

Side Channel Attacks

Beyond the classical cryptanalysis

Anestis Bechtsoudis – bechtsoudis.com



Patras Linux User Group

Contents

- ◆ 1. Introduction
- ◆ 2. Cryptanalysis
- ◆ 3. Attack Scenarios
- ◆ 4. Countermeasures
- ◆ 5. Conclusions

1.

Introduction

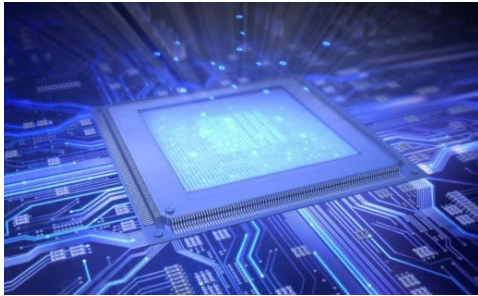


Introduction

- ❖ Information intensive society – imperative need for security
- ❖ Cryptographic systems - purpose:
 - Prevent unauthorized access
 - Warranty authenticity & integrity
 - Protect Privacy

Introduction

Cryptographic implementations



Dedicated Hardware



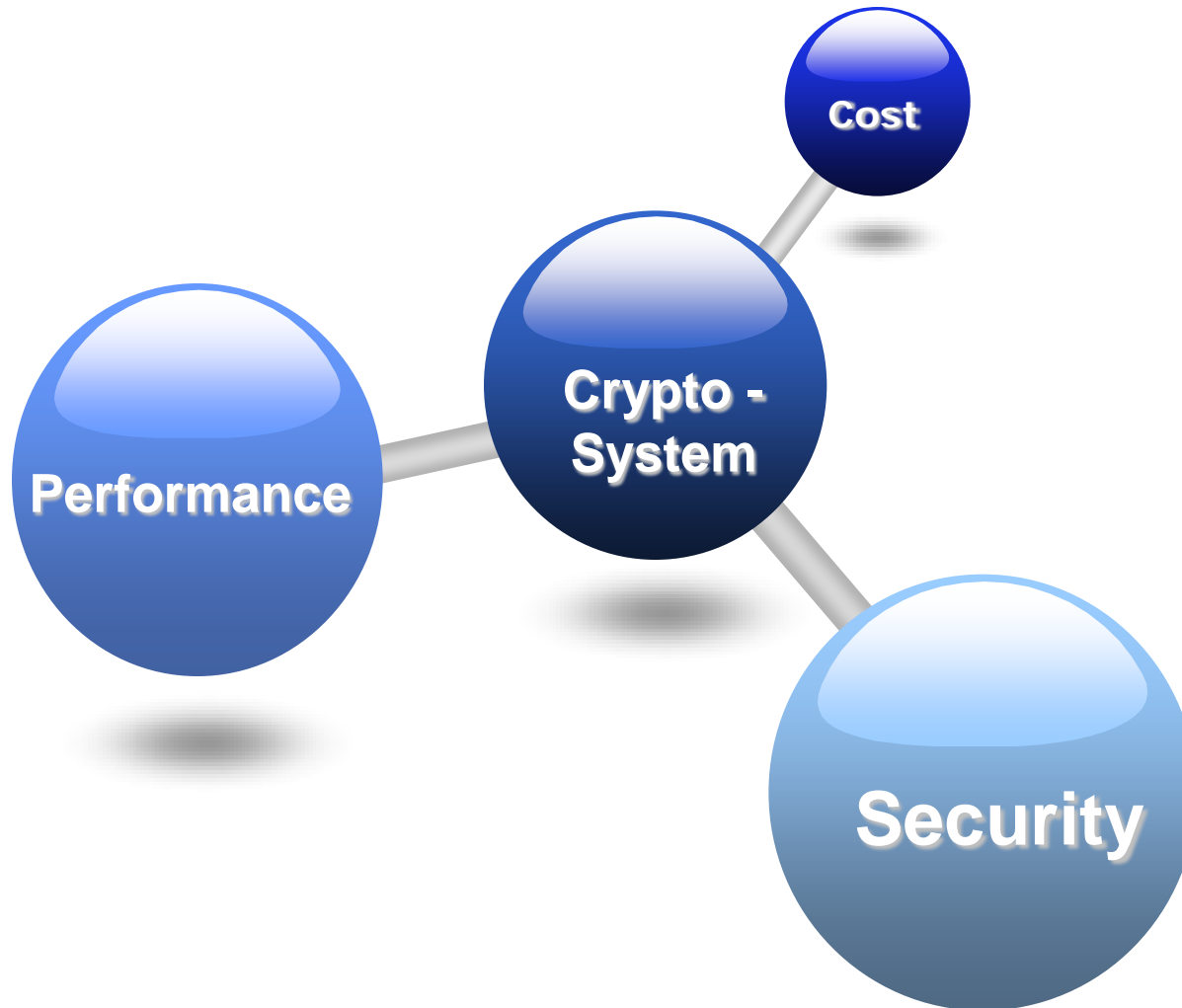
Software



Smart Cards

❖ Application specific security level

Introduction

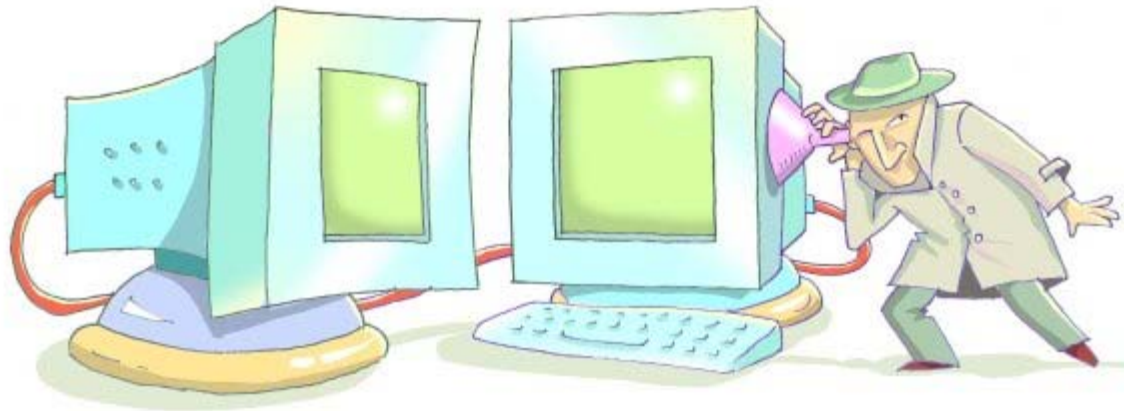


Introduction

- ❖ Cryptographic algorithm classes:
 - Secret/Symmetric key ciphers (AES, DES, IDEA, Cast, Camellia, XTEA)
 - Public/Assymmetric key ciphers (RSA, ECC)
 - Hash functions (SHA2/1, MD5, whirlpool)
- ❖ Despite security level – modern ciphers designed to achieve high throughputs with low resources

2.

Cryptanalysis



Cryptanalysis

- ❖ *Definition:* The study of techniques to reveal the secret parameters of a security system
- ❖ Classical approach:
 - Weaknesses in the algorithm – mathematical model
 - Attacks based on: ciphertext-only, known plaintext, chosen plaintext/ciphertext ...
 - Black box approach of the system

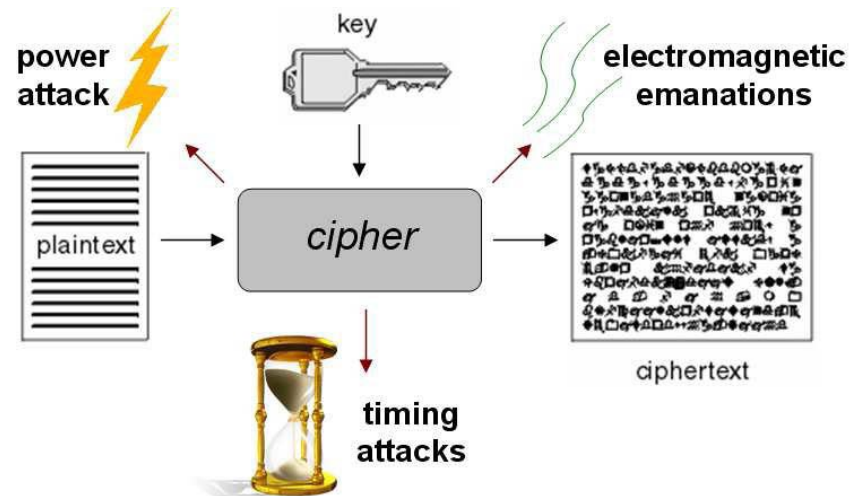
Cryptanalysis

- ❖ The cryptographic primitive is actually implemented in physical devices
- ❖ Modern approach:
 - The attacker knows much more for the running device
 - Side channel leakage

Cryptanalysis

❖ Side channel attacks: Any observable information emitted as a byproduct of the physical implementation of the cryptosystem

- Timing attacks
- Power analysis
- Fault injection
- Cache observation
- Noise analysis
- Electromagnetic analysis



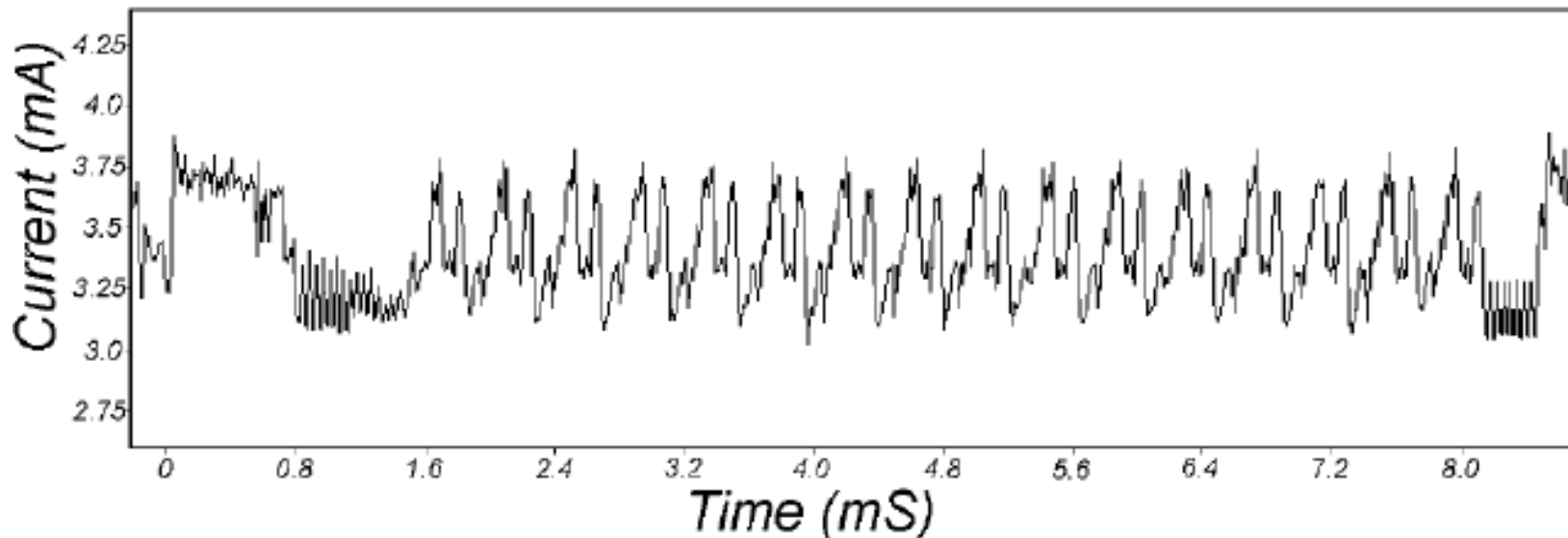
3.

Attack Scenarios



Cryptanalysis

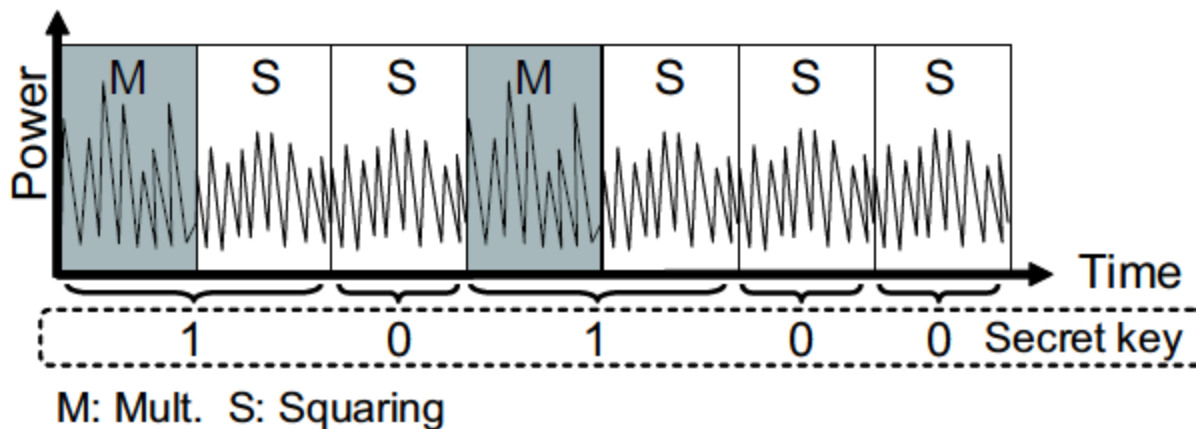
- ❖ Simple Power Analysis
- ❖ Shunt resistor in power line – measure drop voltage / resistor = current



Cryptanalysis

Input:	$X, N,$ $E = (e_{k-1}, \dots, e_1, e_0)_2$
Output:	$Z = X^E \bmod N$

```
1 : Z := 1;
2 : for i = k - 1 downto 0
3 :   Z := Z * Z mod N;    - squaring
4 :   if (ei = 1) then
5 :     Z := Z * X mod N;  - multiplication
6 :   end if
7 : end for
```



Cryptanalysis

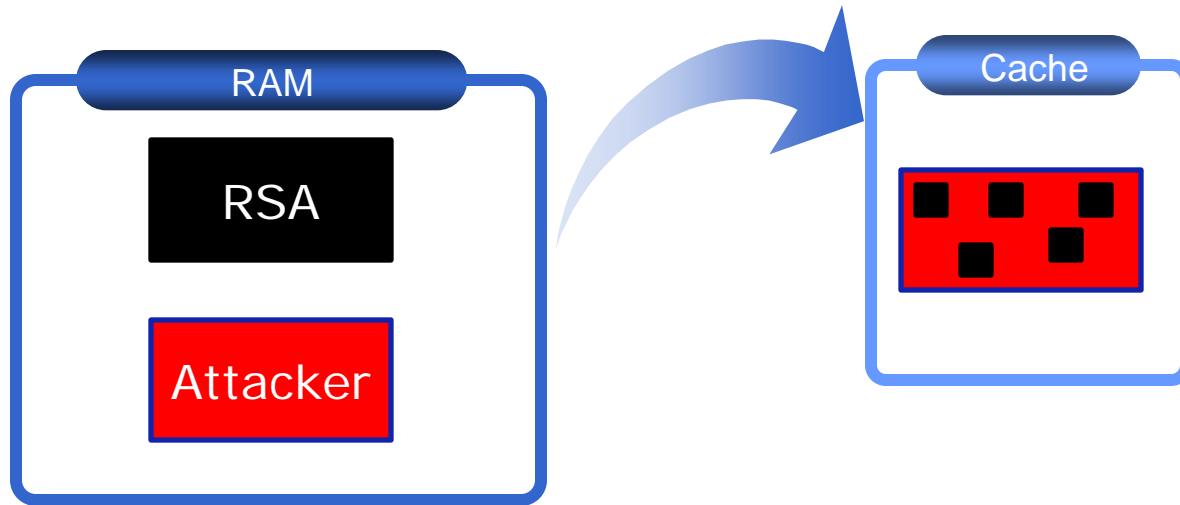
Attack on OpenSSL implementation of RSA algorithm in a SMT CPU

- ❖ RSA Core operation: module exponentiation implemented with series of $\wedge 2$ and $*$
- ❖ The encryption key is divided into segments
- ❖ For each $*$, a multiplier is selected from pre-computed constants stored in a LUT
- ❖ Segment of KEY is used to index the LUT

Cryptanalysis

- ❖ Attacker manages to run simultaneously
- ❖ Attack process sequentially and repeatedly accesses an array, thus loading data to occupy all cache lines
- ❖ At the same time he measures the delay for each access to detect cache misses (ex. rdtsc timer in intel x86)
- ❖ Victim's cache accesses evict attacker's data, enabling detection from the attacker

Cryptanalysis



- ❖ The attacker can identify which table entry is accessed -> the index used -> segment of the key



Cryptanalysis

Every day more and more
sophisticated & hybrid attacks

4.

Countermeasures



Countermeasures

❖ Ideal approach:

- Mathematical model taking into account all side channel characteristics
- Design crypto systems basing on this model

❖ 100% Impossible - Difficulties:

- Large number of parameters
- Different type of traces

Countermeasures

- ❖ Software Solutions:
 - Constant execution paths
 - Avoid conditional branches
 - Hashing values before using them
- ❖ Creative coding
- ❖ Performance penalties

Countermeasures

- ❖ Hardware Solutions:
 - Power balancing
 - Dummy operations – Add delays
 - Balancing Hamming Weights
- ❖ Performance penalties
- ❖ Increased power consumption

5.

Conclusions



CONCLUSION

Conclusions

- ❖ Must take under great consideration the side channel leakage
- ❖ Impossible to model the attacks – too many different attacks – too many parameters
- ❖ Limit the threat as much as possible

Conclusions

Side Channel Attacks highlight
the need for **co-working** of
software, hardware, algorithm & protocol
designers

Questions?

