Torturing OpenSSL

Valeria Bertacco
University of Michigan

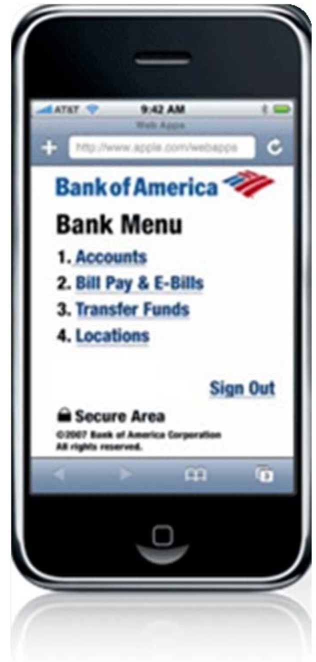
with Andrea Pellegrini and Todd Austin

Back Hat ‘12 Jul 25, 2012

Cryptography is all around us

email

online shopping



internet banking

electronic passports

playstation 3

blu-ray player

Value of Cryptography

The Security Division of EMC

$2.1 billion

1,300 employees

by Symantec

$5.7 billion

1,000 employees

$39 billion

18,000 employees

$82 billion

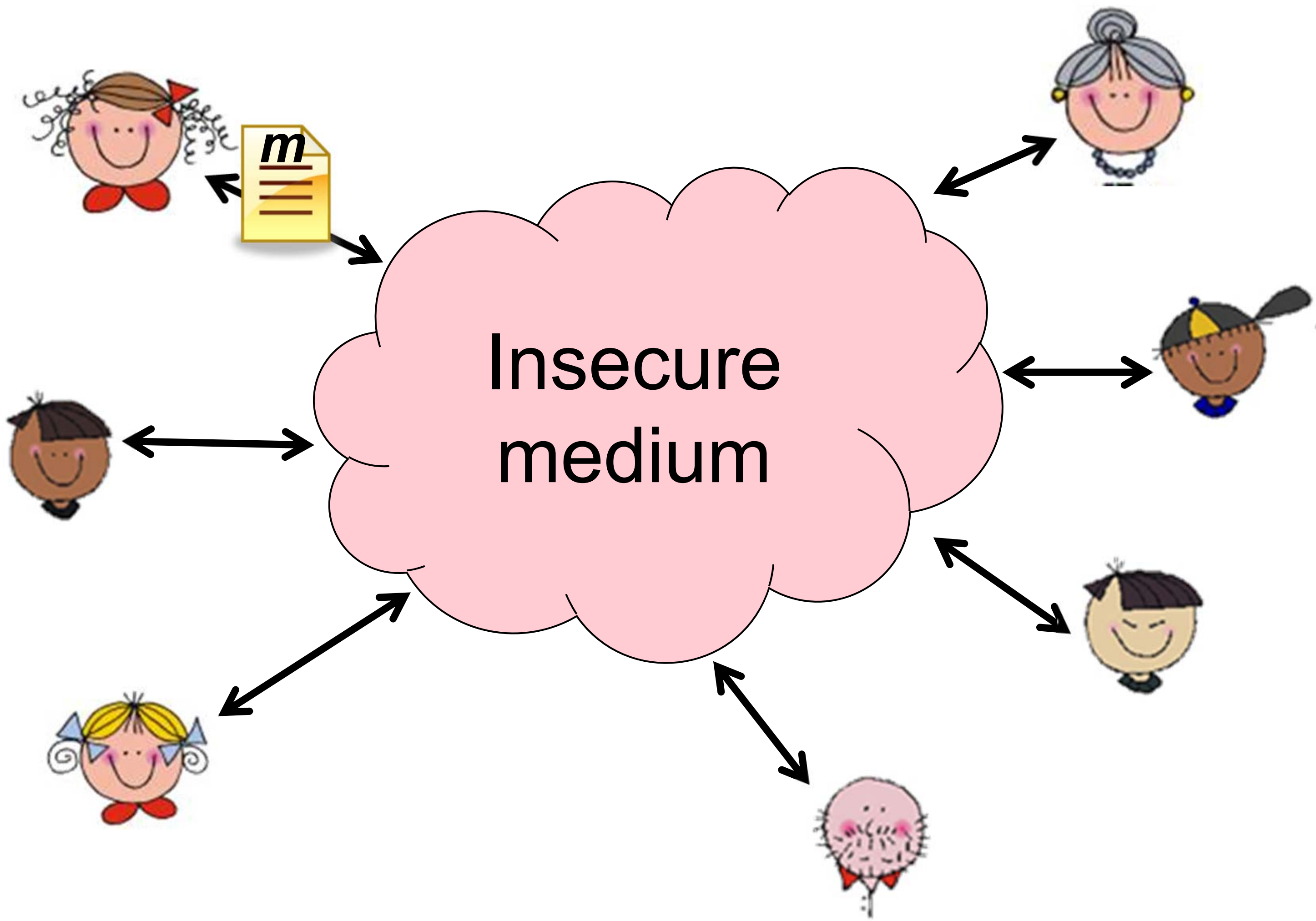
34,000 employees

Source: Wiki & NASDAQ

Outline

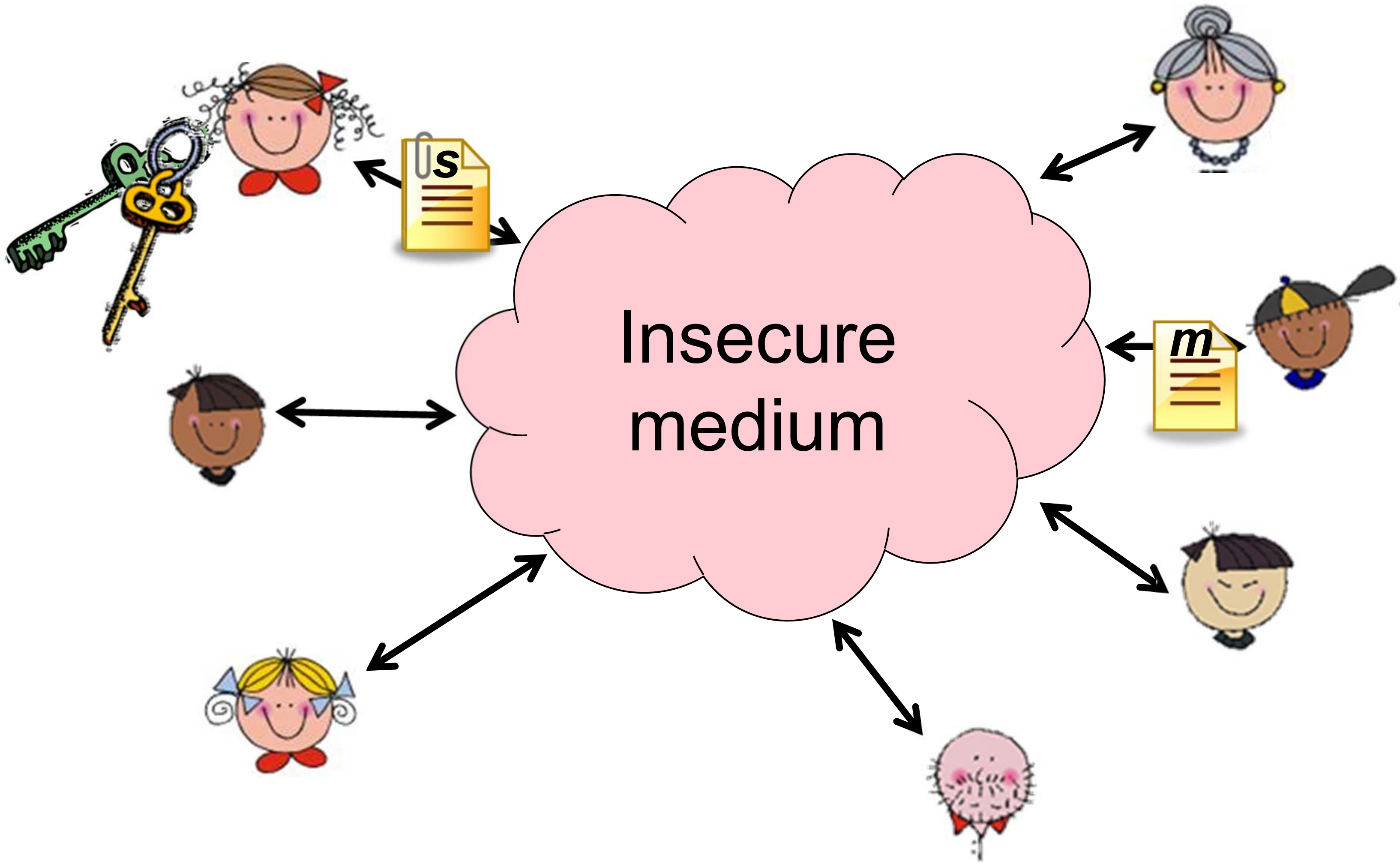
- Cryptography Introduction
- RSA authentication
- Attacks to RSA authentication
- OpenSSL implementation
- Private key extraction
- Fault injection
- Conclusions

What is Authenticated Communication?



How do we enable authenticated communication?

Asymmetric Cryptography

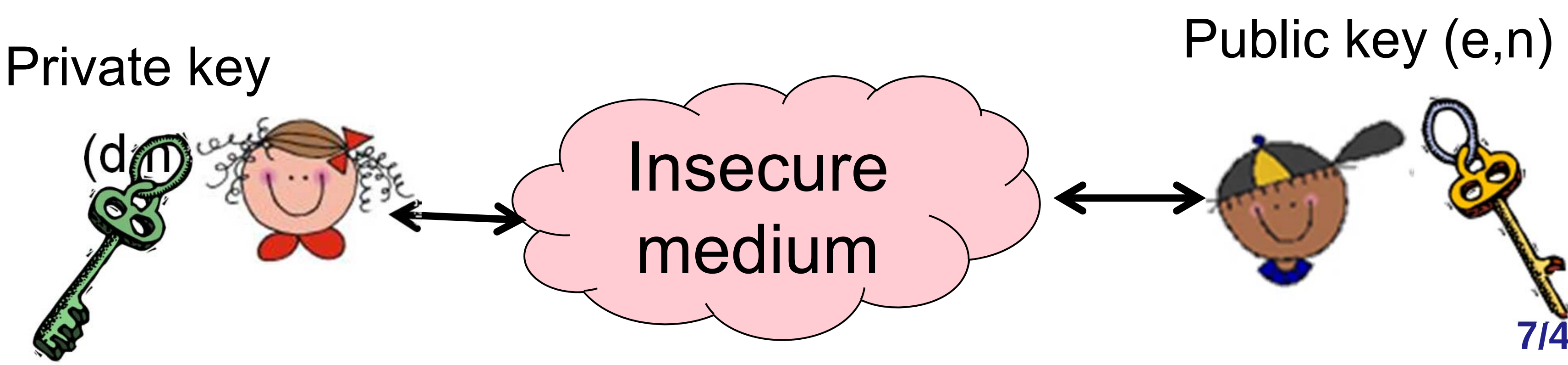


RSA Keys

The protocol is based on *two number pairs*, called keys

1. Choose two large prime numbers p & q
2. Compute n = p*q
3. Choose two numbers, d & e such that:
   $d*e = 1 \bmod ((p-1)(q-1))$

   Effect: $m^{de} \bmod n = m \bmod n$
4. Keep (d,n) as the secret private key
5. Advertise (e,n) as the public key



RSA Authentication

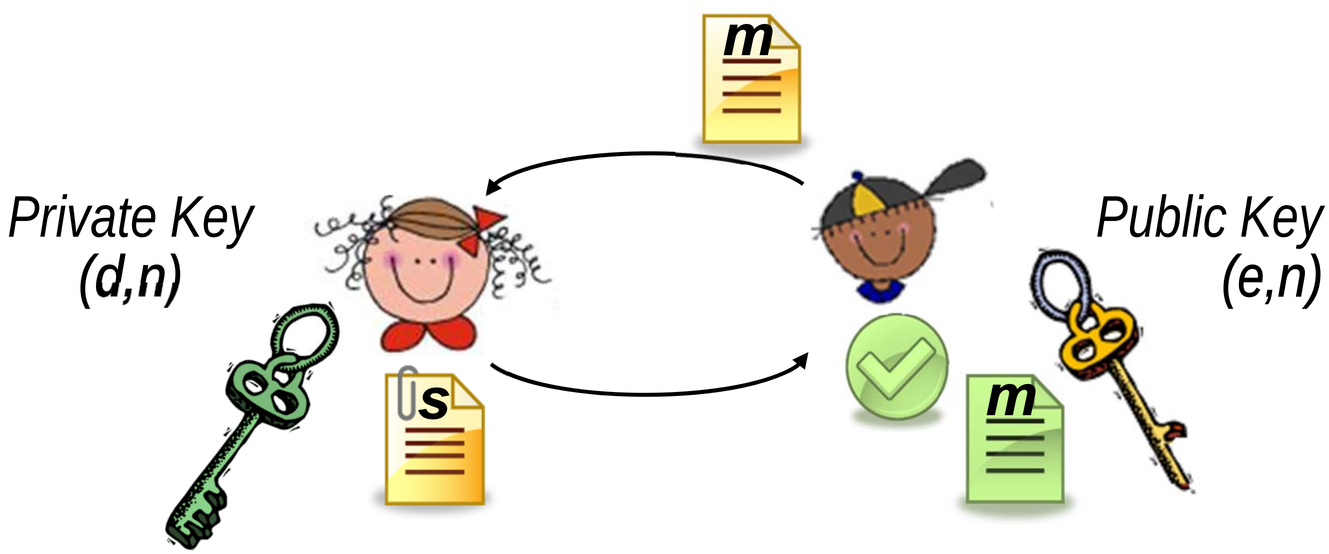
Correct Authentication:

- Server challenge:

$$s = m^d \bmod n$$

- Client verifies:

$$m = s^e \bmod n$$



Outline

- Cryptography Introduction
- RSA authentication
- Attacks to RSA authentication
- OpenSSL implementation
- Private key extraction
- Fault injection
- Conclusions

Are These Algorithms Secure?

(i.e., cryptanalysis)

Attacking the algorithm

by guessing key

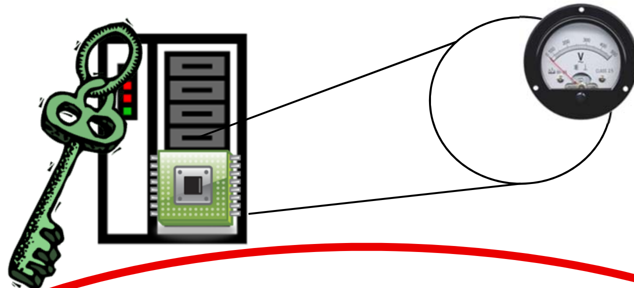
13506641086599522334960321627880596993888147560566702752448514385152651060485953383394028715057190944179820728216447155137368041970396419174304649658927425623934102086438320211037295872576235850964311056407350150818751067659462920556368552 94….

2009: Researchers brute forced a 768bits key over several computation years

Attacking the implementation

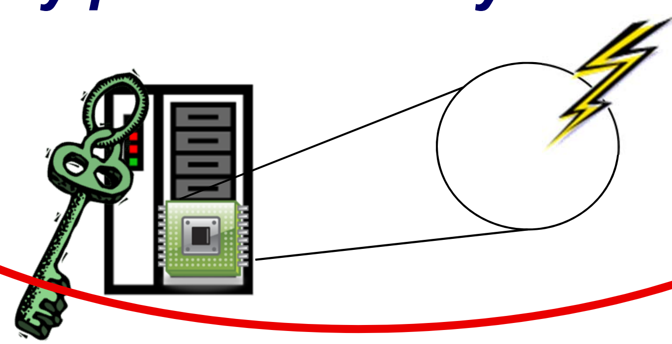
Side-channel

by monitoring side effects



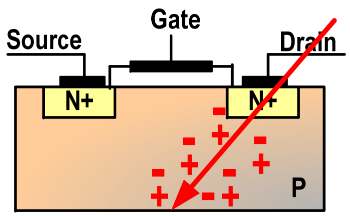
Fault-Based

a faulty processor may leak secrets



Attacks via Transient Faults

- Transient fault:
  a short perturbation of a logic value in a circuit:
  - Typically lasts <1 clock cycle
  - If latched, can cause permanent computation errors
- Transient faults occur naturally in silicon due to
  - Cosmic rays
  - Alpha particles
  - Location, density, frequency cannot be controlled

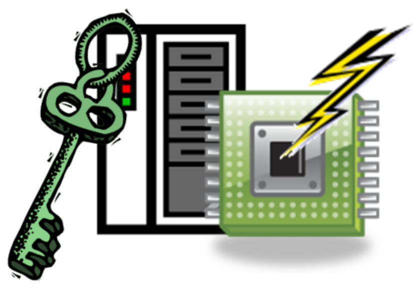


- This talk's focus:

Is it possible to perpetrate a security attack via transient faults?

Fault-Based Attacks

Cause errors in the system: a faulty computer may leak secrets



- Theoretical on some RSA implementations
  - Chinese Remainder Theorem
  - Left-to-right exponentiation

"On the Importance of Checking Computations", Boneh et al.

- Demonstrated on simple components
  - Smart Cards & Microcontrollers

"Fault attacks on RSA with CRT: Concrete results and practical countermeasures", Aumuller et al.
"A practical fault attack on square and multiply", Schmidt et al.

Faulty RSA Authentication

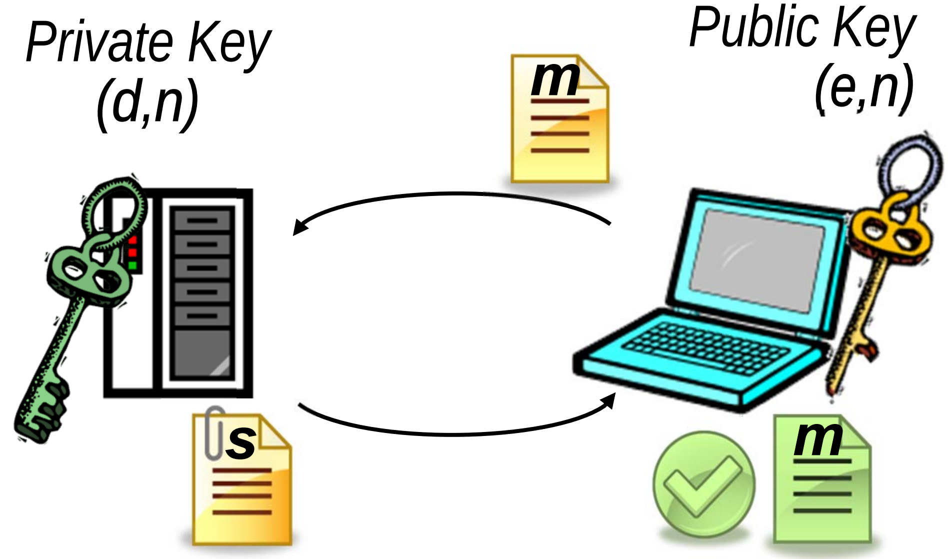
Correct Authentication:

- Server challenge:

$s = m^d \bmod n$

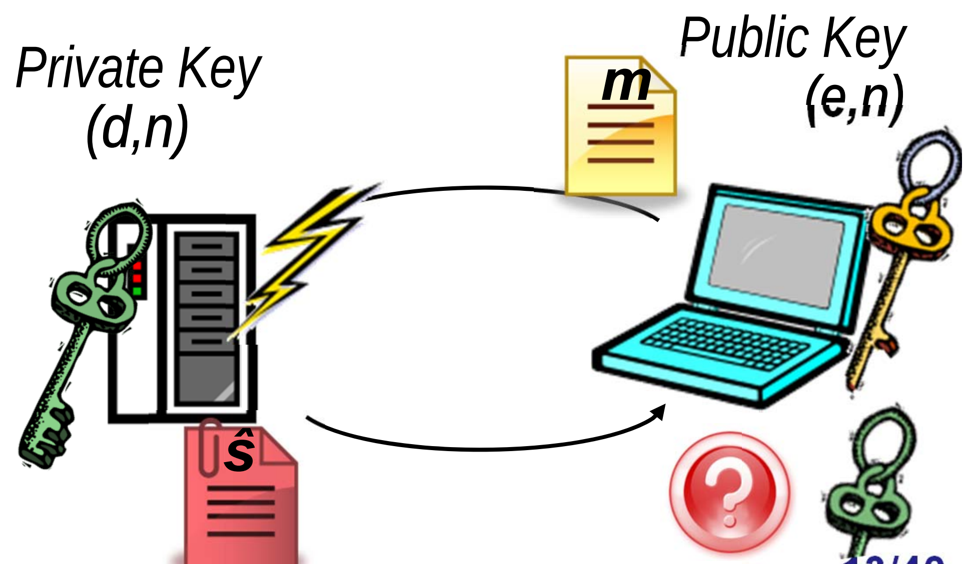
- Client verifies:

$m = s^e \bmod n$

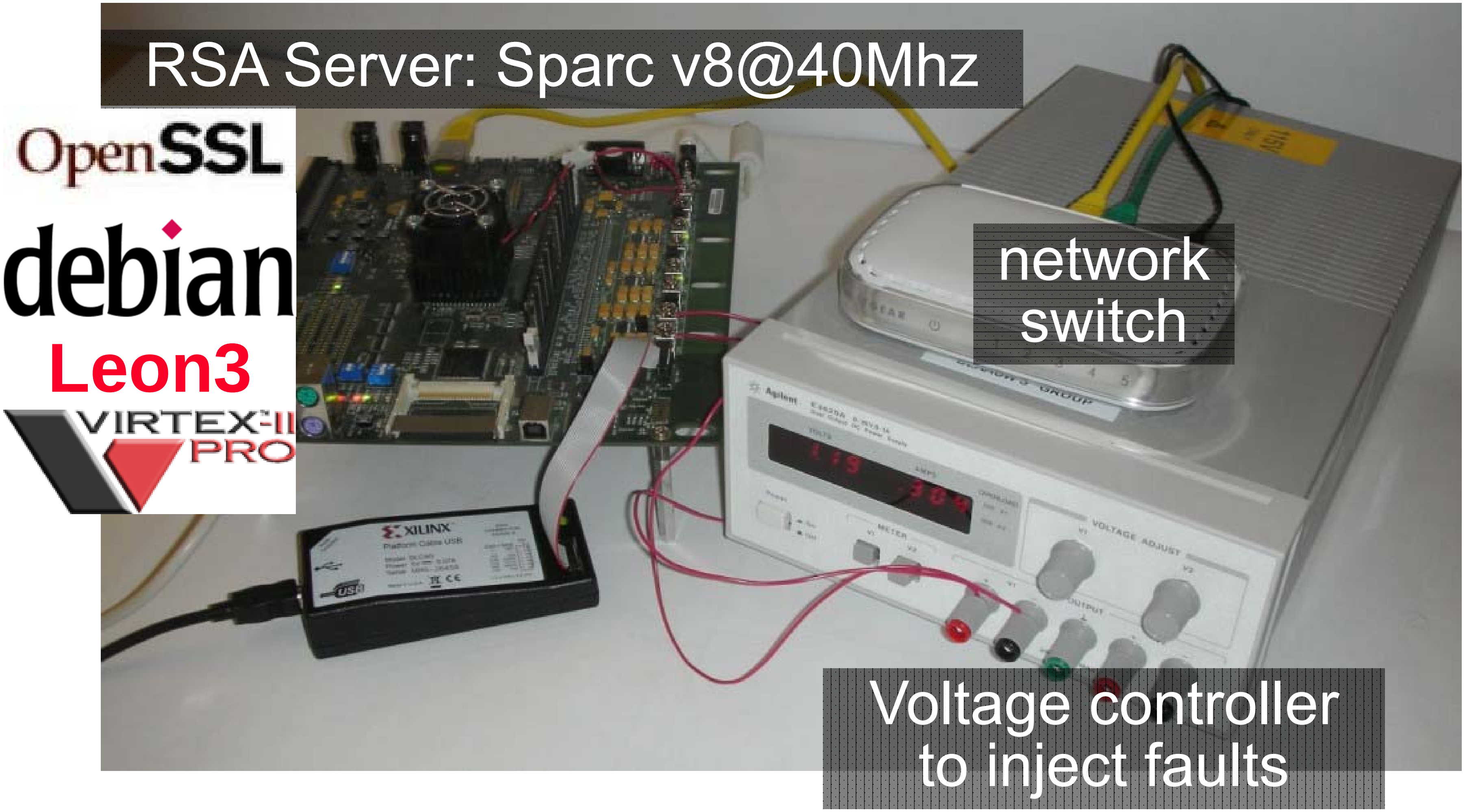


Faulty Server:

$\hat{s} \mathrel{!=} m^d \bmod n$



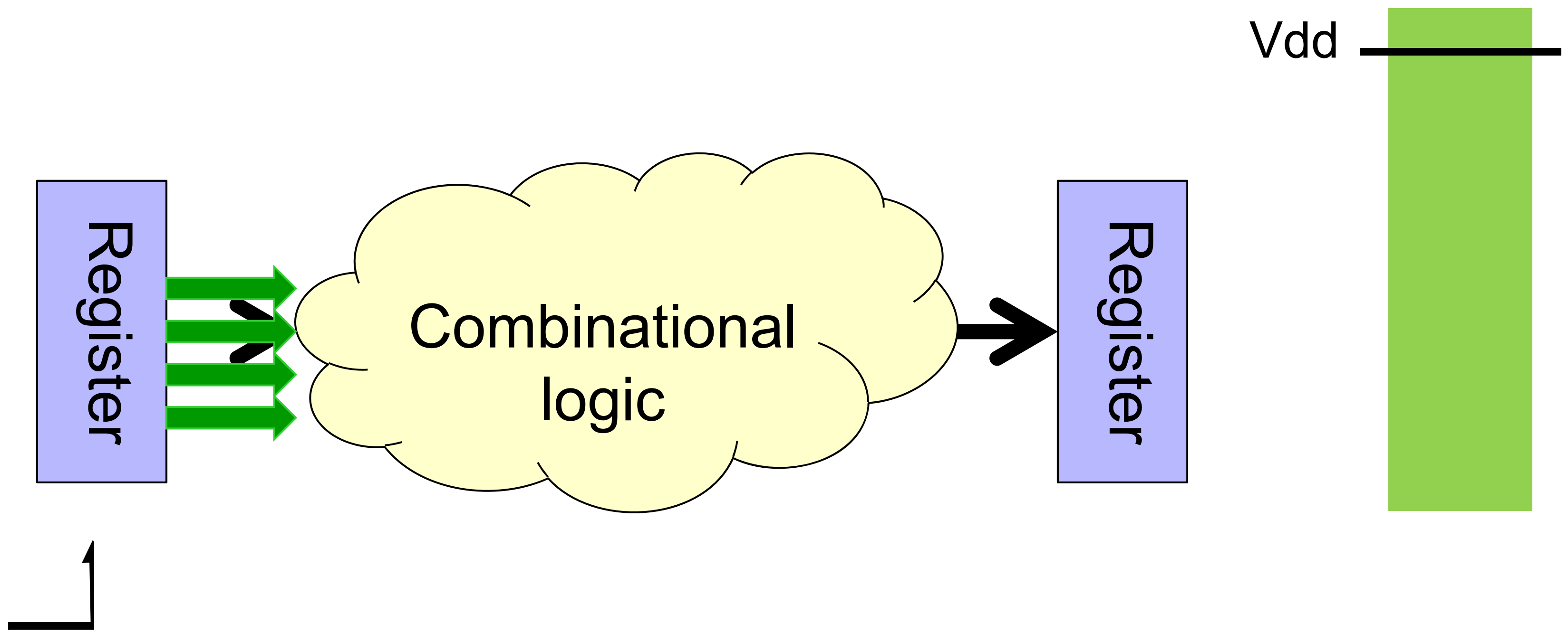
Our Experimental Platform



How I Transported It To Black Hat

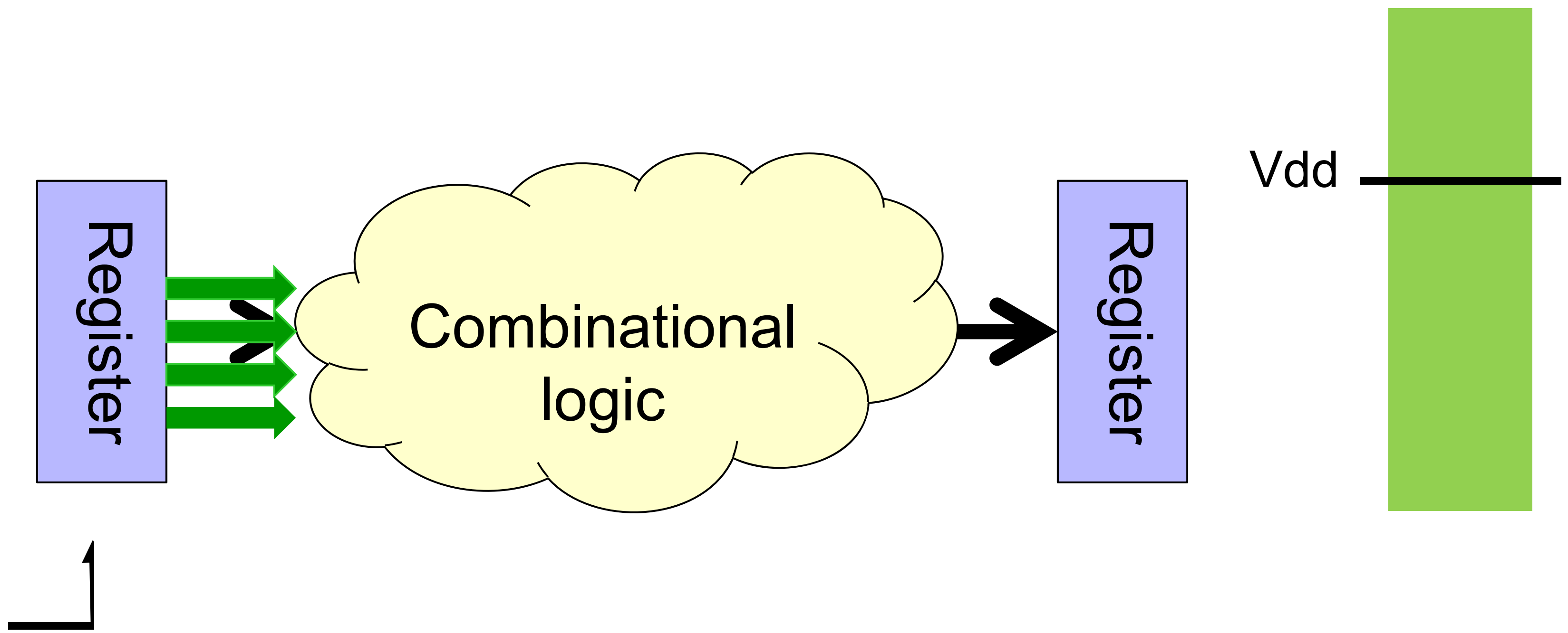
Correct Sequential Circuit

How can we inject faults in a digital system?



Faulty Sequential Circuit

How can we inject faults in a digital system?



The lower the voltage, the less energy the electric signals in traversing the logic cloud

Outline

- Cryptography Introduction
- RSA authentication
- Attacks to RSA authentication
- OpenSSL implementation
- Private key extraction
- Fault injection
- Conclusions

Computing: $s=m^d \bmod n$

Fixed Window Exponentiation, used in OpenSSL

The algorithm partitions the exponent into windows:

$$d = \boxed{1101}\boxed{1011}\boxed{0001}\ldots\boxed{1101}\boxed{1001}\boxed{0101}$$

```
s=1
for each window:
  for each bit in window: //4times
    s = (s * s) mod n
  s = (s * m^d[window]) mod n
return s
```

Computing: $s=m^d \bmod n$

$$d=214=\underbrace{1101}_{\text{window 1}}\ \underbrace{0110}_{\text{window 2}}$$

```
s=1
for each window:
  for each bit in window: //4times
    s = (s * s) mod n
  s = (s * m^d[window]) mod n
return s
```

$s=1$

$s=(\cdots(m^{1101})^2)^2)^2)^2$

$s=m^{1101}$

$s=(\cdots(m^{1101})^2)^2)^2)^2)m^{0110}$

$$s=(\cdots(m^{1101})^2)^2)^2)^2)m^{0110}$$

Faulty Signature: $\hat{s} != m^d \bmod n$

$$d=214=\underbrace{1101}_{\text{window 1}}\ \underbrace{0110}_{\text{window 2}}$$

```
s=1
for each window:
  for each bit in window: //4times
    s = (s * s) mod n
  s = (s * m^d[window]) mod n
return s
```

$s=1$

$\hat{s} = (\cdots(m^{1101})^{2})^{2}) \pm 2^{f})^{2})^{2}$

$s = m^{1101}$

$\hat{s} = (\cdots(m^{1101})^{2})^{2}) \pm 2^{f})^{2})^{2})m^{0110}$

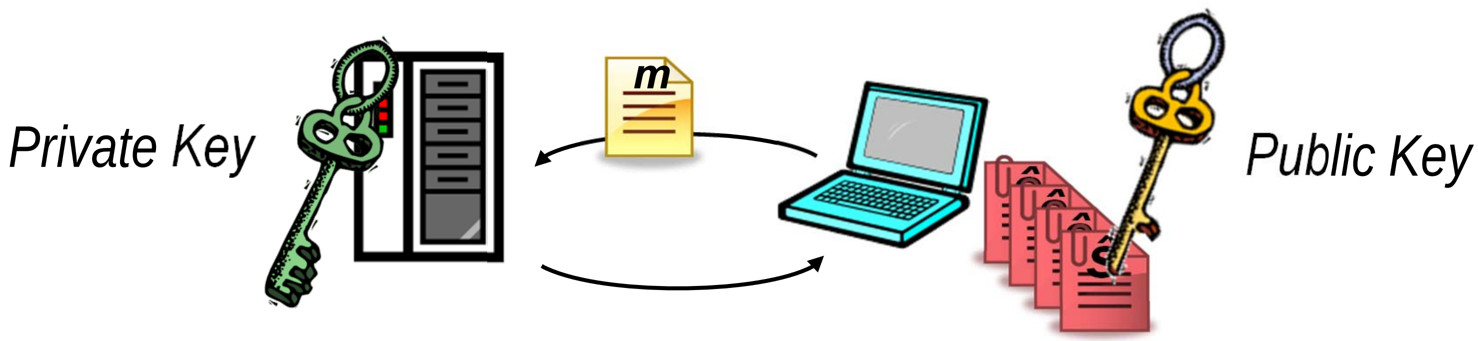
$$\hat{s} = (\cdots(m^{1101})^{2})^{2})\pm 2^{f})^{2})^{2})m^{0110}$$

Outline

- Cryptography Introduction
- RSA authentication
- Attacks to RSA authentication
- OpenSSL implementation
- Private key extraction
- Fault injection
- Conclusions

Retrieving the Private Key

- The attacker collects the faulty signatures



faulty signatures

Reconstructing the Signature

The private key is recovered one window at the time, guessing where and when the fault hits

$$d = d_k\ X_{k-1}\ X\ \ldots\ X$$

Which multiplication?

$$\hat{s} = (\cdots(m^{\boxed{d_k}})^{64})m^{\boxed{d_{k-1}}})^2)^2)^2 \boxed{\pm 2^f})^2)^2)^2)\, m^{d_{k-2}})^{64} \ldots m^{d_0}$$

Already known

Value?

Which bit?

For each window value to be guessed and signature we test:

- 1024 error positions
- 2 possible error values (0→1 or 1→0)
- 6 squaring iterations

Offline Analysis

With a sufficient number of corrupted signatures the attack is polynomial w.r.t. the length of the key

Already known

Value?

Which multiplication?

$$\hat{s} = (\cdots(m^{\boxed{d_k}})^{64})m^{\boxed{d_{k-1}}})^{2})^{2})^{2}\boxed{\pm 2^{f}})^{2})^{2})^{2})\ m^{d_{k-2}})^{64} \ldots m^{d_0}$$

Which bit?

- Performing this check takes about 100 seconds
- In the worst case we have 2^6 values to check!
- If no faulty signature can confirm the value of the guess, we must extend the window

The Whole Truth About This Search

But, how do we deal with the unknowns d_0, d_1, d_{k-2}?

$$\hat{s} = (\cdots(m^{\boxed{d_k}})^{64})m^{\boxed{d_{k-1}}}\overbrace{)^2)^2)^2 \boxed{\pm 2^f})^2)^2)^2}) m^{d_{k-2}})^{64} \ldots m^{d_0}$$

We can reduce the red part to **m** by:

1. Multiplying both sides by $(((m^{d_k})^{64})\, m^{d_{k-1}})^{64}$
2. Raising both sides to the **e** power

$$(\hat{s}\,(((m^{d_k})^{64})\, m^{\mathbf{d_{k-1}}})^{64})^{e} =$$

$$= (\cdots(m^{d_k})^{64})m^{\mathbf{d_{k-1}}}\overbrace{)^2)^2)^2 \pm 2^{\mathbf{f}})^2)^2)^2}{}^{e}\, m$$

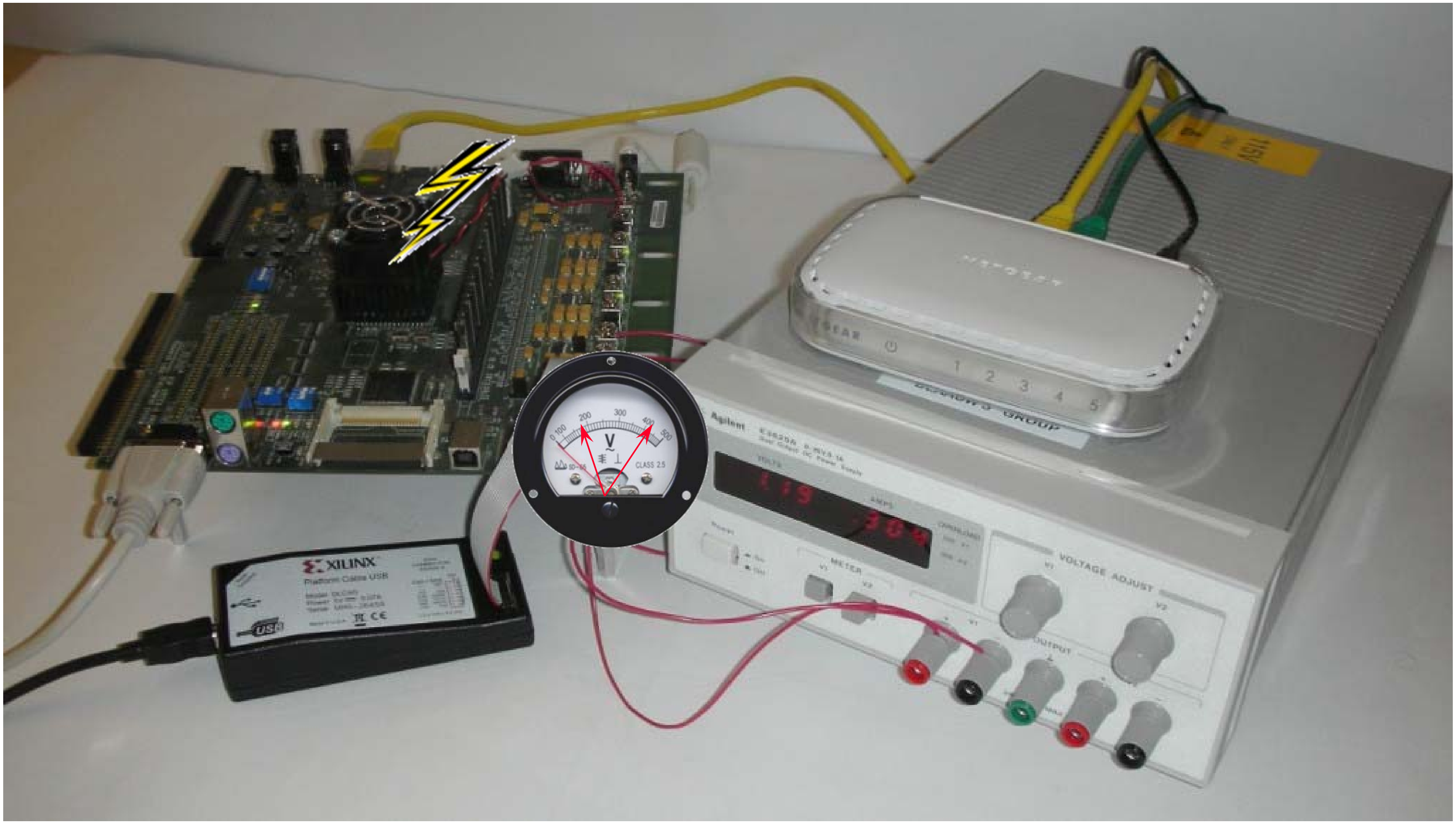
Now everything is in terms of known or to-be-guessed terms!

Outline

- Cryptography Introduction
- RSA authentication
- Attacks to RSA authentication
- OpenSSL implementation
- Private key extraction
- Fault injection
- Conclusions

Our Setup

- Faults manifests on the multiplier of the server's CPU



Fault Injection Mechanisms

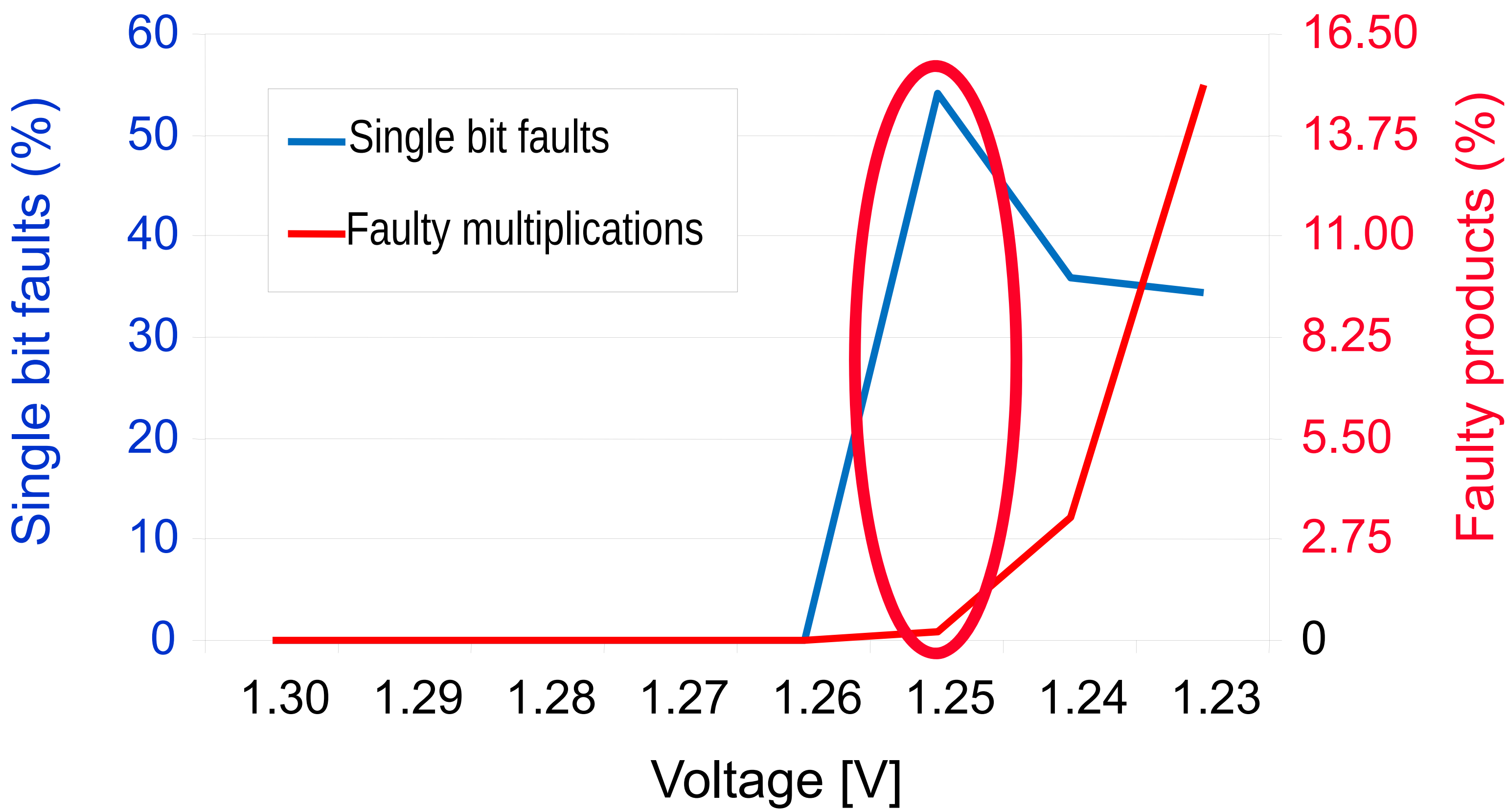
How to make hardware fail:

- ✓ Lower voltage causes signals to slow down, thus missing the deadline imposed by the system clock
- High temperatures increase signal propagation delays
- Over-clocking shortens the allowed time for traversing the logic cloud
- Natural particles cause internal signals to change value, causing errors

All these sources of errors can be controlled to tune the fault injection rate and target some units in the design

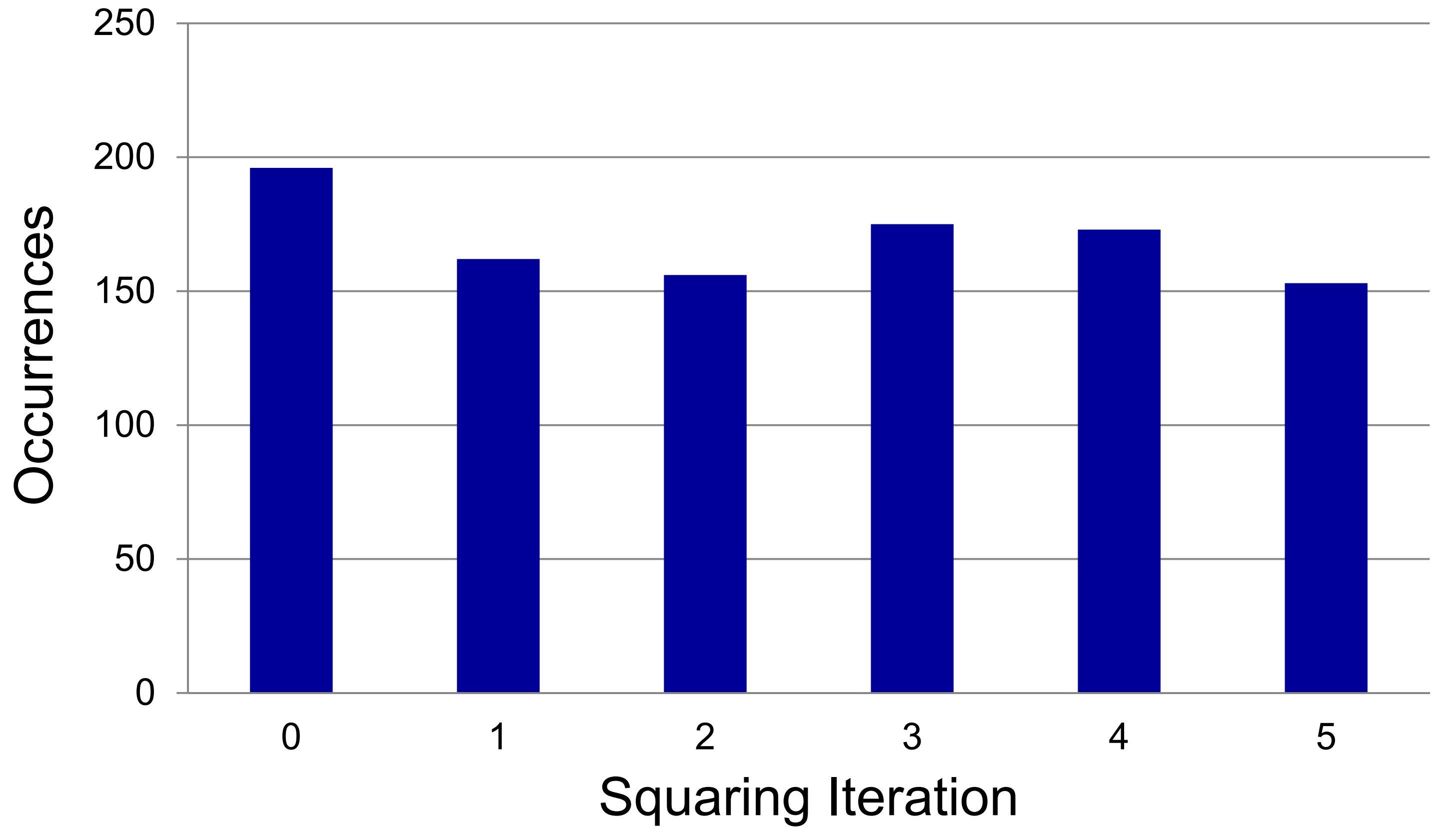
Fault Injection

A corrupted signature leaks data if only one multiplication is corrupted by a single bit flip



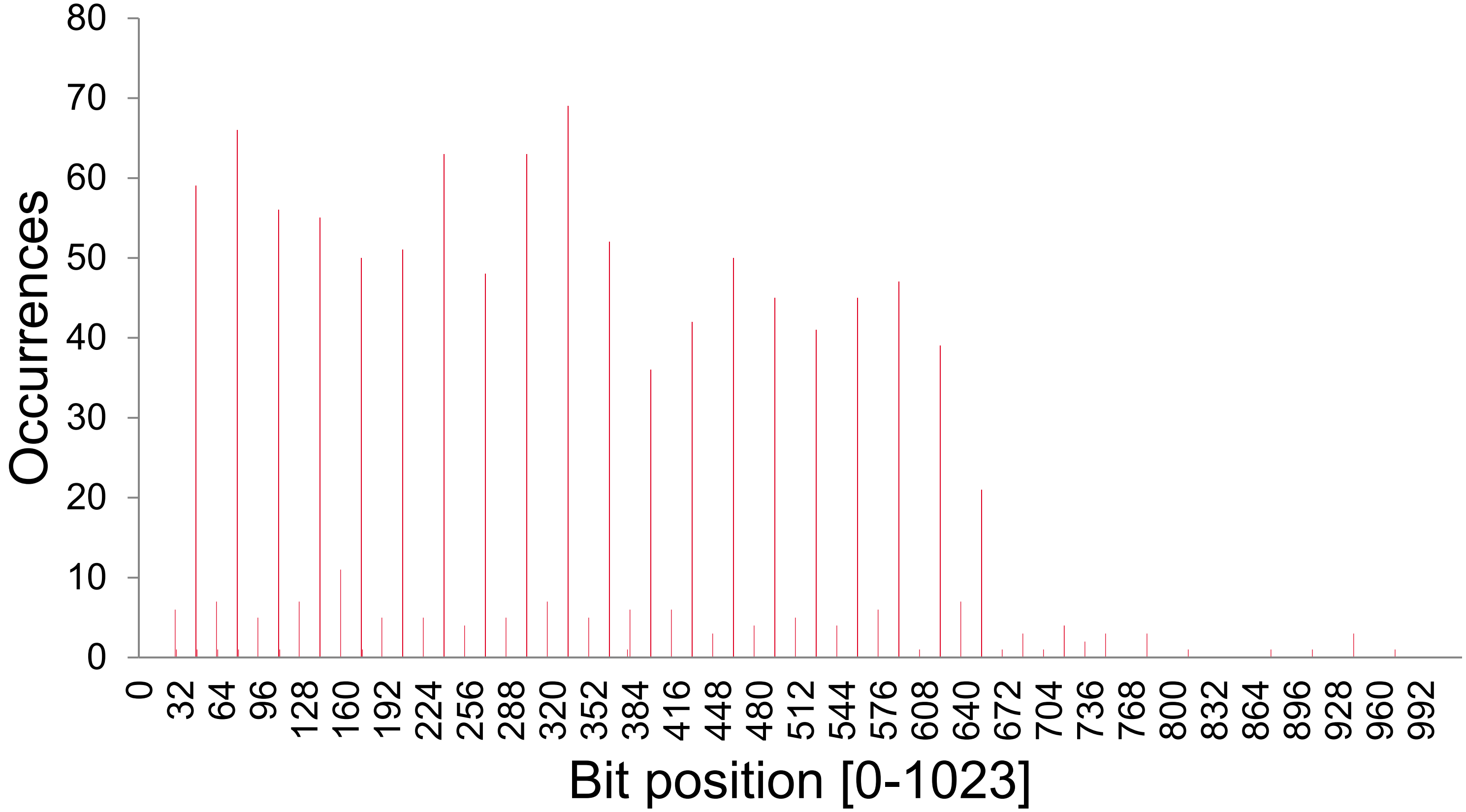
Fault Distribution

The attacked algorithm uses 6-bit windows: any of the 6 squaring iterations has the same probability to fail



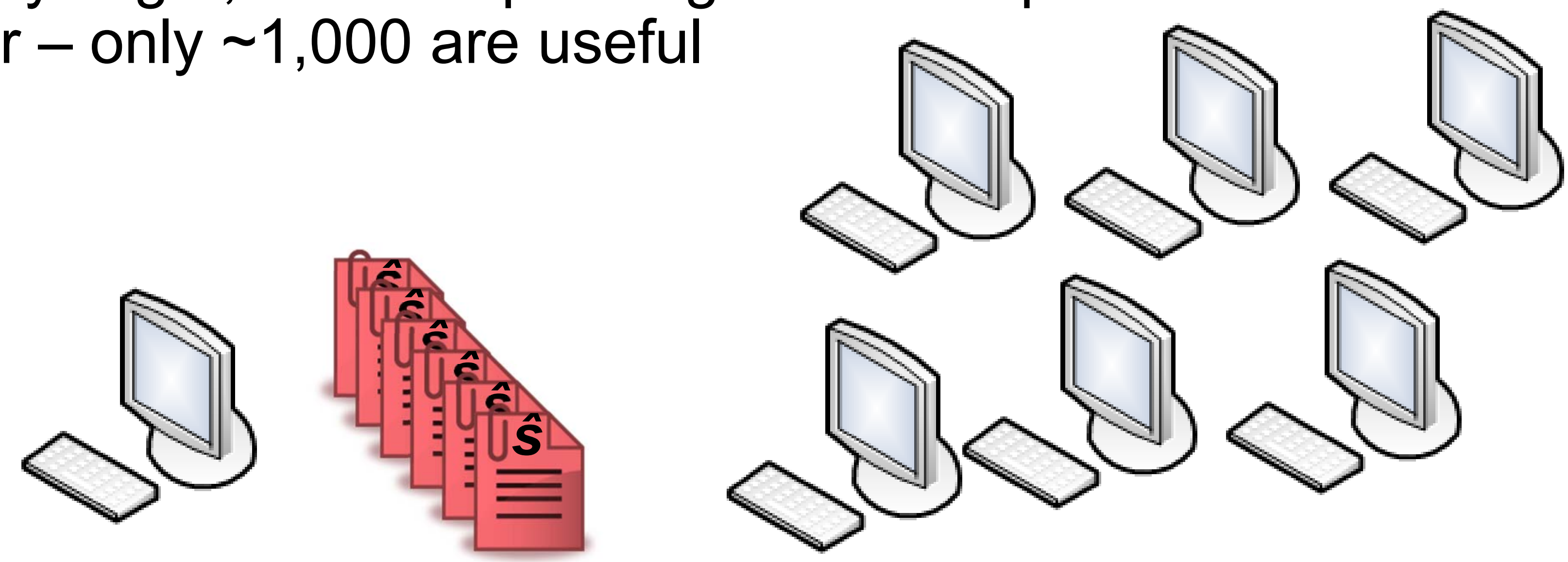
Fault Position

The faults affects some bit positions more than others, proving that the critical path of the multiplier is failing



Offline Analysis

- In practice 40 bit positions typically affected by faults
  → the computation time is reduced to 2.5 seconds
- Analyzing 8,800 corrupted signatures requires 1 CPU-year – only ~1,000 are useful



- Signatures can be checked in parallel
- Using 80 servers the 1024-bit key was retrieved in 104 hours

Physical Attack

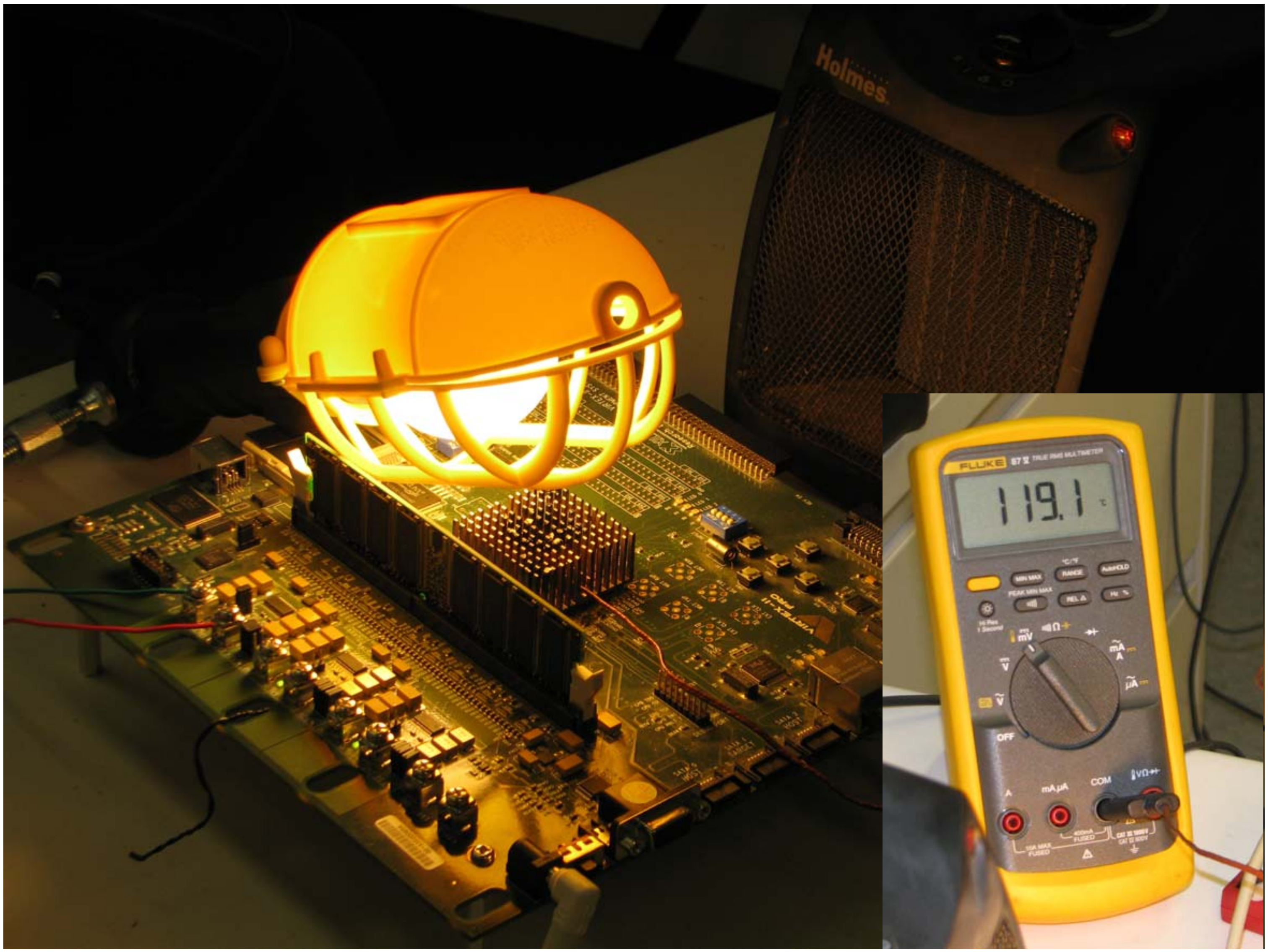
Fault Injection Mechanisms

How to make hardware fail:

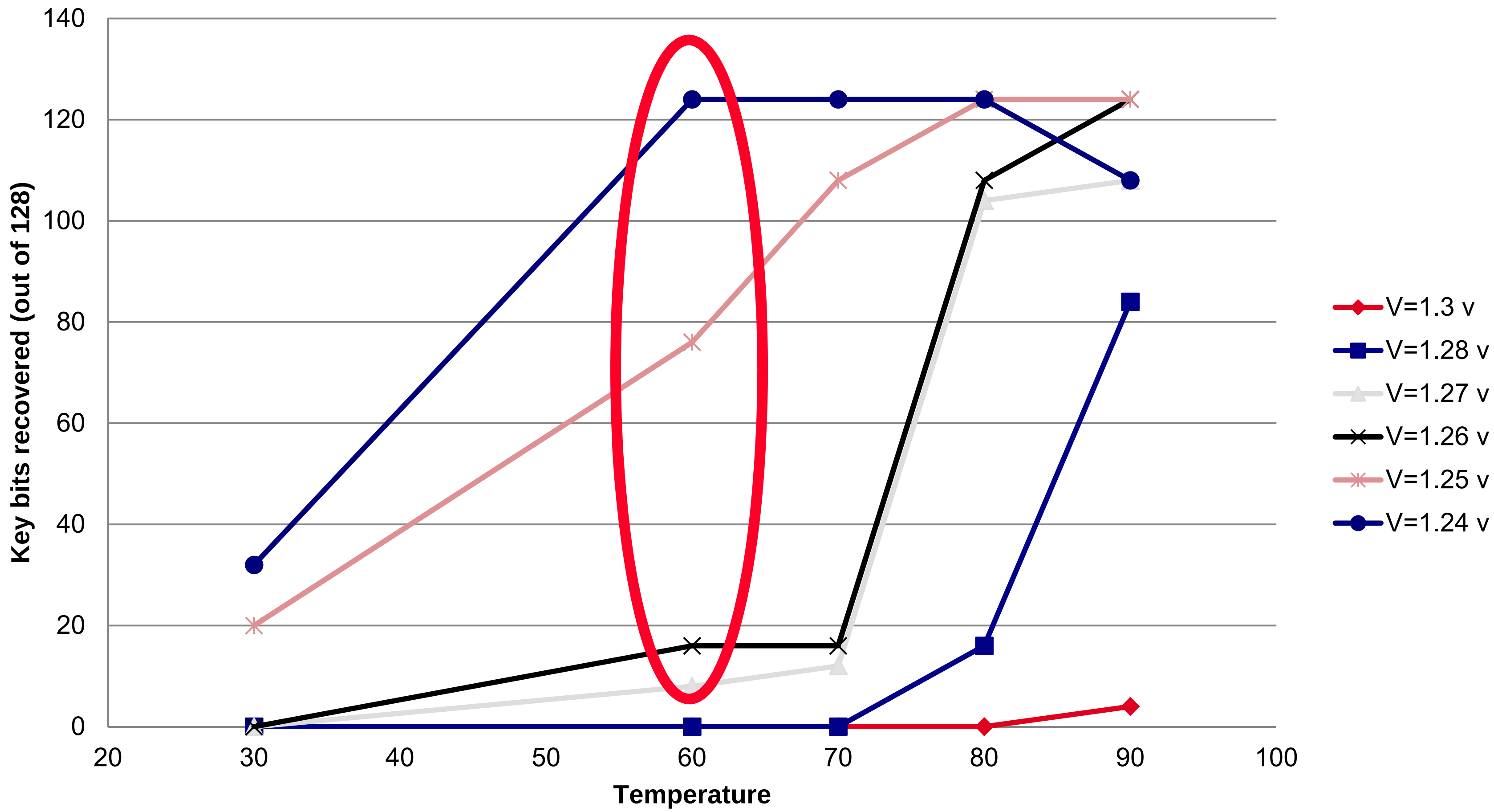
- Lower voltage causes signals to slow down, thus missing the deadline imposed by the system clock
- ✓ High temperatures increase signal propagation delays
- Over-clocking shortens the allowed time for traversing the logic cloud
- Natural particles cause internal signals to change value, causing errors

Course project by:
Armin Alaghi, William Arthur, Prateek Tandon

Temperature-Induced Faults

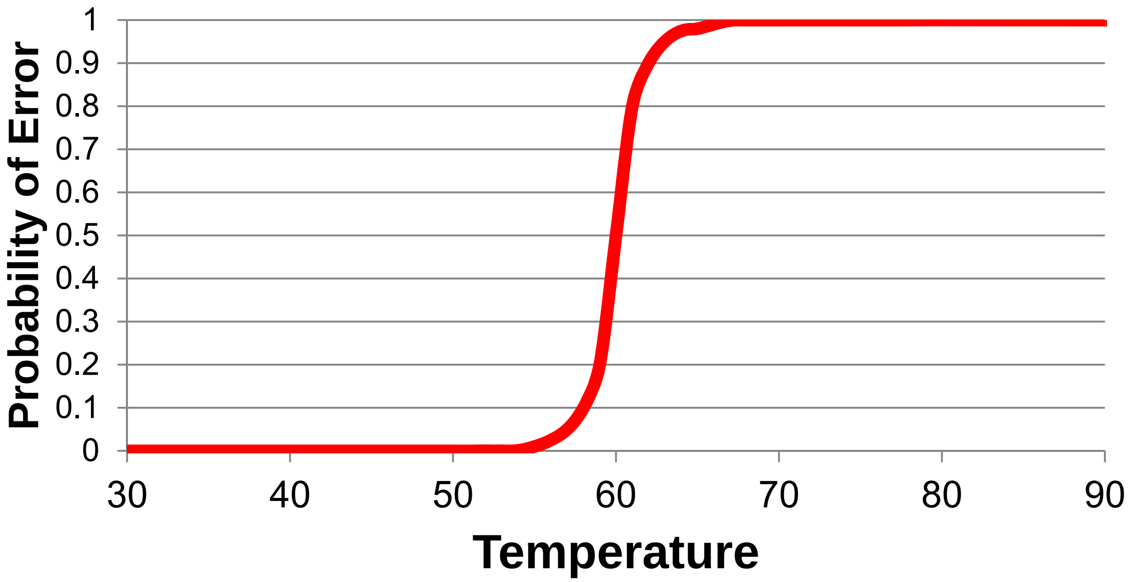


#Key Bits Revealed (128-bit RSA)



Challenges

- Controlling temperature
- Thermal runaway when attacking 1024-bit
  - Solution: Use heat sink, moderate temperature
  - Runtime is an issue



- Extracted 30% of the private key (283/1000 corrupted, 91 useful messages)

Conclusions

- Transient faults can leak vital private key data
- Fault-based attack devised for OpenSSL 0.9.8i 's Fixed Window Exponentiation algorithm
- Attack demonstrated on a complete physical Leon3 SPARC system
- **Software fix using "blind"ing**
  available in OpenSSL to protect against timing attacks
  – make sure to deploy
- Published: "Fault-based Attack of RSA Authentication" - DATE 2010

Take Away for the Security Conscious

- Always keep OpenSSL and all cryptographic libraries updated
- Always make sure that the HW is working in proper conditions
  - Do not overclock
  - Cool the system properly
  - Avoid power fluctuations

- A computer system operating outside its nominal conditions might not fail dramatically: *however, silent data corruptions are even more dangerous*