A Smaller, Stronger FPGA–based Voting Machine

EVT/WOTE ’09
AUGUST 10, 2009

Ersin Öksüzoglu
Dan S. Wallach
Previously on VoteBox

- VoteBox
  - Full featured DRE voting machine
VoteBox (Classic)

- **Pre-rendered user interface**
  - simplifies the **graphics** subsystem & **code size**

- **Network ballot replication**
  - increases the **availability** of voting records

- **Challenge option**
  - casts the votes **as intended**

- **Elgamal ballot encryption**
  - allows **tallying** the votes independently
Elgamal Homomorphic Encryption

- One way of encryption
  \[ E(c, r, g^a) = \langle g^r, (g^a)^r f^c \rangle \]

- Two ways of decryption
  \[ D(\langle g^r, g^{ar} f^c \rangle, a) = \frac{g^{ar} f^c}{(g^r)^a} \]
  \[ D(\langle g^r, g^{ar} f^c \rangle, r) = \frac{g^{ar} f^c}{(g^a)^r} \]
In a tampered VoteBox, we cannot detect privacy attacks
  - The random number can be used as a subliminal channel

VoteBox still needs to be smaller

<table>
<thead>
<tr>
<th>EVM</th>
<th>Language</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pvote</td>
<td>Python</td>
<td>460</td>
</tr>
<tr>
<td><strong>VoteBox</strong></td>
<td>Java</td>
<td>14500</td>
</tr>
<tr>
<td>Diebold AccuVote TSX</td>
<td>C++</td>
<td>64000</td>
</tr>
<tr>
<td>Sequoia Edge</td>
<td>C</td>
<td>124000</td>
</tr>
</tbody>
</table>
VoteBox Nano: First FPGA–based EVM

- Pre-rendered GUI
  - Minimized code size for easier inspection

- Challenge option
  - End to end cryptography

- True Random Number Generator
  - Better random numbers

- Session ID
  - Additional tamper–evidence mechanism

Hardware and software hybrid
A blank chip that the user can program on the field
- Emulate any chip

- Used for prototyping custom silicon
  - Accelerate designs taking the advantage of the parallelism

- Widely deployed in the industry ($2.75 billion in 2010)
  - Fast time to market
  - Low initial cost
  - Re-programmable hence easy to update
Xilinx Spartan-3E Starter Kit (~$150)

- 500k gate FPGA Chip
- Flash RAM
- DRAM
- VGA port
- Dot Matrix LCD (2x16)
- A rotary encoder
- RS232 serial ports
- Buttons and switches
- USB configuration port
- No CPU, GPU, network chip
VoteBox Nano Lacks

- Network replication and storage facilities
  - We have limited space on board

- Ethernet communication module
  - Instead we have RS232 port

- High resolution bitmap based GUI
  - We have character graphics
VoteBox Classic vs. VoteBox Nano

**STEP 1: Read Instructions**

To make your choice, click on the candidate’s name or on the box next to the candidate's name. A green checkmark will appear next to your choice. If you want to change your choice, just click on a different candidate or box.

**STEP 2: Make your choices**

- Gordon Bearce, Nathan Maclean (REP)
- Vernon Stanley Albury, Richard Rigby (DEM)
- Janette Froman, Chris Aponte (LIB)

**STEP 3: Review your choices**

- Gordon Bearce, Nathan Maclean
- Vernon Stanley Albury, Richard Rigby
- Janette Froman, Chris Aponte

**STEP 4: Record your vote**

Click to go back to instructions

Previous Page

Next Page
To make a choice highlight the candidate's name by turning the knob. Click the knob to vote for the highlighted candidate then select the next page to continue. To de-select a candidate click again.

Ballot Definition

VoteBox Nano
JTAG (Joint Test Action Group)

- IEEE port standard for IC’s to:
  - Debug
  - Program
  - Monitor

- Daisy chain connection for all the components on board
  - One wire data in
  - One wire data out

For FPGAs, JTAG is used for
1. Bitstream upload and download
2. Software upload and download
3. Accessing software debugger
Programming FPGA At (Hardware)

- USB  JTAG
- Triggers
- Session ID
- Captured from TRNG

34FE1630
F009EFBA
1036E5CC
F61EABC5
789F2EA0
495F24A0
B9DE245S
001B5B42
2CE4A025

Session ID
..9F23

Bitstream

Software

Programming
Done !!!
Programming FPGA (Software)

USB → JTAG

Done !!!
The design is ready!

Software

Bitstream

FPGA is sealed

Session ID

Write it down!
Attestation

Readback bitstream

Done !!!

Same ?

Seal is broken

Original bitstream

Compare

Bitstream from FPGA

..0932

..CC21

..0932

..7FED

..1456

..3247

..6831

..127F

..E2D6

..E12C

..FAFA

..ED92

..259A

..2201

..F032

..CC21

Same ?

Seal is broken
Interesting Attacks

- Upload a new bitstream

- Change software
  - JTAG port is monitored
  - Session ID is read-only
## Source Code Length (Software)

<table>
<thead>
<tr>
<th>EVM</th>
<th>Language</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pvote</td>
<td>Python</td>
<td>460</td>
</tr>
<tr>
<td><strong>VoteBox Nano</strong></td>
<td>C</td>
<td>996</td>
</tr>
<tr>
<td>VoteBox (Stripped)</td>
<td>Java</td>
<td>~7300</td>
</tr>
<tr>
<td>VoteBox (Full)</td>
<td>Java</td>
<td>14500</td>
</tr>
<tr>
<td>Diebold AccuVote TSX</td>
<td>C++</td>
<td>64000</td>
</tr>
<tr>
<td>Sequoia Edge</td>
<td>C</td>
<td>124000</td>
</tr>
</tbody>
</table>
Comparison

- **Pvote**
  - 460 lines Python
  - Python Libraries
  - Linux Kernel
  - PR-GUI
  - SHA1

- **VoteBox (Full)**
  - 14500 lines JAVA
  - JAVA Libraries
  - Linux Kernel
  - PR-GUI
  - Network ballot rep.
  - Elgamal enc. DSA

- **VoteBox Nano**
  - 122 kB executable
  - FPGA Modules Custom Modules
  - PR-GUI
  - Challenge
  - Session ID
  - TRNG
  - Elgamal enc. DSA
We have shown that a very compact EVM can be built using an FPGA with following features:

- Externally verifiable attestation
- True Random Number Generator
- Elgamal Encryption and DSA
- Challenge Option
- Pre-rendered GUI
- No underlying OS
Cast or Challenge [Benaloh]

- At the last step, the voter is given two options
  - **Cast**
    - The votes are valid
    - Usual flow
  - **Challenge**
    - The votes are invalidated
    - FPGA reveals the random numbers

- FPGA only publishes the random numbers, the secret key is still safe

- With a certain amount of challenges, the results are reliable enough
TRNG has 128 ring oscillators, each consisting of 3 inverters.

\( f_s \) is 25 MHz and throughput is 195 kB/s.
FPGA Structure

- Jumpers
- 50 MHz Clock
- Knob
- DRAM
- FLASH RAM
- Buttons
- LCD Display 2x16 character
- LEDs
- Slide Switches

IO ports

4-input LUT
- FF
- MUX
- Carry

4-input LUT
- FF
- MUX
- Carry

Motherboard/Development board
Trivial Attacks

- Theft of the device
  - No secret data is stored in long term

- Tapping serial port
  - The votes are encrypted
  - Encryption is probabilistic
Hardware Modules

FPGA Area Utilization

- Crypto Accelerator: 27%
- Microblaze CPU: 18%
- DRAM interface: 14%
- TRNG: 8%
- Other: 23%
- LCD: 1%
- Debug: 2%
- RS232: 2%
- VGA: 5%

<table>
<thead>
<tr>
<th>Hardware</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crypto Module</td>
<td>760</td>
</tr>
<tr>
<td>TRNG</td>
<td>520</td>
</tr>
<tr>
<td>Other</td>
<td>483</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1763</strong></td>
</tr>
</tbody>
</table>
JTAG port

TDI: (Test Data In)
TDO: (Test Data Out)
TCK: (Test Clock)
TMS: (Test Mode Select)

The line is tripwired to the Session ID
FPGA (Field Programmable Gate Array)

- 500k gate FPGA Chip
- Flash RAM (16 MB)
- DRAM (32 MB)
- VGA port
- Dot Matrix LCD (2x16)
- A rotary encoder
- RS232 serial ports
- Buttons and switches
- USB configuration port
- Ethernet Port
- PS/2 port
- 8 LEDs

Xilinx Spartan-3E 500 Starter Kit
JTAG port

TDI: (Test Data In)
TDO: (Test Data Out)
TCK: (Test Clock)
TMS: (Test Mode Select)

For FPGAs JTAG is used for
1. Bitstream upload and download
2. Software upload and download
3. Accessing software debugger