



c/o 3548 Beechwood Boulevard  
Pittsburgh, Pennsylvania 15217-2767

## **iVotronic Software Verification Protocol: Allegheny County Proposals.**

Collin Lynch, President, VoteAllegheny  
9/28/2008

The iVotronic is a software-dependent system. Absent a machine-independent record such as a voter-verified paper ballot, the correctness of the vote rests entirely on the correctness of the hardware and software used. If errors exist in either aspect or the system is compromised through the introduction of uncertified components then the vote cannot be trusted.

Our faith in the security and accuracy of the system rests on the premise that the ITAs (Independent Testing Authorities) and State inspections will identify all security problems and other systemic errors, and that we will employ only the certified hardware and certified software, and do so only in an approved manner.

Studies conducted on the iVotronic system by researchers from the University of Pennsylvania [1] as well as others in the States of Ohio [2] and Florida [3] have demonstrated existing vulnerabilities in the system.

Existing tests employed by the county including version number examination and parallel testing may be circumvented by malicious software. Any virus-infected PC still makes the same start-up chime and any compromised voting machine can still print "9.1.4.1." Similarly, a skilled virus can detect a predefined pattern of voting, be signaled by inside actors, or use a variety of other methods to distinguish testing from a real election.

The purpose of public software verification is to verify, inasmuch as it is possible, that the software we employ is the certified software. This requires an open public examination process covering a statistically significant number of systems selected at random from the population. If any of these conditions is not met then the process itself is meaningless. The role of public participation and examination is to prevent corruption of the process by inside actors whether intentional or unintentional.

The best possible analogy to the process is that of disease detection in public health. Our goal is to determine whether or not any member of the population is infected. Being unable to examine the full population of more than 4,000 iVotronics as well as attendant PEBs (Personal Electronic Ballots) we must examine a sample of the population. In order to ensure that the conclusions we draw will apply to the full population without bias the selection must be random. And as the goal is security of the public's vote the process itself must be public to avoid the potential for compromise by motivated insiders.

## **(1) Software Verification Overview – Goals and Practicalities**

In an ideal design, Software Verification is a two-part process. Prior to an election, after the iVotronics have been prepped and all ballot programming is complete, a statistically significant number of districts are randomly selected through a public process.

Once done each iVotronic in the district is opened up and the internal memory chips removed. These are then inserted into an EEPROM reader connected to a clean system. The contents of the chip are then compared to a known clean copy of the certified system as escrowed by the State and NIST (National Institute of Standards and Technology). The same process is then carried out for each PEB in the selected polling place.

It is necessary to conduct the verification on a district-by-district basis as the systems within the district interact in the course of an election through the exchange of PEBs during setup, voting and closing of polls. Any single infected or altered system could potentially pass a virus to other systems or compromise the results during total accumulation.

If the process is conducted, and if all sampled districts are found to be clean then we can say, with some confidence, that the remainder of the machines are clean. If any one is found to be compromised then all remaining systems must be examined along with steps taken to determine the source of the compromise. Optimally, this process would be repeated once the election is over with a different random sample. This remains a potential for future elections.

As noted above, public participation is necessary at every stage of this process to control the effect of malicious or negligent inside actors. The manipulated recount in Cuyahoga County, Ohio, demonstrates the risks of inside actors. Absent public participation, both in the design of the process and its execution, it will be as vulnerable to compromise as the existing closed-door testing. Public participation is needed to protect public elections.

Unfortunately under present circumstances we cannot meet this ideal design in its entirety. First and foremost, we lack a reliable software-independent method to dump the contents of the PEBs. While work is being undertaken to identify one, such a process will not be ready before the November 2008 election. Secondly, it may not be feasible, under the present time scale, to verify the systems by district. Therefore we need to carry out the above process sampling randomly from the full set of iVotronics as the population. Randomly selecting machines from the population is necessary to prevent any bias in the selection from making the results useless. Care must be taken to treat all the machines identically after the study so as to keep the answer consistent. In order to make a claim about all machines it is necessary to ensure that the selection was made evenly from the full set.

## **(2) System Selection**

Strictly speaking, in order to have 100% confidence that all of the systems are clean we would have to inspect all 4,000+ iVotronics, PEBs, scanners, and tabulators. For the present election that is clearly infeasible. As in disease detection, it is necessary to analyze a sufficient number of systems to ensure that the results can be generalized with a guaranteed level of confidence to the remaining population, that is the rest of the systems in the county. The goal is to determine statistically whether we can state with a given level of confidence that the number of infected systems in the population is below an acceptable threshold. To do this, we need to define, as a county, the following parameters:

*Confidence Level:* The desired level of confidence we wish to have in our outcome. A level of 95-99% is standard for most empirical research.

*Acceptable Threshold:* The maximum percentage of systems that we will permit being infected. This threshold is discussed in more detail below.

Having settled these two parameters we can then use a standard formula for calculating the minimum sample size. This formula will tell us, for a given level of desired confidence and a population size what the required number of systems is.

### **Acceptable Threshold.**

As noted below, if even a single system is compromised then the integrity of an election is in doubt. However, as noted above, it is infeasible to test all iVotronics prior to this election. With that in mind it is necessary to determine what an acceptable level of infected systems is.

This can be viewed as a gaming case. If a system is compromised we must assume that it will be compromised with the purpose of throwing votes to one side or another in a given election. This can be done by changing votes or by simply shutting down the machine entirely. What is then necessary is to determine how many machines can go down entirely or move votes to one side or the other without affecting a race.

The best mechanism to set this is to define a minimal margin of confidence, that is, what is the narrowest race we can have confidence in? This may be done by analyzing historical races and considering what the typical or average margin of victory is. We then determine how many votes it takes to make up that margin and then permit less than half of them (preferably one third or less) to be false. From this we get the acceptable margin of compromised votes.

Having done that, we determine the average number of votes per machine in those same historical elections. Once that is done we then divide the number of votes by the number of votes per machine to return the number of machines that may be compromised without affecting the acceptable margin.

It is important, when doing this, that we not set this margin based upon "abnormal" races with routinely high margins (*e.g.*, infrequent races such as County Executive) or other races that are affected by a party-line split as those race margins are unlikely to be representative of the whole. It is also important that the margin not be selected based upon the present race for post-election verification as that margin remains suspect until proven otherwise.

Given historic margins in both local and national races, especially closely fought presidential, congressional, and council elections, we would not recommend a threshold greater than 1%.

**Sample Statistics**

In order to compute an optimal statistical breakdown it is necessary to determine the complete number of machines in the county's inventory as well as the acceptable level. The table below illustrates a range of values for 99% confidence depending upon the acceptable margin of compromised systems. For each margin level it also indicates the number of potential systems each of which may store up to 350 votes under state law per election. These numbers were calculated assuming a county population of 4,000 systems.

# Systems Tested	Acceptable Margin	# Systems Compromised	Est. # Compromised Votes
3,000	00.1%	4	1,400
300	01%	40	14,000
99	03%	120	42,000
30	10%	400	140,000

In other words, if we verify the software in 30 machines out of 4,000, we can be 99% confident that no more than 400 machines out of the 4,000 have been compromised.

**Recommendations**

As we stated above, given historically close races we recommend that the county adopt a threshold no greater than 1%. This margin will also aid in guaranteeing the safety of subcounty races such as council and judiciary positions. Thus, as per the table above we recommend that the county examine at least 300 iVotronics randomly selected from the population as a whole.

**(3) iVotronic Verification**

As described above, the individual system examination is an extraction process. For the purposes of this extraction a clean PC, an EEPROM reader, and a secure hash of the certified system software are required. While the PC and reader are hardware that may be purchased by the county the hash must be computed from the clean copy of iVotronic firmware, version 9.1.4.1, which must come from the escrowed copy at the EAC (Election Assistance Commission).

**Software Hash**

A hash function is a mathematical function that takes a large number, say the number representing all the bits in the binary code for a program, and generates a much smaller number with a special property. While the number is unique it is practically intractable to guarantee that an arbitrary number maps to the same value. For the purposes of software verification computer code may be treated as a single large number and passed through a hash function to obtain a fingerprint.

Thus, given a hash function such as SHA1 it is possible to generate a “clean” fingerprint representing the escrowed valid program code. Given a suitable hash function this produces a relative guarantee that no attacker can produce arbitrary malicious code that the hash function will map to the same fingerprint.

As with other security matters the appropriate choice of hash function is dependent upon current security practice. While MD5 was previously an industry standard it has now been weakened and retired from use. We therefore recommend the use of SHA1 or SHA256.

### **Clean PC**

To produce a clean PC on which to conduct the tests, burn a bootable CD ROM or DVD ROM containing: the system software necessary to run the PC; drivers for the EEPROM reader; software to compute the hash; and the “clean” hash fingerprint of the escrowed program code. This disc is then used to boot a diskless, and networkless PC to run all tests. The read-only nature of the media reduces the prospect of corruption during the tests and enables tighter control of the hash values. Construction of a bootable CD ROM using Knoppix or other open-source projects is feasible and well within the scope of existing county equipment.

### **Verification Procedure**

The iVotronics are equipped with three internal EEPROM memory chips, one of which may be removed. In theory each of these chips contains an identical copy of the memory and the system is designed to fail if this is not the case. For the present purposes we will operate on the principle that they are identical.

For each iVotronic in a selected district the system must be opened up in front of witnesses and the EEPROM chip removed. It is necessary to extract the software separately from the system as any software-dependent extraction (as is done on Diebold systems) may be compromised. A virus, sitting on a compromised system when asked to dump the memory contents may simply elect to dump a "clean" copy thus avoiding detection. Compromises of this form have been discussed previously [4].

Once removed, the chip may then be inserted into the EEPROM reader, which is then used to extract its full contents to the clean PC for reading. Once read, a hash fingerprint will be computed from the chip contents and compared to the clean-system hash fingerprint. If the two differ then the target system is running uncertified software. If they are identical then the target system is, for the purposes of this test, clean at this point in time.

### **(5) Remediation.**

If an infected system is found, it will then be necessary to expand in scope to cover the remainder of the population. Continuing with the analogy of a disease, if a single system is infected then it is likely that others are as well. This may be due to the spread of a virus from PEB to iVotronic and back, or an insider with access to multiple systems. In either case the location of a single compromised system requires a reexamination of all remaining systems in the population. Even if the compromised software appears to be benign the fact of its presence renders the system illegal to use. Prior to election day it may be possible to reset or repair infected systems before deploying them into the field.

## **(6) Caveats.**

This process was designed with the iVotronic systems in mind. For a complete verification process a similar procedure must be conducted on the M650 scanners employed by the county as well as the central tabulation systems, and the flash-chip programming units, all of which may also alter the outcome of an election. While the scanners may be audited by means of a manual recount of the ballots this is not possible for the central tabulators.

Additionally, the sample calculations above deal with county-wide races, and the margins for each. Smaller races such as Judges of Elections or even individual County Council seats do not cover the entire county. Therefore a smaller number of corrupted systems is necessary to compromise those races. In order to set an acceptable margin that encompasses those races it would be necessary to reproduce the margins using the number of machines for that race as a guide. Thus, to prevent fraud within a given council district or township it would be necessary to determine the margin of votes necessary to alter that race and then sample to that higher standard.

This process is not a complete security solution, merely a necessary addition to the present tool set. Proper software verification will reduce the class of software attacks that predate election day. However, it will not prevent or even detect software replacement that occurs on and is reversed within an election day. Nor will it detect the use of incorrect hardware or systems compromised by the replacement of individual chips. Apart from visible differences it is not practical to perform arbitrary hardware verification as two chips may appear identical but perform radically different functions. Genuine verification of hardware components requires expensive tools such as scanning electron microscopes and/or JTAG hardware probes. This must be performed manually for each component. Therefore, in the absence of visible differences, the detection of uncertified hardware remains an impracticable problem.

That having been said, software verification would represent a vast improvement on the existing state of affairs.

## **References**

- [1] Security Evaluation of ES&S Voting Machines and Election Management System  
Adam Aviv, Pavol Černý, Sandy Clark, Eric Cronin, Gaurav Shah, Micah Sherr, and Matt Blaze,  
[http://www.usenix.org/events/evt08/tech/full\\_papers/aviv/aviv.pdf](http://www.usenix.org/events/evt08/tech/full_papers/aviv/aviv.pdf)
- [2] Ohio EVEREST Testing reports:  
<http://www.sos.state.oh.us/SOS/elections/voterInformation/equipment/VotingSystemReviewFindings/EVERESTtestingReports.aspx>
- [3] Sarasota Florida Study:  
<http://www.cs.berkeley.edu/~daw/papers/sarasota07.pdf>
- [4] Security Analysis of the Diebold AccuVote-TS Voting Machine  
Ariel J. Feldman, J. Alex Halderman, and Edward W. Felten  
<http://itpolicy.princeton.edu/voting/>