Secure Data Export and Auditing using Data Diodes

Douglas W. Jones and Tom C. Bowersox

Department of Computer Science
The University of Iowa
Iowa City, IA 52242
jones@cs.uiowa.edu, tom-bowersox@uiowa.edu

Abstract

Data diodes, that is, devices which permit only one-way communication, without even a reverse channel for acknowledgment, have many potential uses within voting systems. A practical design for a data diode is presented that is simpler and more nearly self-evident than previously published designs. Communication protocols appropriate for use in the voting context are described. Throughout, we emphasize designs that permit a relatively naïve observer to determine that it meets key security constraints.

The Problem

The results of any election must be published, for example, in newspapers or on the Internet. This generally requires communication from the election management system used for canvassing the election to any of a variety of systems outside the election administration security domain. At the same time, the election management system must be protected against intrusion from the outside.

Many election officials deny that their systems are vulnerable to attack, flatly stating that their election management system has no public network connections.[1] This denial cannot be taken at face value if the election management system provides up-to-date election results on a public server.

In one system we examined, we have found a remarkably baroque data export path best characterized as security through extreme obscurity.[2] In other cases, vendors recommend using an air gap with "sneakernet technology" to carry data across this gap[3] Just because the electronic media are hand-carried across an air-gap does not imply that there is no reverse channel! One can easily imagine hand-carrying data back and forth in a thumb drive or any other reusable medium in such a way that contagion is carried into the election management system with each shuttle across the air gap.

Data export on write-once disposable media such as ink on paper or recordable CD-ROMs is safe, but this may be just cumbersome enough that many jurisdictions will cheat. This moved us to develop an easily audited one-way on-line data connection for use between election servers and systems connected to public data networks.

The basic idea we exploited was an electro-optical data-diode. This is not a new idea; it has even been patented.[4] The patent explicitly limits itself to transmission from an unsecured computer to a secured computer, exactly the opposite direction from the data transfers that concern us here. In every case we discuss, data is exported from a secure environment to an insecure environment containing potential threats.

Another class of similar devices are data pumps.[5] Practical implementations of data pumps include multiple microprocessors and buffer memory. As such, the total complexity of the pumps now on the market is such that convincing a naïve observer of their trustworthiness would be quite difficult.

We reject data pumps not only because they are very complex, but because they maintain a very low bandwidth reverse channel for handshaking. In the context of elections, we cannot afford a reverse channel that permits even a single bit to be transmitted in the wrong direction. Consider the risk posed by a reverse channel communicating the message "nobody is watching so it is safe to cheat now."

The connection between an election management system and insecure public data networks must be auditable. That is, outsiders with limited technical knowledge must be able to inspect the connection and easily verify that data can only be transmitted in one direction and not in another. This led us to focus on devices of minimal complexity, so that a person with only the most rudimentary knowledge of electronics can easily verify that the device cannot be used to transmit data in the wrong direction.

A Prototype Design

We rejected use of USB, Firewire, Ethernet, and other high performance input/output ports because all of them require handshaking at low levels in the protocol stack in order to support one-way communication at high levels. We rejected fiber-optic devices because it is hard to tell the direction of data transfer without disassembling the device. The only widely available communications interface that meets our simplicity constraint is the RS-232 asynchronous serial data port.

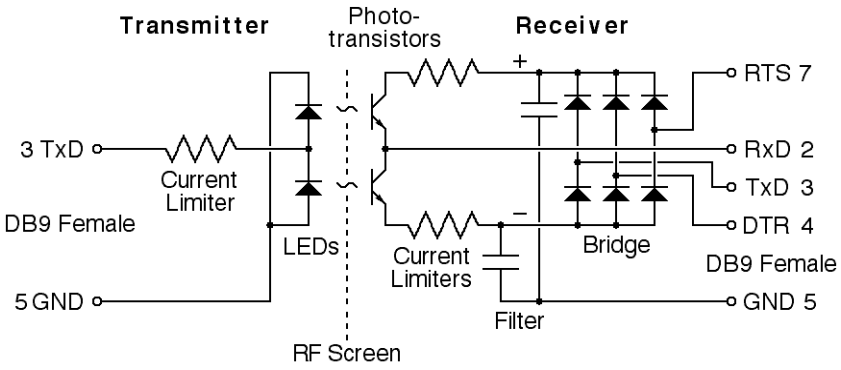

Figure 1: Schematic of the data diode.

Our device does not require the use of any integrated circuits and contains fewer than 20 discrete components; this is far simpler than the data diode discussed in the patent previously cited. Current commercial realizations of that patent are even more complex, being based on high-performance fiber-optic technology.[6] [7]

Our prototype uses a pair of red LEDs for differential data transmission. It fits comfortably in an inexpensive mint tin; with the tin open, the enclosed circuit board can be removed and inspected from all sides to see that none but the required components are present. All circuit board traces are on the component side, with a layout that makes it easy to compare the board, as fabricated, with the documentation. The documentation is written in a tutorial style so that someone with minimal knowledge of electronics can audit the design and construction of the device.

Our original intention was to use cadmium-sulfide photocells in the receiver because these do not resemble any light-emitting devices. Unfortunately, the cadmium-sulfide photocells we tested were all very slow, preventing operation at speeds above a few baud, and they had high resistance even under full illumination, requiring use of transistor amplification. The phototransistors we ended up using have packages identical to those used for LED's, somewhat reducing auditability, but they have high enough gain that a student who understands little more than Ohms law can figure out how the circuit shown in *Figure 1* works.

Most current RS-232 interfaces appear to use variants of the MAX 232 chip.[8] The PC serial ports all had output short-circuit currents near 20mA on data and control pins, and open-circuit output voltage near 11V. Our data diode operates from this power. The current limiters are sized to operate at up to 15V and the device should function at 7V. This is not the full 5V to 25V open-circuit voltage range permitted by the RS-232 standard.

Our circuit may be broken in two so that the transmitter and receiver can be mounted on opposite sides of a window. It is very common to have such a window between the secure vote processing area and the area reserved for election observers. We have not designed packaging for this or other variant configurations.

In testing our prototype, we found that the addition of rectifier diodes in series with the LEDs was an effective way to limit the data rate of the device to 2400 baud. Without these, it functions reliably at 9600 baud under indoor lighting.

Software Considerations

Because our device has no reverse channel, the transmitting system cannot determine if any data is being received or if that data was received correctly. While this restriction would be entirely unacceptable in many applications, it poses few problems in the context of an election management system. Here, we can safely rely on forward error correction, and we can declare the entire system to be operational only if the system administrators can observe that data is indeed being received on the insecure side of the diode.

Auditability is greatly complicated by deep protocol stacks. We therefore propose an extremely simple transmission protocol designed specifically to allow naïve users to observe the data stream and easily determine that it contains only the data required and nothing else. Variant hardware using a second receiver to monitor the LEDs is clearly within reason. Such hardware would permit observers to attach wiretaps that directly monitor the data stream without the ability to interfere with it. This allows election observers to monitor the data being published by the election management system, without having to rely on the possibly insecure web server that provides the official public portal to the results.

We propose that election results be released as a continuous repeating stream, in much the same way that stock market data was traditionally distributed. This guarantees that the receiver will eventually receive all of the results, even if some results are dropped or corrupted in transit.

```
<ITEM PRECINCT="123" CANDIDATE="Jones" VOTES="12" />32669
<ITEM PRECINCT="123" CANDIDATE="Bowersox" VOTES="16" />15819
```

Figure 2: A reasonable format for data export.

For the purpose of auditability, all transmission must be in the clear. Others have insisted that all data transmission be encrypted.[9] We permit use of complex algorithms only for error detection and message authentication, and even there, we prefer algorithms that a novice programmer can understand. While a simple additive checksum may be insufficient, commonly used error detection algorithms such as CRC-16 are too complex to explain to a typical undergraduate.

The example data presented in *Figure 2* includes the checksum over the ASCII characters of each record appended immediately after that record. We have used a checksum scheme of intermediate complexity that we believe is both sufficiently naïve to meet our auditability constraints and sufficiently robust for useful error detection. To compute the checksum c_n of a message m, where m consists of characters m_1 through m_n, encoded in ASCII, we use c_0=0 and c_i=$(5c_{i-1}+m_i)$ mod 2^{16}. Other initial values, multipliers and moduli may be used, so long as the modulus and multiplier are relatively prime.

It would be interesting to find a message authentication code that could be computed without exceeding this level of complexity. In support of this possibility, we note that our checksum algorithm bears a close resemblance to linear congruential pseudorandom number generators.[10] We speculate that the values of c_0, the multiplier and the modulus could serve as a key for a useful message authentication code based on an algorithm of this sort.

Our requirement that the content of the data be self evident drives us very quickly toward something that resembles XML.[11] W have opted, therefore, to make each record in the data stream into a well-formed XML element. The context of an infinite data stream precludes full XML compliance because there can be no prologue. We have opted to encode each record as an empty-element tag, encoding the data in that item as XML attributes. The alternative, where data fields are encoded as element content, tends to produce larger and therefore less readable representations of the data.

We report one record per line of output, with the line end character immediately following the checksum. In order to allow resynchronization after a transmission error, we require that the communications line go dead for a few character times after each record.

We recognize that the Election Markup Language proposed by the Oasis Election and Voter Services Technical Committee may be relevant in this context.[12] The recommended method of including digital signatures in XML documents may also be relevant.[13] Unfortunately, the examples we have seen that demonstrate these are sufficiently verbose and cumbersome that we believe that they interfere with our basic auditability requirement.

Covert Channels

It is important to note that both asynchronous communication and textual data formats such as XML offer significant possibilities for covert channels. A corrupt election management system could attempt to covertly export the cryptographic keys necessary to forge the message authentication codes it uses. If successful, this could allow an attacker to forge election results. We must therefore block any covert channels that might exist, so that auditors can easily tell that only the authorized data is being exported from the election management system.

Covert data may be included in asynchronous data by modulation of the inter-character delays. The interfaces provided by most operating systems make both the control and measurement of these delays difficult, so we consider this channel to be a mild threat. We recommend, however, that source code audits of voting systems take note of any attempt to introduce non-constant time delays in their output.

XML allows very easy inclusion of covert data within the data stream.[14] We therefore require the use of a canonical form, although we are find the current W3C recommendation to be both overly verbose, for example in the way it forces use of end tags, and insufficient, for example in the way it fails to regulate use of white space outside of tags.[15] Manual auditing for compliance with a canonical form is straightforward if our data format is sufficiently constrained, and it is easily automated.

Other Possible Applications

There are other possible uses for data diodes in the election domain. Many of these can use the same device and the same data format we have proposed. For example, when uploading data from precinct voting equipment to the election server, possibly by way of satellite vote collection centers, there is a threat that an adversary might inject some kind of attack into the precinct voting system via the communications line. In the extreme case, a corrupt election administration might use the election management system itself to attack the equipment at the precinct. The risk of such attack can be reduced if a one-way data diode is inserted between the precinct voting equipment and the modem used for reporting the results, as suggested in *Figure 3*.

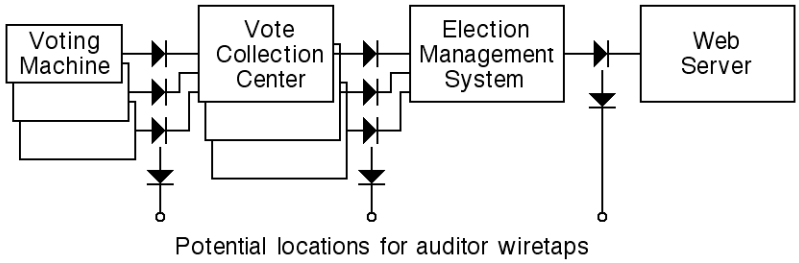

Figure 3: Possible points for insertion of data-diodes in an election system.

Just as public confidence can be increased by allowing election observers to wiretap the communication between the election management system and the web server used to publish the results, wiretaps on the inputs to the election management system may also have value. Once the voting machines have gathered the precinct election totals, they are routinely printed and posted at the precinct. Therefore, disclosing the electronic transmission of the exact same data to auditors poses no problems so long as the auditors can only listen to the transmission and not interfere with it.

These additional one-way connections are applicable only to the uploading of data from the voting system after the polls close. During the election, it is best if there be no communication, while before the election, we need to download ballot configuration data into the voting machines. The data paths used for this are outside the domain of this paper.

Finally, it is interesting to consider the application of data diodes within the precinct voting system itself. The time-of-day clock that is required in order to time-stamp event-log records maintained by the voting system can also be used to enable time-dependent attacks.[16] Clearly, if we can cut off access to the time-of-day clock outside of the auditing function, we can limit the opportunity to carry out such an attack. *Figure 4* illustrates how a data diode can be used to accomplish this.

Inserting a data-diode as suggested here prevents the vote collection mechanism from being able to determine that the system is operational. As with our other proposed uses of data diodes, only the downstream subsystem can determine whether the system is operational. Here, we solve this problem by attaching a ready light to the event-log mechanism. If this light is off, users would not be permitted to use the system.

Recent disclosures of multiple attack paths from removable memory cards into a precinct voting system[17] suggest placing another data diode between the vote collection mechanism and the subsystem that records data on removable media.

Clearly, data diodes used within a voting system need not be based on the RS-232 standard. Simple optical couplers should suffice. As with our RS-232 data diode, however, simple board layout that allows an auditor to visually determine that there is no hidden channels are essential.

In this case, the auditor being addressed by the use of the data diode may not be a suspicious election observer, but rather an official at the independent testing authority that certifies the voting system. What we gain by careful insertion of data diodes between system components is a reduction in the size of the critical parts of the voting system software that must be subject to close inspection.

Acknowledgement

This work was partially supported by NSF Grant CNS-05243 (ACCURATE).

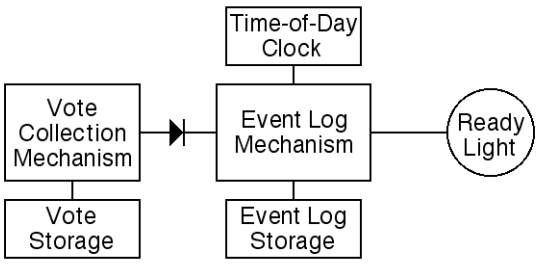

Figure 4: A data diode within a voting machine

1 *Maryland's Better Way to Vote – Electronic Voting, Myths vs. Fact*, Maryland State Board of Elections, 2004; available from http://www.elections.state.md.us/citizens/voting_systems/mythvsfact.pdf. See Myth 5, bullet 3.

2 Douglas W. Jones, *Observations and Recommendations on Pre-election Testing in Miami-Dade County, 2004*; available from http://www.cs.uiowa.edu/~jones/voting/miamitest.pdf. Section 7.

3 *Election Support Guide*, Diebold Election Systems, Revision 1.0, 2002; available from http://www.meer.net/~fairelection/documents/doc/DieboldEsg20040811.pdf. Section 13, Item 1.

4 Curt A. Nilsen, *Method for Transferring Data from an Unsecured Computer to a Secured Computer*, U.S. Patent 5,703,562, Dec. 30, 1997.

5 Myong H. Kang, Ira S. Moskowitz and Stanley Chincheck, The Pump: A Decade of Covert Fun, *Proc. 21st Annual Computer Security Applications Conference*, Tuscon, December 2005, available from http://www.acsac.org/2005/papers/Kang.pdf.

6 Owl Computing Technologies Inc, http://www.owlcti.com/.

7 Tenix Datagate Inc, http://www.tenix.com/.

8 *MAXIM +5V-Powered, Multichannel RS-232 Drivers-Receivers, 19-4323; Rev 14*; 8/04, Maxim Integrated Products, Sunnyvale; available from http://pdfserv.maxim-ic.com/en/ds/MAX220-MAX249.pdf.

9 *Direct Recording Electronic (DRE) Technical Security Assessment Report*, Compuware Corp., November 21, 2003; available from http://www.sos.state.oh.us/sos/hava/compuware112103.pdf. See Requirement 1.27.

10 S. K. Park and K. W. Miller, Random number generators: Good ones are hard to find, *Communications of the ACM, 31*(10), pages1192--1201, October 1988, available from http://portal.acm.org/citation.cfm?id=63042.

11 *Extensible Markup Language (XML) 1.0, 3rd Ed.*, W3C Recommendation 04 February 2004; available from http://www.w3.org/TR/2004/REC-xml-20040204/.

12 Oasis Election and Voter Services Technical Committee, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=election.

13 *XML-Signature Syntax and Processing*, W3C Recommendation 12 February 2002; available from http://www.w3.org/TR/xmldsig-core/.

14 Shingo Inoue, Kyoko Makino, Ichiro Murase, et al, A Proposal on Information Hiding Methods using XML, *Proc. of the 1st NLP and XML Workshop*, Tokyo, November 30, 2001, pages 55-62; available from http://www.afnlp.org/nlprs2001/WS-NLPXML/.

15 *Canonical XML Version 1.0*, W3C Recommendation 15 March 2001; available from http://www.w3.org/TR/xml-c14n.

16 Douglas W. Jones, *E-Voting – Prospects and Problems*, Tau Beta Pi 31st Annual Paul D. Scholz Symposium, April 13, 2000, University of Iowa; available from http://www.cs.uiowa.edu/~jones/voting/taubate.html.

17 Harri Hursti, *Diebold TSx Evaluation*, Security Alert, May 11, 2006, Black Box Voting, Inc.; available from http://www.blackboxvoting.org/BBVtsxstudy.pdf