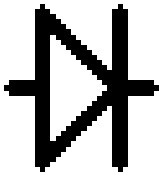
Secure Data Export and Auditing using Data Diodes



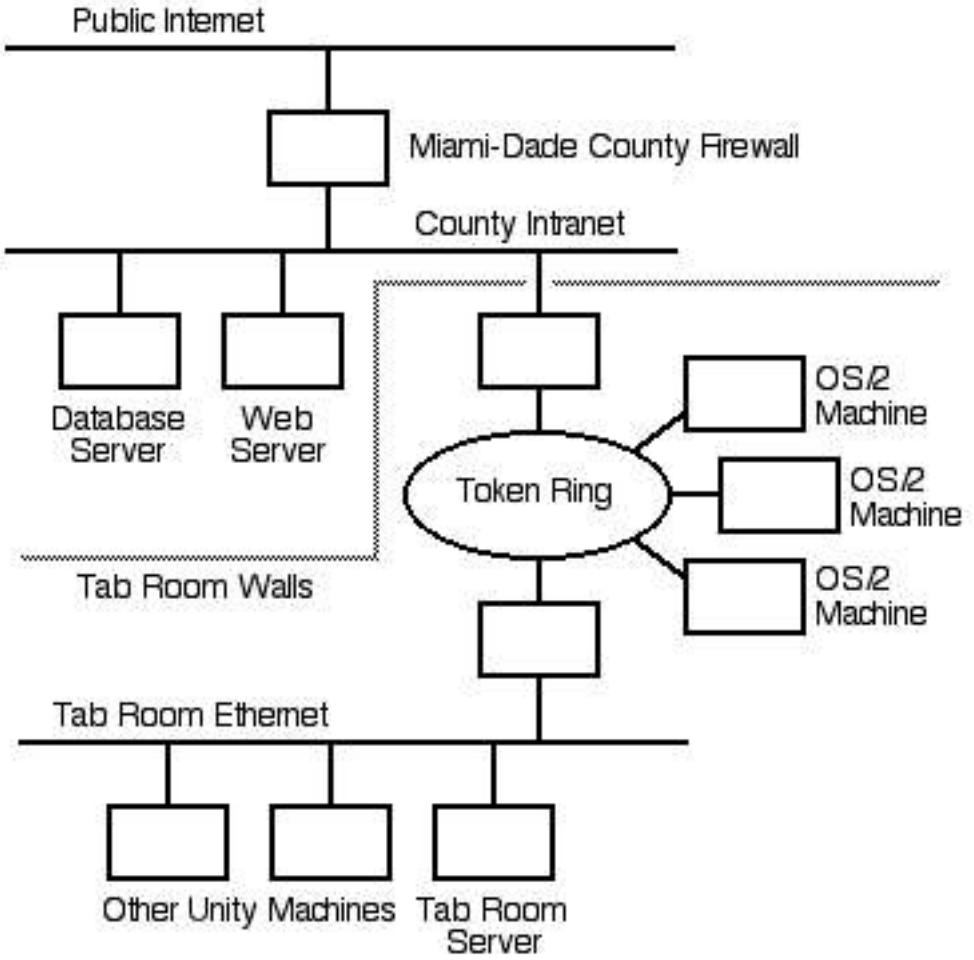
Douglas W. Jones and Tom C. Bowersox
Department of Computer Science
THE UNIVERSITY OF IOWA

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

http://www.cs.uiowa.edu/~jones/voting/diode/

The Problem

- Election result must be put on the net.
- Election database must be protected.
- Conflict resolved by:
  - Sneakernet or
  - Even odder solutions

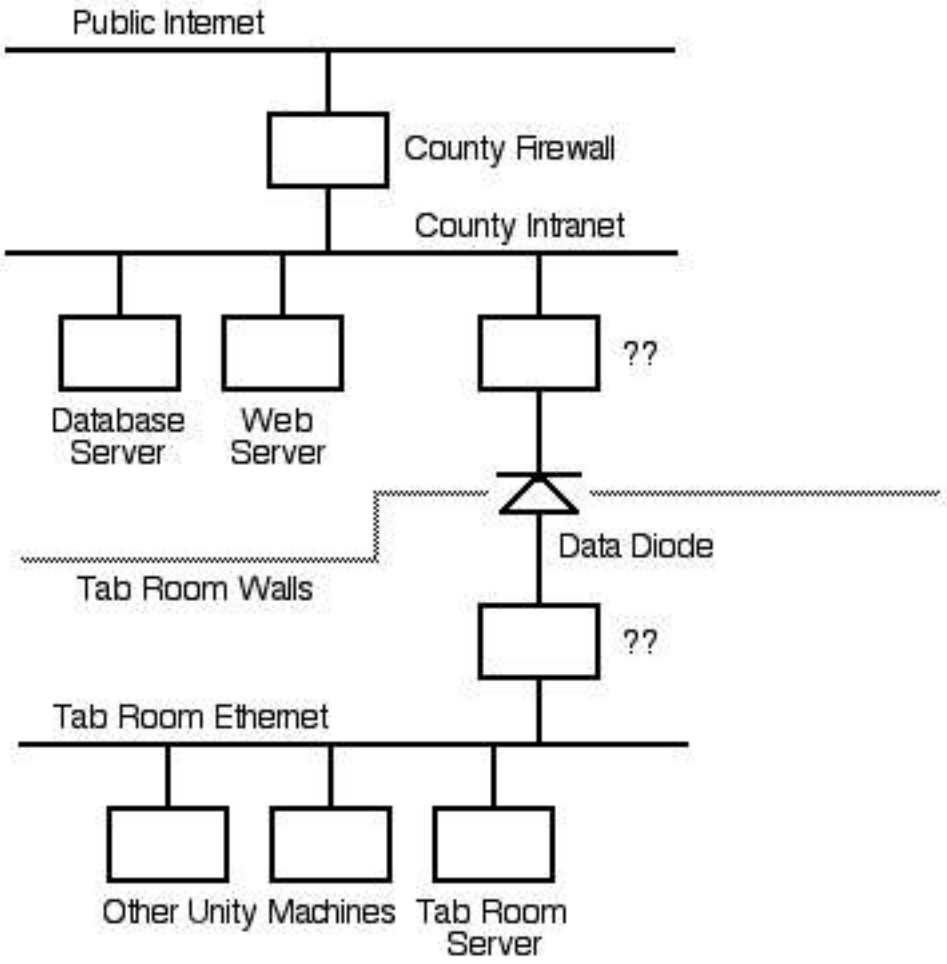


Miami Dade County Solution
Security through obscurity

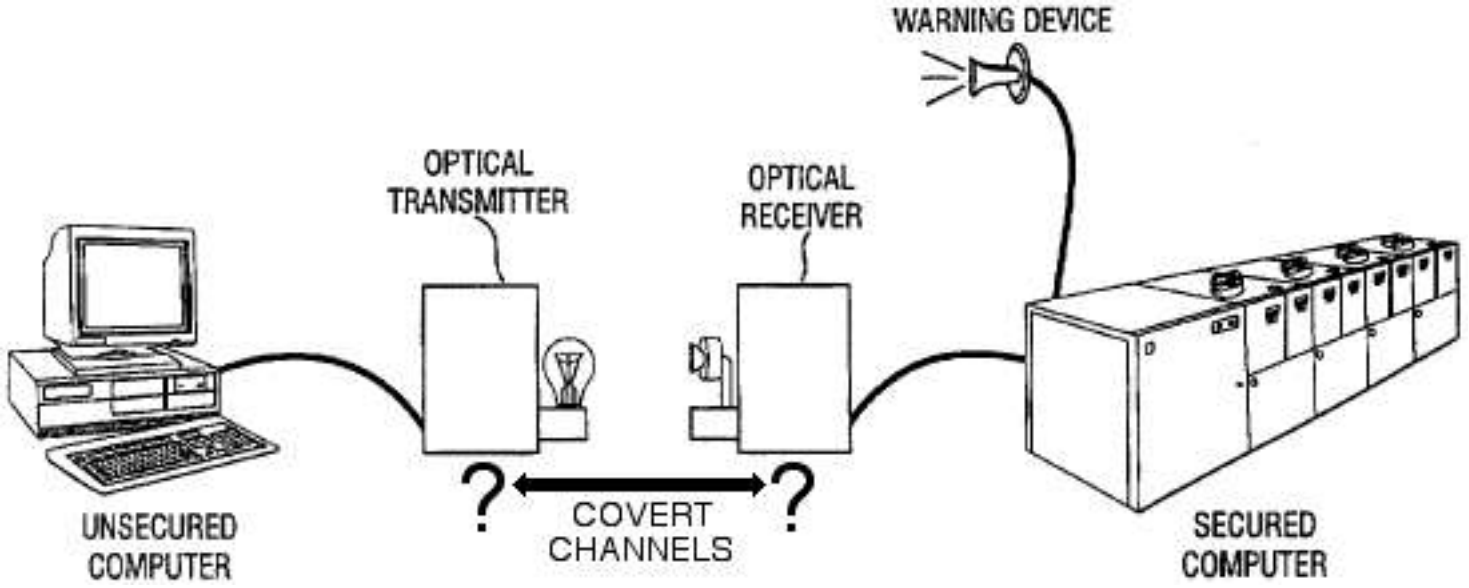
What we need

A data diode

- Allow data export
- Prevent data import
- Design understood by
  - Election observers
  - Election officials
  - Losing candidates



US Patent 5,703,562

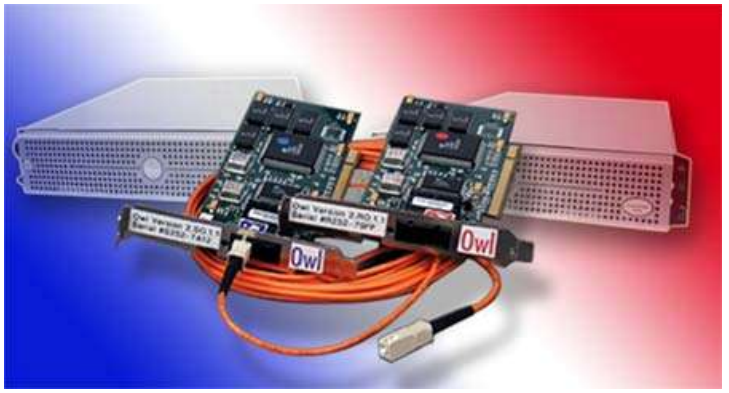


- Claims limited to up-hierarchy transmission
- Example given for RS-232 implementation
  - Transmit: 1 IC + 8 components + 5 volt supply
  - Receive: 1 IC + 4 components + 5 volt supply
- Explain this to a naïve suspicious observer!

Commercial Data Diodes

Owl Computing Technologies

www.owlcti.com



Tenix DATAGATE

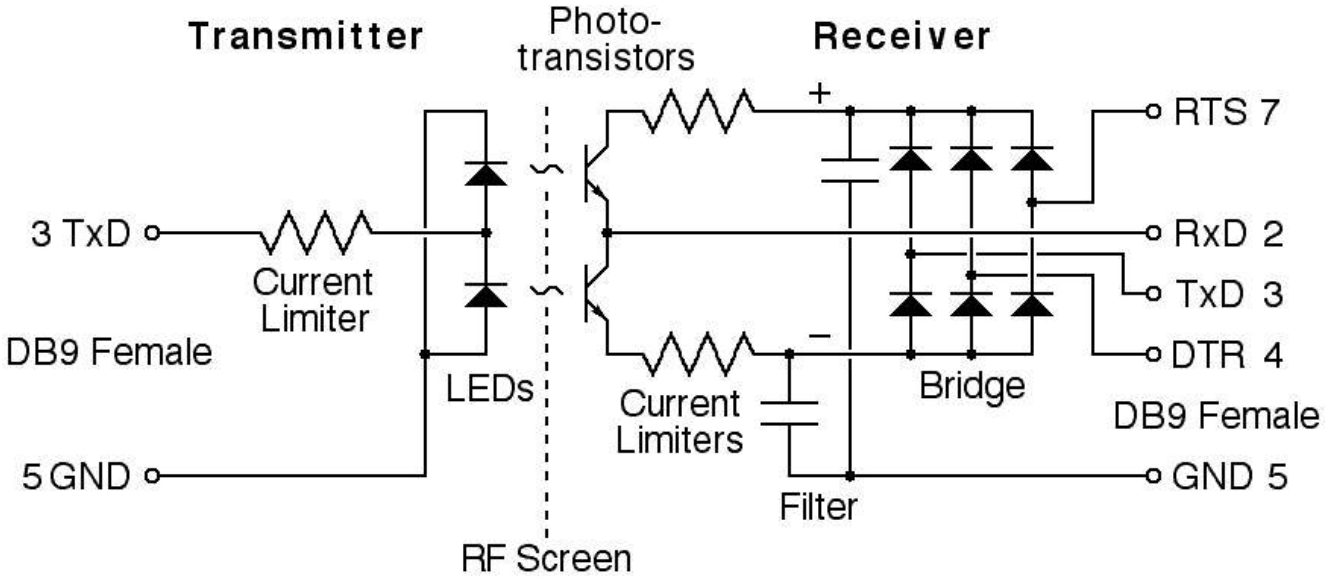
www.tenix.com

- Both based on fiber-optic technology
- Tenix Data Diode certified to EAL 7 under Common Criteria

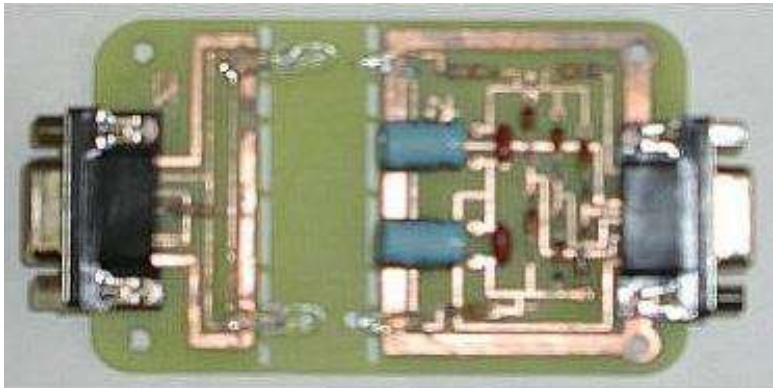
Design Transparency

- EAL 7 certification
  - Insufficient if the certifying agency is not trusted
  - What if the vendor cheats after certification?
- Therefore, we need
  - Complete design transparency
  - Open documentation
  - Rights of observers to inspect entire mechanism
  - Minimal complexity

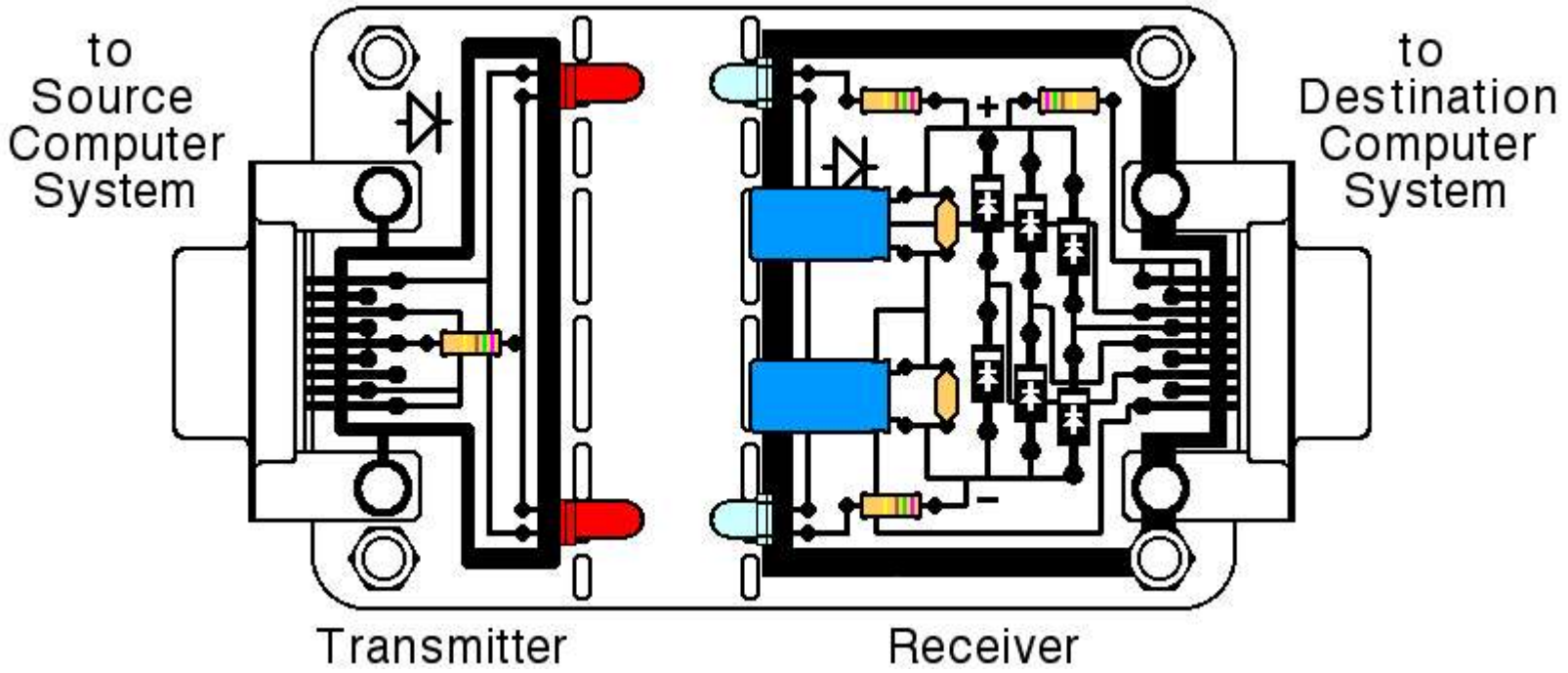
Our Design



- Avoid all black boxes
  - no 3-terminal devices
  - No ICs
- Extreme simplicity
  - Use RS-232



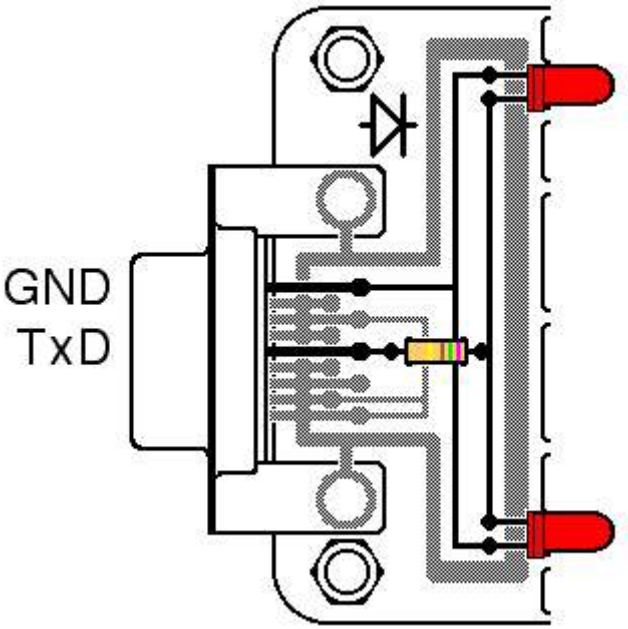
Explaining the circuit board



- Must explain function of
  - Every circuit trace
  - Every component

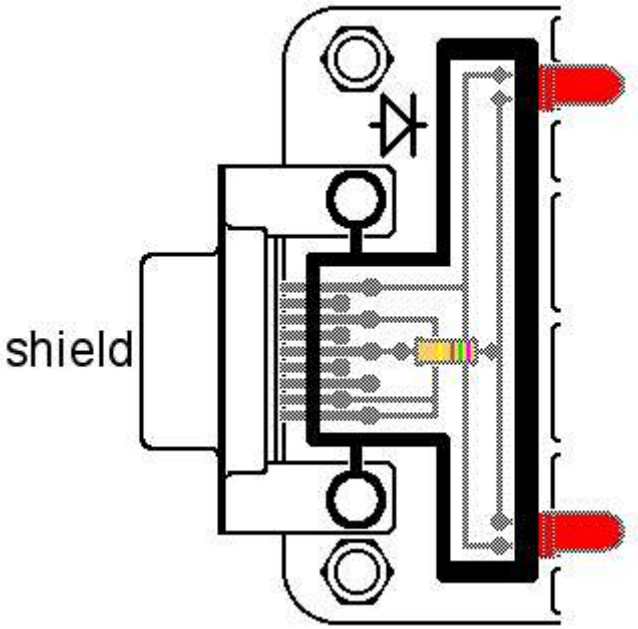
Explaining the Transmitter

- GND: signal ground
- TxD: transmit data
- When TxD is positive
  - Top LED lights
- When TxD is negative
  - Bottom LED lights
- Resistor needed as
  - Current limiter



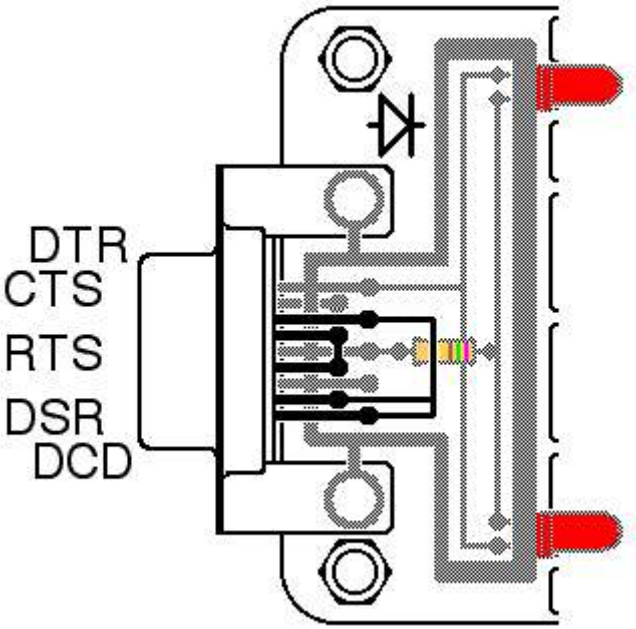
Explaining the Transmitter

- The shield pin in the cable
  - Connects to metallic sheath
- The shield pin on the board
  - Connects to trace that surrounds the electronics
- Together
  - These make it difficult to use conductors inside the shield as radio antennas



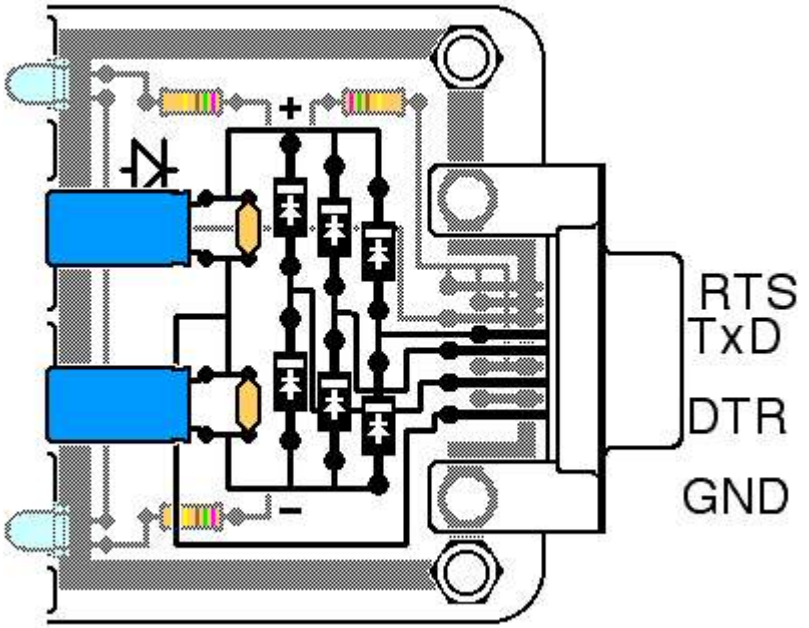
Explaining the Transmitter

- The loopback connections
  - Tell computer we're ready
- RTS to CTS
  - Request To Send (input)
  - Clear To Send (output)
- DTR to DSR and DCD
  - Data Terminal Ready (input)
  - Data Set Ready (output)
  - Data Carrier Detect (output)



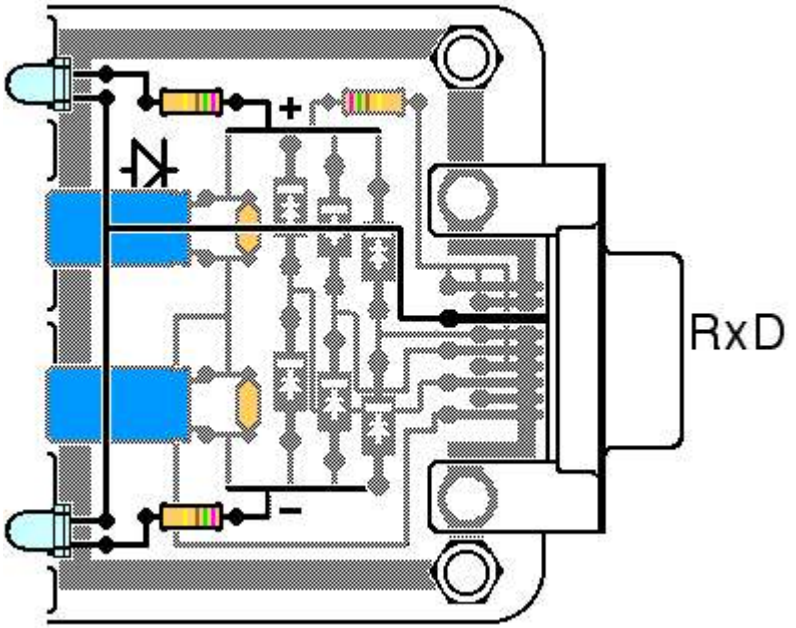
Explaining the Receiver

- The power supply
  - Uses RTS TxD and DTR
- Power from Serial cable
- Power from special cable
  - 2 batteries
  - AC power from wall outlet
- Capacitors and Diodes
  - Permit 60Hz operation



Explaining the Receiver

- The Receiver itself
  - Uses the power supply
  - Transmits to RxD output
- Top photodiode
  - Pulls RxD positive
- Bottom photodiode
  - Pulls RxD negative
- Resistors needed as
  - Current limiters



Using the Data Diode

- No reverse channel (almost)
  - Must rely entirely on forward error correction
  - Checksums (or better) to reject bad data
  - Redundancy to provide for correction
  - Operational status determined from downstream
- Sending from high to low security domain
  - Covert content in data is a big issue
  - Unlike most low to high transmission

Auditors and Wiretaps

- Data exported from EMS is public
- Observers should not trust the web server
- So, observers should be allowed wiretaps
  - Directly observe data-flow to server
  - Directly verify that data conforms to spec
- Free air (as opposed to fiber optic) optical data diodes offer excellent access to the data stream by observers!

Exporting Election Results

Using relational database terminology election results are a single relation over:

- Precinct (or split, for split precincts)
- Race (or contest)
- Candidate (or position with respect to contest)
- Votes for that candidate in that race in that precinct

What we need to do is export this entire relation

OASIS EML, A Bad Idea

- Requires header
  - Data diode invites an infinite stream
- Verbose
  - human audit difficult
- Covert channels
  - Complex rules for canonical form
- Difficult to checksum

```
#-- EML-20081104-US-CA-Santa_Clara_County-2216-1274.xml --#
<?xml version="1.0" encoding="UTF-8"?>
<CastVote xmlns="440-castvote.xsd">
<ElectionEvent>
  <Event>
    <EventName Id="n1274s213">
      Santa Clara County, CA, USA (2008-11-04)
    </EventName>
    <EventQualifier>Precint 2216</EventQualifier>
  </Event>
  <Election>
    <ElectionName>Presidency</ElectionName>
    <Contest>
      <ContestName>President</ContestName>
      <Selection>
        <Option>
          <OptionName>V. I. Lenin</OptionName>
        </Option>
      </Selection>
    </Contest>
  </Election>
  <Election>
    <ElectionName>Presidency</ElectionName>
    <Contest>
      <ContestName>Vice-President</ContestName>
      <Selection>
        <Option>
          <OptionName>Karl Marx</OptionName>
        </Option>
      </Selection>
    </Contest>
  </Election>
  <Election>
    <ElectionName>Senate</ElectionName>
    <Contest>
      <ContestName>Senator</ContestName>
      <Selection>
        <Option>
          <OptionName>William Lloyd Garrison</OptionName>
        </Option>
```

Reasonable Data Formats

- A repeating stream of checksummed records
- Tab separated fields?

```
IC15 President Lincoln 25   16384
CV06 Mayor     Thomas  42   32768
```

- XMLish but not really XML

```
<ITEM PRECINCT="IC15" RACE="President"
CANDIDATE="Lincoln" VOTES="25" />53895

<ITEM PRECINCT="CV06" RACE="mayor"
CANDIDATE="Thomas" VOTES="42" />41274
```

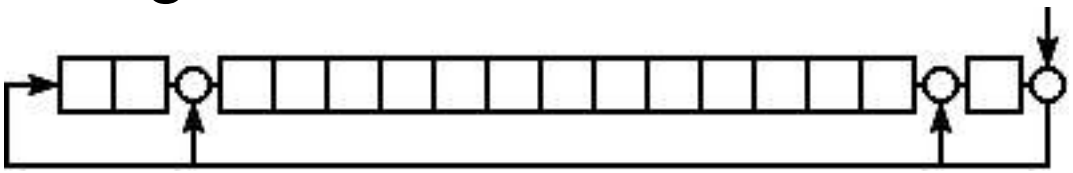
- We opt (on weak grounds) for XMLish

Covert Channels

- The Risk
  - Covert export of security keys from EMS
- The Defense
  - Rigid format constraints on data
    - No optional, permutable, or alternate elements
    - No free use of whitespace or line ends
  - Code audit on real-time checks in transmit code
    - No non-constant time delays allowed in transmitter

Transparent Checksums

- We have a transparent data diode design
- We have a transparent data format
- We need a transparent checksum algorithm
  - Understandable using highschool math
  - Easy to code in a bad programming language
- CRC-16 is not transparent!
  - Try explaining this: $X^{16} + X^{15} + X^2 + 1$
  - Or this:



Transparent Checksums

- A classic transparent but weak checksum
  - $S_0 = 0$; $S_{i+1} = (S_i + C_i) \bmod 256$
- A modest proposal
  - $S_0 = 0$; $S_{i+1} = (5S_i + C_i) \bmod 65536$
  - Akin to multiplicitative congruence PRNG
- What multipliers and moduli are best
- Is there a cryptographically secure hash code that meets our transparency goals?

Code to checksum data stream

```
#include <stdio.h>
/* filter to checksum each block of angle-bracketed text
   Reads from stdin and copies to stdout.
   Appends decimal checksum to each closing angle bracket.
   Angle brackets are included in the checksum.
   NOTE:  This code is dumb, bracket nesting is ignored and
   bracked imbalance is not checked.                          */
main ()
{
        int ch;
        unsigned int sum = 0;
        while ((ch = getchar()) != EOF) {
                putchar( ch );
                sum = (sum * 5 + ch) % 66636; /*accumulate*/
                if (ch == '<') {
                        sum = '<';            /*initialize*/
                } else if (ch == '>') {
                        printf("%1u",sum);
}       }       }
```

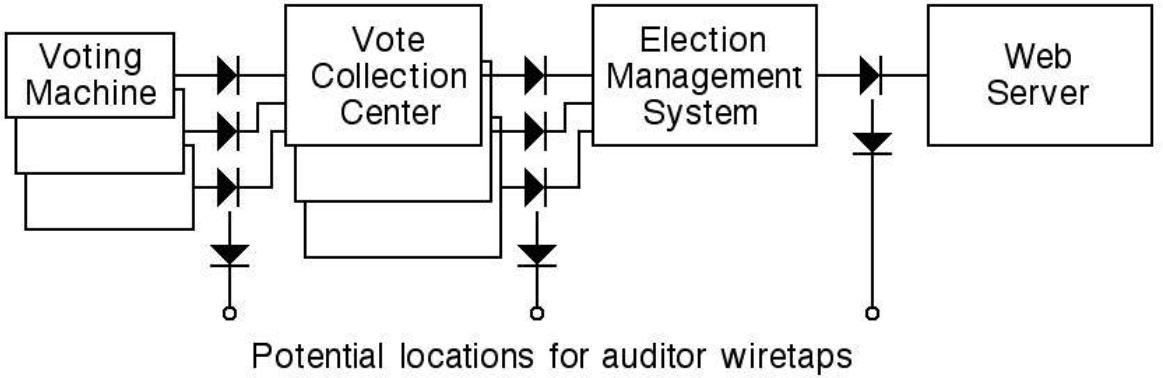
Even this is hard to explain, but it's in reach of a student who only has a semester of programming, perhaps in VB or worse

A Prototype Application

- Scaffolding
  - Extract results from example county data
  - Inject in model EMS database
- Demo code
  - Cyclically scan EMS database
  - Export through data diode
- Decent quality prototype application code
  - Receive data from data diode to mirror database
  - Server-side web application for results

Other Applications

- Upstream



- In voting machine