote. This is not like examining punch cards for hanging chad! Here the election official is reviewing an electronic document that everyone else also has access to. The rules can be crisp and established beforehand. For example, a ballot may be acceptable even if one of the digital signatures fails to verify. There should essentially no controversy or room for bias to creep in.

Publication of the voter list

We also propose that the list of voters who voted should be made publicly available. Of course, for each precinct the number of names on the list of voters should be equal to the number of votes on the published list of votes for that precinct. There should be no way to link voter names with actual votes of course. The list of names might be published in alphabetical order. The list of votes might be published in random order within precinct, as determined by the storage unit.

Anonymity

How much anonymity is required of a voting system? We propose that *anonymity down to the precinct level* is sufficient. That is, a voter has sufficient anonymity if it is known which precinct he voted in, and if all of the votes for that precinct are published. The voter's ballot is mixed in with several hundred other such ballots, and then published. In general, this should provide sufficient anonymity to prevent coercion and/or vote-buying, which is the major reason for anonymity in the first place.

There may be exceptional situations, as for example when very few people vote in a given precinct. Perhaps in such cases the level of aggregation should be at the county level instead of the precinct level. (Presumably standards of this sort would be determined as a matter of law in each state.)

We note that we do propose that the electronic ballot format include an identification of the election official who initialized the voter's frog. This provides a level of accountability for such election officials; if there are too many votes for one official, then that official could be investigated. The official's identification could be a coded ID rather than a name, so that a voter couldn't try to identify his vote in the published list by looking for the name of the

election official who signed him in. The election official could provide the ID on a coded key that he or she inserts into the frog initialization device when they begin their tour of duty.

Access control

The vote-casting device is the heart of this proposal.

Controlling access to this device is thus controlling access to the voting process. We propose that the pollworkers enforce physical access control to the vote-casting device, just as they would enforce physical access control to an optical scanner for an op-scan voting system. Only voters who have appropriately identified themselves as registered voters would be allowed to use the vote-casting device, and then they could use it once only.

The vote-casting device could even be in an enclosed private space, together with the vote-generation equipment, as long as some mechanism was in place to prevent someone from voting twice (e.g. voting also with a spare frog from his pocket). Perhaps the vote-casting device needs to receive a signal from the election official before it is activated or re-activated.

Printing and Storage Costs

Printing and storage costs are significant with current paper-based systems. Paper ballots have to be individually numbered and printed up ahead of time in several languages in sufficient quantity for the maximum likely number of voters; over-supply is more-or-less required to ensure that each voter will be able to vote.

Similarly, storage of large quantities of paper can be expensive.

Frogs can be purchased blank in bulk; there is no printing cost associated with frogs. Buying frogs is more like buying large quantities of blank paper, which can be done cheaply, rather than buying expensive printed ballot documents. Frogs not used in one election can be used in the next.

Frogs can be stored compactly. They only need to be large enough to be handled.

Recounts

The frozen frogs can be examined if a "recount" is required. But, given the signing process by the witness cards, there should be no difference between the frozen frogs and the electronic stored copies. The most likely need would be when the electronic copy in all of the storage units was corrupted (say by a power spike when that ballot was being transmitted to the storage units). Some specialized equipment for reading a large number of frozen frogs quickly would be desirable.

Provisional ballots

If a poll worker can't confirm that a voter is eligible to vote, then a provisional ballot can be used as for, say, an op-scan system. The voter fills out his frog, but is not allowed to cast it. It is placed in an envelope with his name written on it. If the issue is later resolved in the voter's favor, the voter's frog is removed from its envelope and cast by an election official, with suitable observers.

Absentee ballots

Voters who are absent from their county may be able to prepare frogs and mail them in; they are treated similarly to provisional ballots. This has the obvious benefits and risks; our scheme doesn't affect this tradeoff.

Variations

Other media for frogs

Our proposal can be implemented in a variety of ways. Other media might be used instead of "memory cards". For example, a frog might be implemented on a special kind of floppy disk or tiny rewritable CD. The implementation technology may be able to track the evolution in storage devices in the computer and entertainment industries. It is conceivable that the ideas presented here could even be adapted to "paper-based frogs"; the vote-casting equipment would scan the paper ballot, re-display it, and append signatures using 2D barcodes. (But this may be a stretch…)

Preparing frogs at home

There is no reason why a voter couldn't obtain a blank frog, and fill it in on his home computer. He could then bring it in with him when he goes to the pollsite to vote. The election official could check the ballot style, etc. before allowing the voter to proceed. The voter might be required to insert his frog into the vote-generation equipment at the poll-site, just so he is required to have a private opportunity to change any selections he may have been coerced into making at home. Then the voter could cast his ballot at the vote-casting device as usual.

Blank frogs can be freely bought, sold, mailed, traded, written on, etc. They do NOT have to be carefully controlled, numbered, accounted for, or handled in the presence of no fewer than two officials (as is required for blank paper ballots). Only frogs containing frozen, cast ballots need that treatment, and then only to prevent loss and tampering, since there is no danger of forgery or modification.

Voting over the Internet

One might be tempted to use the proposed standard for the format of electronic ballots in an extension of our scheme wherein the ballots are prepared at home and then transmitted over the Internet to the county election officials. We view this as a *radical* and *undesirable* extension, and would argue that it eliminates all of the security constraints present in having the vote-casting device as presented at the pollsite. (We are also against absentee voting and voting by mail except in cases of demonstrated need.)