

ES&S Programming Is Unverifiable

By John Washburn

The Iowa Secretary of State has responded to my reporting on ES&S firmware and has confirmed that ES&S programming is unverifiable. The original article appeared on my blog[1]. This article was picked up by an election integrity activist, Jerry Depew of Iowa Voters [2]. Mr. Depew in turn asked his state election officials to respond to my allegation. The response from the Casey Sinnwell, Assistant Director of Communications to Iowa Secretary of State, Chet Culver, confirms my statement.

It is **impossible** for an election official (either at the state or local level) to verify the software running on an M100 scanner for an election is indeed the version of the firmware certified for use in the state. The inescapable conclusion for Wisconsin is that it is also **impossible** for a municipal or county clerk to verify whether or not the statutory requirement of WI 5.40(2)[3] is met when the programming is delivered by ES&S for any precinct, municipality, or county.

Here is the complete response from the Secretary of State of the State of Iowa:

I thank you for your concern with our voting process and procedures. It is important to continue this dialogue with the public to assure the validity and security of our voting process here in Iowa.

Does ES&S routinely use uncertified software to run elections?

The software by ES&S was certified but, as installed, includes embedded data for the unique election. The problem reported was that you could not verify the software by a simple comparison between the memory chips from a master or between counties in the same election because of the election unique information in each county. To verify the software requires using a master copy and a copy of the election definition, then creating a reference copy of the specific election program to compare against the version used in the election. The technique is slightly more difficult because some EEPROM burner/readers create or fill blank areas with different characters and the sections need to be checked to verify that they are true fills. However, the technique has been used in Florida and other locations and the version of the software verified.

The method has been successfully used to confirm both a certified version and to discover that the wrong version was used in system testing. Moreover, the problem you are referring to does not exist as you see it. The problem is that the ES&S system can not be verified with a simple comparison, due to the election program file is different between counties but, the program itself is still certified and needs to be verified using a different technique rather than just a simple file comparison. I hope this answers your questions and concerns about the integrity of our voting systems here in Iowa. While concerns continually come to mind and improvements are considered, we assure you that Iowa Secretary of State's office in accordance with Iowa Code has taken all necessary measures to ensure the integrity and security of our elections.

*Sincerely,
Casey Sinnwell
Assistant Director of Communications
Iowa Secretary of State, Chet Culver
105 Statehouse
Des Moines, 50319*

However, Casey Sinnwell's explanation, itself, shows the inaccuracy of his claim that ES&S programming can be verified.

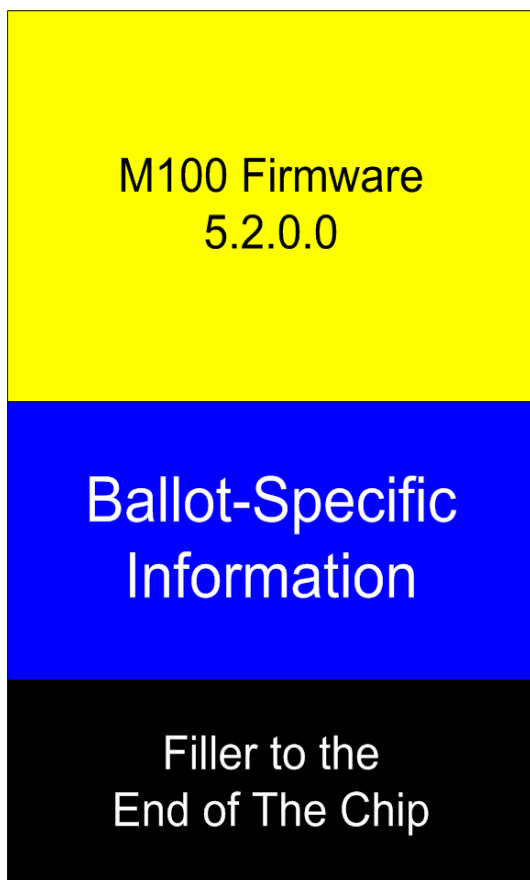
He said that a "different technique" could be used to verify that the software used in the election was the same version as certified software. However, this technique was **NOT** used successfully in Florida. Paul Craft abandoned the effort and confirmed that the process fails to verify compliance. Here is a quote from an email between Paul Craft and the State of California which both describes the attempt at verification and summarizes its failure.

*I was able to validate that part of the system although it did not give me the level of certainty that I really wanted to support a **positive finding** of compliance. [4] [emphasis mine]*

The level of certainty he wanted can be achieved in only one of two ways: 1) using hash values from a cryptographic hash algorithm, or 2) a byte-for-byte comparisons between binary images. Neither approach works for the software components found on ES&S scanners.

Here is a block diagram of the memory layout of ES&S programming if the programming were to conform to requirements section 8.7.1 Volume I, section 9.6.2.4 Volume I, and Appendix B.3 Volume II of the 2002 Voluntary Voting System Guidelines. [5 6 7]

But, as Mr. Casey and Mr. Craft point out the actual arrangement found for the programming of an ES&S system is this. The ballot-specific and machine-specific portions of the programming are inseparable and co-mingled.



Mr. Casey's explanation confirms that it is impossible for a state inspector, such as Dr. Shamos of Pennsylvania or Kristofer Frederick of Wisconsin, to determine whether or not the firmware tested and certified by them is the same software qualified by NASED and tested by the vendor-funded ITA lab. This is **not** because of the potential problem with the memory fill at the end of the memory chip (black portions). The problem is easily avoided during any identification process, despite Mr. Casey's erroneous assertion to the contrary.

The software is unverifiable because it is not possible to isolate and identify only the yellow portions of the ES&S programming. The yellow portion of the programming is what is claimed to be state-certified or is claimed to be NASED-qualified.

The version of software delivered for use in an election cannot be verified by the vendor-funded ITA labs, state certification officials, or local election officials, since only green programming is available for verification as the system undergoing functional testing or is used in an election. The green programming inevitably varies from election to election, so both a hash-value comparison and a byte-for-byte comparison will **ALWAYS** show differences between the system under test and the certified system.

The following discussion assumes the use of hash values, but the same concept is true for a byte-for-byte comparison of binary images. Also the memory discussed below is the memory on the memory chips found on ES&S equipment. It is not the memory found on the electronic ballot boxes of a removable PCMCIA memory card. It is not the memory on the ISO-7816 smart card used as a voter access card. It is the programming on the memory chips of the ES&S equipment.

Think of the hash value as a numerical value of a precise color. The hash value of the programming in the yellow areas of memory would give you a number that is a precise shade of yellow. If the shades of yellow for two systems are identical to each other, then the memory contents in the yellow areas of the memory chip are also the same. If the shades of yellow for two systems are the different, then the memory contents in the yellow areas of the memory chip are also different. Similarly, hash values of the blue areas and the green areas give you numbers for the precise shades of blue and green, respectively.

Just like colors, the hash value of a program is a single value; it cannot be separated into different values that apply to different parts of the program. Knowing the exact numerical color value of green created by mixing blue and yellow paint tells you nothing about the color values of the yellow and blue which were mixed to form the shade of green.

Similarly, knowing the hash value of a program created by combining yellow and blue software tells you nothing about the hash value of the yellow or the hash value of the blue, since the yellow and blue no longer exist as separate programs. They have been combined into green software, just like when you mix paint.

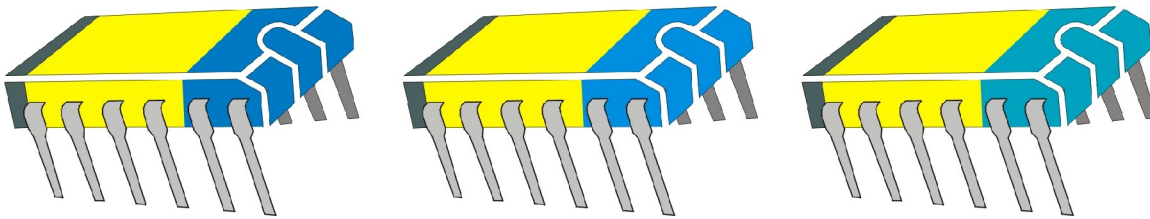
So let's explore the legal peril this unverifiable programming presents to a local election official. Since the confirmation of my allegations came from Iowa, I will use an Iowa illustration. Pottawattamie County, Iowa has 47 reporting units; 46 physical precincts and the county-wide tabulation of absentee and special ballots.

If every part of the programming for the tabulators is correct and error free, the county auditor expects 47 sets of programming in 47 distinct shades of green. This is because the yellow portions (the certified firmware portions) are correct and unchanging and the error-free, and the ballot-specific information is represented by 47 different shades of blue. Mixing 1 shade of yellow with 47 different shades of blue creates 47 different shades of green.

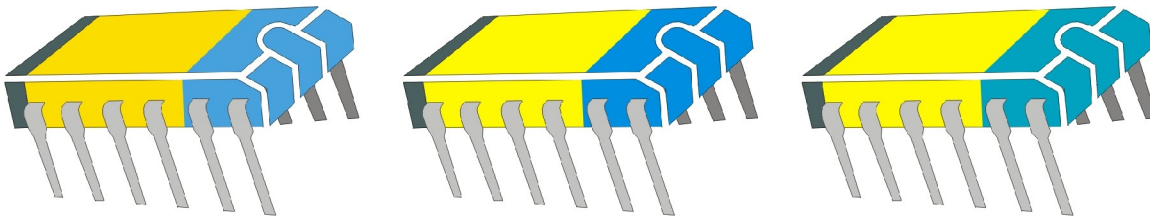
Unfortunately, with the ES&S model, the auditor gets 47 different shades of green whether all the programming is correct, or the firmware is uncertified in some sets, or some of the ballot-specific information is incorrect, or all of the ballot-specific information is incorrect. This is because mixing 47 shades of blue with any of several shades of yellow still creates 47 shades of green.

So no matter what the state of the programming (correct or incorrect ballot programming; certified or uncertified machine-specific programming), the evidence available to the County Auditor is always the same; 47 sets of programming in 47 shades of green and no way to determine the exact shades of yellow and the exact shades of blue which mixed to create the shades of green delivered to the 47 precincts.

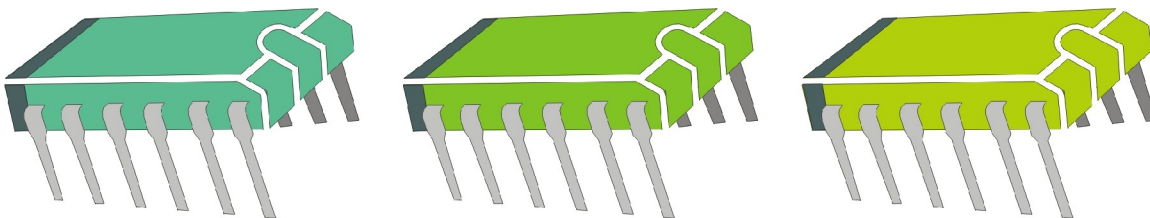
Instead of getting chips like this with certified firmware and different, but correct ballot-definition information:



Or instead of getting this set of chips with uncertified, but **detectable** firmware on one chip:



The auditor gets chips in many shades of green such as this:



The auditor, tester, or clerk has no way to know if the variation in shades of green are from error-free differences in the ballot definitions, errors in the ballot definitions, errors in the machine-specific firmware, or some combination of the three. The result is always the same: different shades of green. **No information** on either the ballot definitions or machine-specific firmware is available.

The programming is thus unverifiable. A County Auditor in Iowa or municipal clerk in Wisconsin cannot verify which version of software was delivered to them. Thus a County Auditor in Iowa or municipal clerk in Wisconsin has no way of confirming or denying that the software delivered to them is actually state-certified or not.

End Notes

¹ <http://washburnworld.blogspot.com/2006/05/ess-products-do-not-conform-to-2002.html>

² <http://iowavoters.org/>

³ <http://tinyurl.com/jjgw>

⁴ <http://www.washburnresearch.org/archive/FCMGroup/CraftFreeman01.pdf>

⁵ http://www.eac.gov/election_resources/v1/v1s8.doc

⁶ http://www.eac.gov/election_resources/v1/v1s9.doc

⁷ http://www.eac.gov/election_resources/v2/v2ab.doc