The Return of Insecure Brazilian Voting Machines

Diego F. Aranha
Aarhus University
89 PUBLICATIONS 726 CITATIONS
SEE PROFILE

Pedro Yóssis Silva Barbosa
Universidade Federal de Campina Grande (UFCG)
13 PUBLICATIONS 25 CITATIONS
SEE PROFILE

The Return of Insecure Brazilian Voting Machines

Diego F. Aranha, UNICAMP

dfaranha@ic.unicamp.br

@dfaranha

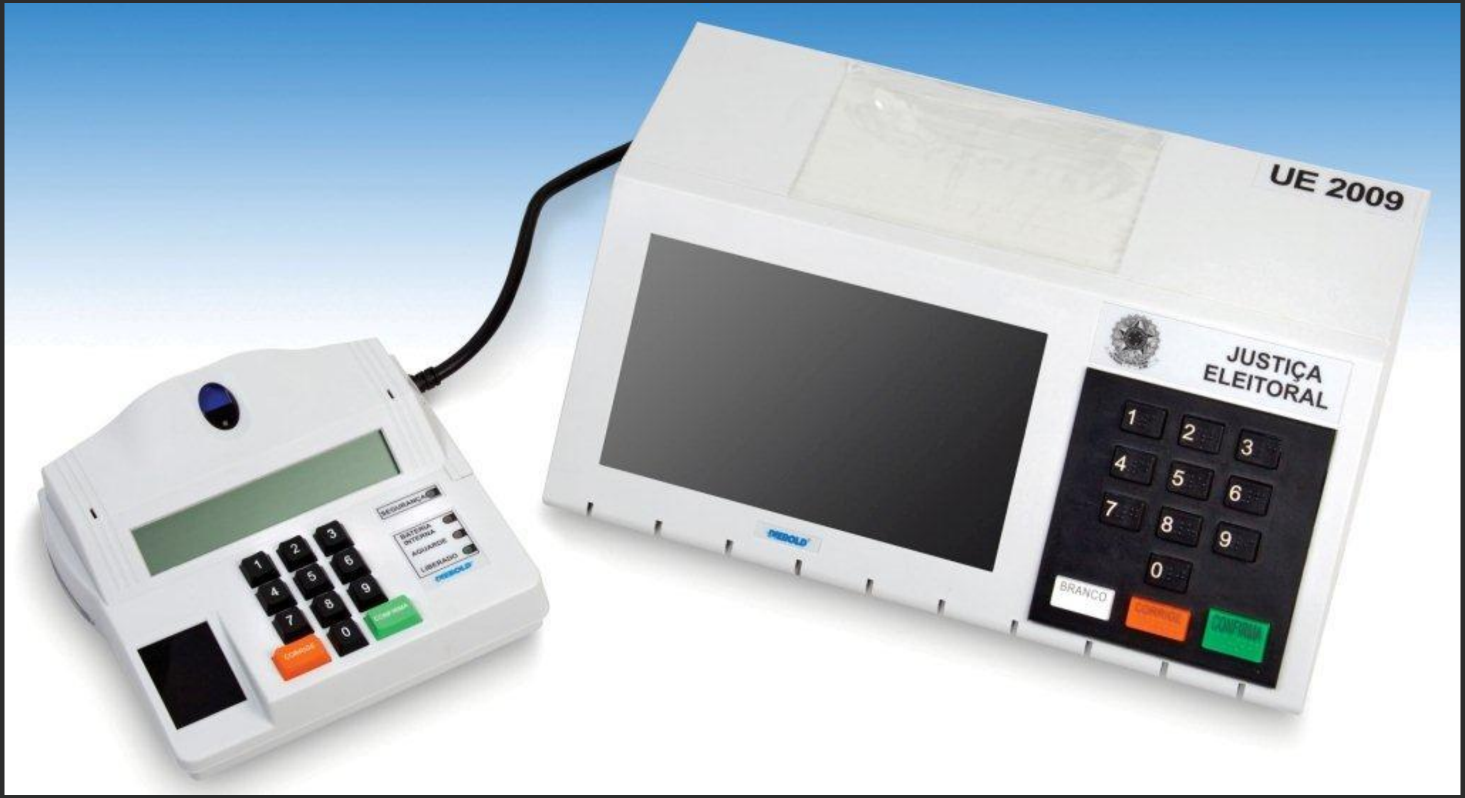
Joint work with Pedro Barbosa, Thiago Cardoso, Caio Lüders, Paulo Matias

Context

Brazilian elections are special:

- Massive (140M voters, 81% turnout)
- Held every 2 years
- Became electronic in 1996 (fully in 2000)
- Controlled/executed/judged by a single entity (SEC - Superior Electoral Court)

Context



Source: Diebold

Context

Brazilian paperless DRE voting machines:

- **Claimed** 100% secure (but only tested in 2012...)
- Hardware manufactured by **Diebold** (> 0.5M)
- Software written by SEC since 2006 (> 24M LOCs)
- Adopted GNU/Linux in 2008 (after **Windows CE...**)
- Experimented with **paper records** in 2002
- Identify 50% of the voters with **fingerprints** since 2011
- Highly vulnerable against **insiders**

Algorithm

1. Software **installation** (a card installs 50 machines)
2. Zero tape **printed** (7-8 AM)
3. Voting session **opened**
4. Votes **cast**
5. Voting session **closed (5PM)** and poll tape **printed**
6. Media **written** with public products (PT, DRV, LOG)
7. Public products **transmitted** to central tabulator

Public Security Tests

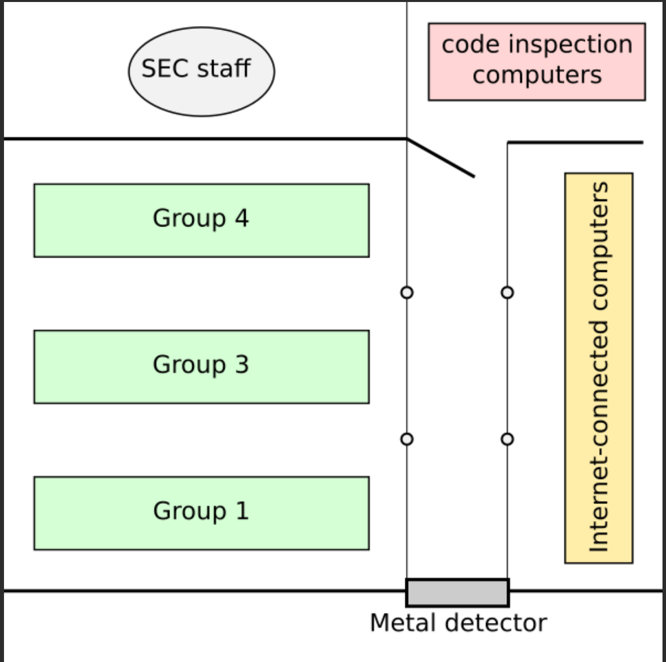
Objective: Untraceable violation of **ballot integrity/privacy**

Extremely restricted tests:

1. No **pen/paper** when inspecting source code
2. Only **3 days** to inspect code and **4 days** to mount attacks
3. Participants needed to be **pre-approved** by SEC
4. Attacks needed to be **pre-approved** by SEC
5. No **guarantees** about software version (correct or recent?)
6. Intrinsic **conflict of interests**

Public Security Tests?

"Brazil is the **only** country to **openly** evaluate its voting system"



Vulnerabilities from 2012

- Serious vulnerability in **vote shuffling mechanism**
- Massive **sharing** and insecure **storage** of keys
- Voting software checks **itself** through signatures
- No **ballot secrecy** or **integrity** of software/results
- **Insecure** development process
- **Inadequate** threat model
- Internal culture lacks **transparency**

Digital Record of the Votes (DRV)

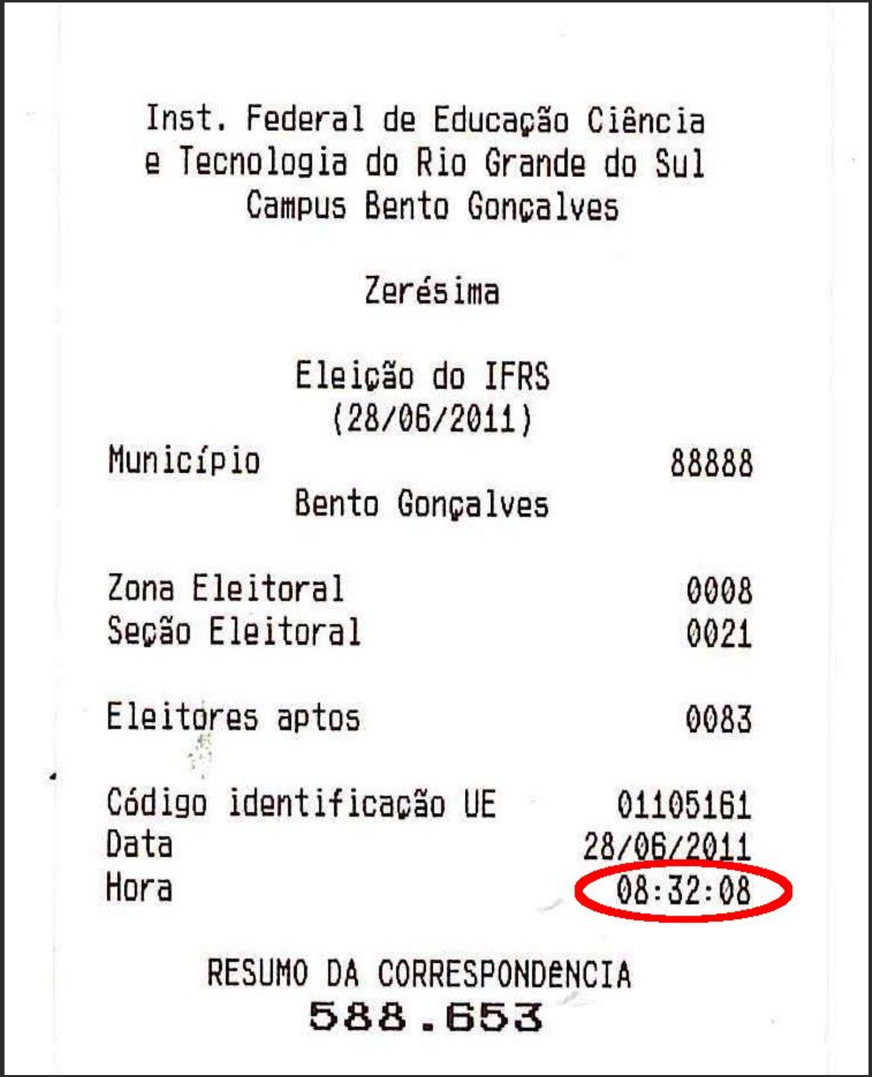
Governor	Senator	President
71	31	37
	BLANK	
13		
71	NULL	
		BLANK
		37

Warning: Advanced Cryptanalysis

```
grep -r rand *
```

Match in DRV.cpp! Seed?

```
srand(time(NULL))
```



Inst. Federal de Educação Ciência
e Tecnologia do Rio Grande do Sul
Campus Bento Gonçalves

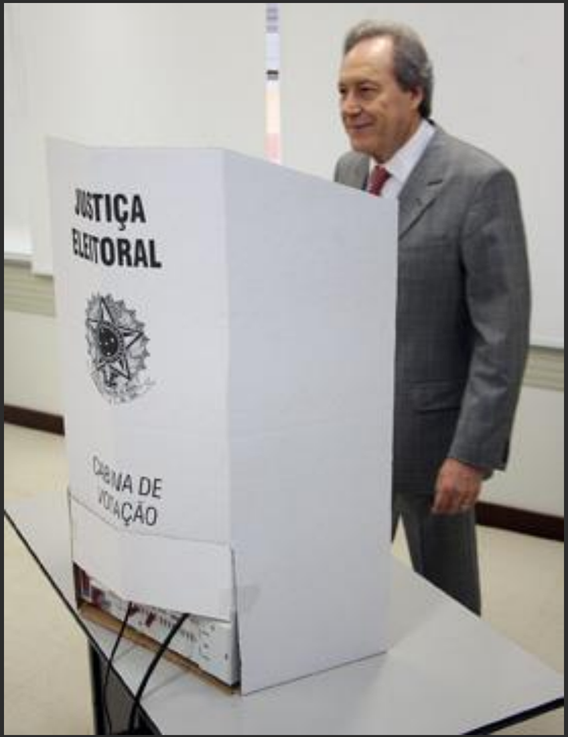
Zerésima

Eleição do IFRS
(28/06/2011)

Município	88888
Bento Gonçalves	
Zona Eleitoral	0008
Seção Eleitoral	0021
Eleitores aptos	0083
Código identificação UE	01105161
Data	28/06/2011
Hora	08:32:08

RESUMO DA CORRESPONDENCIA
588.653

Defense in depth?



```
File 1/1: lew.jpg
File name       : lew.jpg
File size       : 47009 Bytes
MIME type       : image/jpeg
Image size      : 276 x 360
Camera make     : Canon
Camera model    : Canon EOS-1Ds Mark III
Image timestamp : 2010:10:03 11:20:37
```

Conclusions from 2012

- Trivial to recover votes in order
- Trivial to recover a specific vote

Eliminate the DRV and do not store metadata!

"Fixed" by adopting **custom** algorithm with system entropy, although voting machine has **two hardware RNGs**

Installation as attack vector

2017: Researchers would not have access to cryptographic keys...

...but only because they erased them!

```
grep -r KEY *
```

Match in ueminix.c!

```
#define UEMINIX_BLOCK_KEY {0x34, ...}
```

Technical details

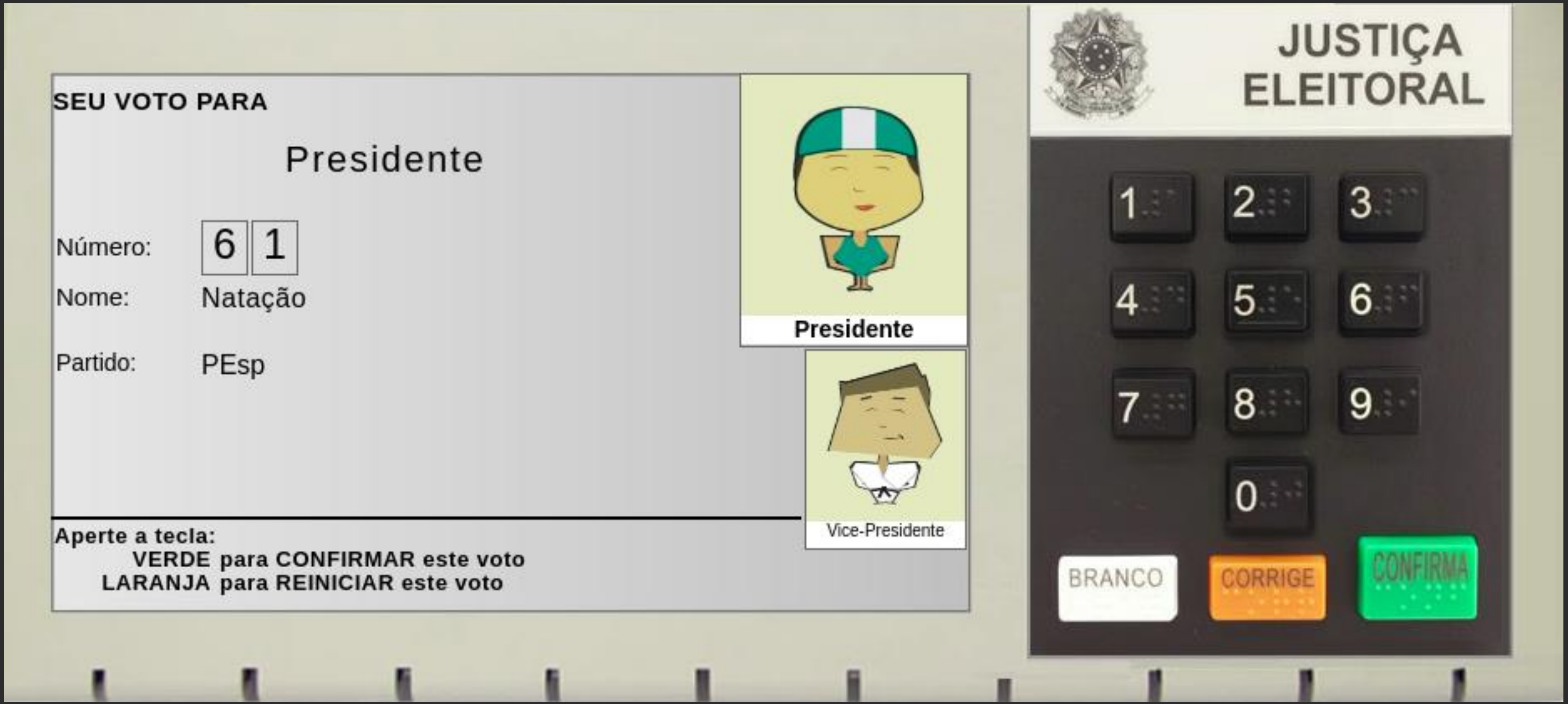
Many deployed cryptographic algorithms:
- Install cards encrypted with AES-XTS-256'
- ECC-based signatures for integrity checking

Signatures both in userland and kernel mode.

Keys for signing **results** also stored in install cards.

There is more!

- Found two shared libraries **without** signatures
- Manipulated **LOG contents**
- Tampered with key generation for **DRV**
- Plugged-in USB keyboard to **issue commands**
- Voting software was **linked** against them
- Changed software version/**screen contents**
- **Arbitrary code injection/execution**



SEU VOTO PARA

Presidente

Número: 6 1

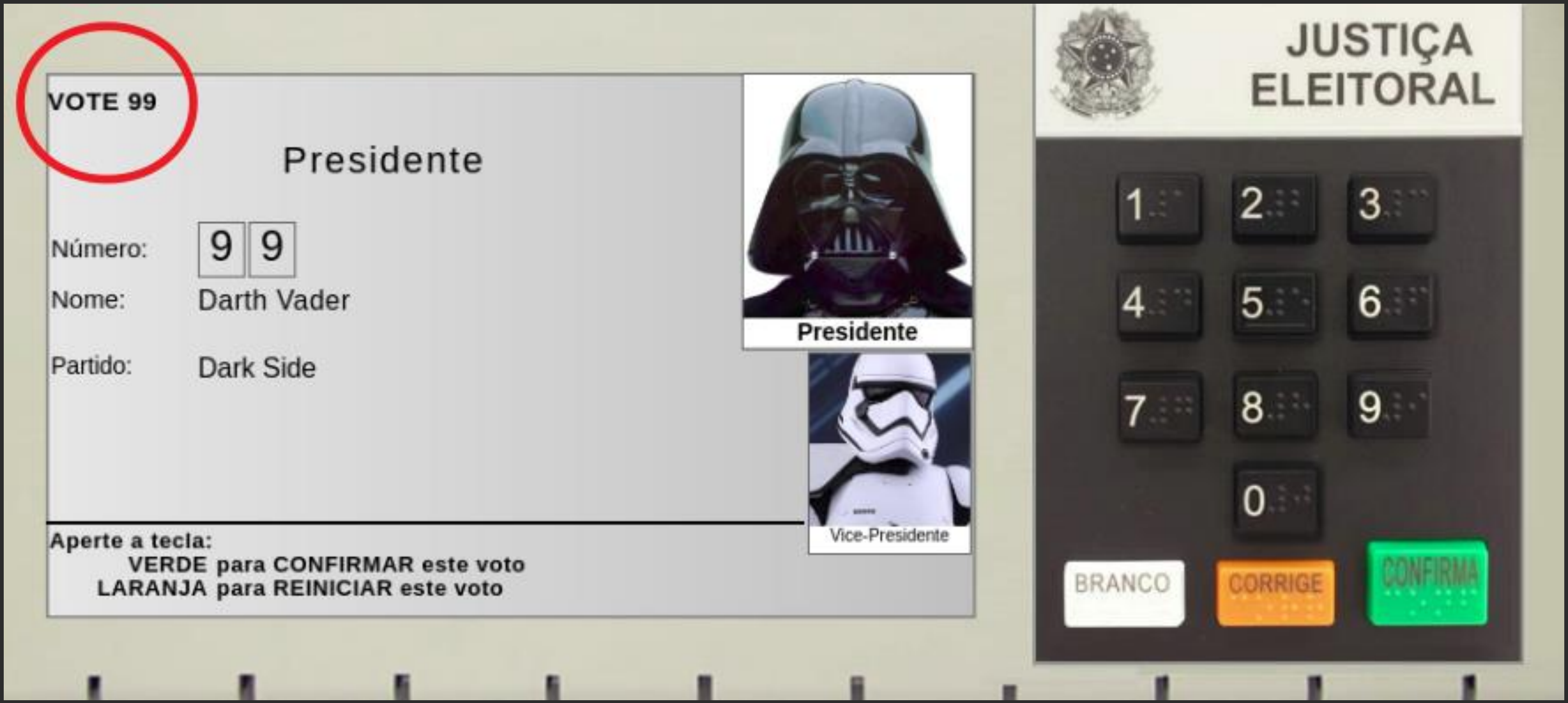
Nome: Natação

Partido: PEsp

Presidente

Vice-Presidente

Aperte a tecla:
VERDE para CONFIRMAR este voto
LARANJA para REINICIAR este voto



VOTE 99

Presidente

Número: 9 9

Nome: Darth Vader

Partido: Dark Side

Aperte a tecla:
VERDE para CONFIRMAR este voto
LARANJA para REINICIAR este voto

Conclusions from 2017

- **Insecure** encryption of install cards
- **Insecure** integrity checking
- Another team found same key without access to source code (**fully external attack**)

Automate signing, deploy proper key management!

"Fixed" by deriving keys from BIOS, still shared by all voting machines and vulnerable to **insiders**.

Current problems

1. Software is **secret** for > 20 years
2. Software is demonstrably **insecure**
3. No paper record for **recount**
4. No effective means to **audit** the system
5. **Conflicts of interest** everywhere
6. **Insider attacks** completely disregarded

Future

1. Voter-Verified Paper Audit Trail for **security**
2. Auditable software for **transparency**
3. Social control mechanisms for **participation**
4. Technical community needs to be **vocal**

With increasing political polarization, it is critical that elections can be **independently verified.**

Thanks! Questions?

Diegо F. Aranha, UNICAMP
dfaranha@ic.unicamp.br
@dfaranha
http://www.ic.unicamp.br/~dfaranha

References:

[1] Software vulnerabilities in the Brazilian voting machine.
In: Design, Development, and Use of Secure Electronic Voting Systems (2014)
[2] Crowdsourced integrity verification of election results. (2016)
[3] The Return of Software Vulnerabilities in the Brazilian voting machine. (2018)