Side-channel attacks in cryptography

IMDEA Software Institute

Introduction

Cryptography has become an essential element in our lives. Many tasks that are part of our daily routine would not be possible without it: performing secure bank transactions through the internet, confidential communications, web services authentication...

It is crucial to formally analyze the cryptographic primitives that are developed and used to gain confidence about their security and trustability. The standard way of analyzing these primitives is to define strong security requirements for them and prove that if a primitive did not satisfy these requirements, there would exist an *efficient* algorithm breaking a *hard problem*. For example, the security of two of the most widely used encryption algorithms: *RSA* and *ElGamal*, can be reduced to breaking two well-known and studied problems: *computing roots over composite order groups* and *the discrete logarithm*, respectively. It is believed that there exists no efficient algorithm to solve these problems, which is serves as evidence of the hardness of the mentioned encryption schemes.

However, well-founded theoretic primitives need to be implemented in practice and thus, the previous analysis is not an absolute guarantee of the security of the schemes. Implementations are error prone, engineers can make mistakes and unexpected vulnerabilities might appear when implementing a cryptographic primitive. Furthermore, even when these implementations are tested, checked and validated could still be vulnerable to another class of exploits: *side-channel attacks*.

Side-channel attacks have been successfully used for breaking actual cryptographic implementations. These attacks do not exploit weaknesses in the implementation nor the underlying mathematical basis of the primitives. Instead, these attacks are based on physical information gained from the interaction with the primitive. Specifically, they use secret-dependent variations of non-functional properties such as timing, power consumption, electromagnetic leaks, traffic volume...

Timing attacks

Timing attacks constitute a very important class of side-channel attacks, since they can be performed remotely. Actual implementations of cryptographic algorithms usually perform computations in non-constant time. This is due to efficiency optimizations. When such operations involve secret information, there exists a potential leakage from these timing variations.

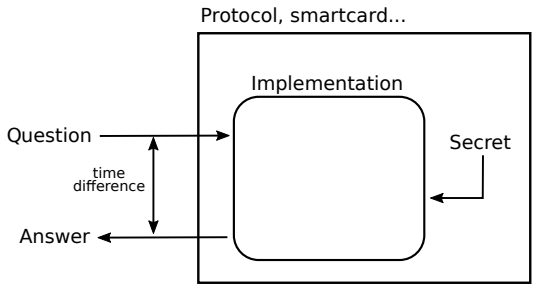

Figure 1: The timing attack principle

The first timing attacks against actual cryptosystems were demonstrated in 1996 by Kocher [Koc96]. Later, in 2005, Brumley and Boneh [BB03] demonstrated remote timing attacks, extracting private keys of an OpenSSL-based web service, followed by other works [BB07, BT11].

As an example, consider the algorithm for modular exponentiation described in Figure 2. This algorithm is frequently used in cryptosystems such as RSA and ElGamal (when implemented over the integers).

```
modexp(b, e, m) :
    r = 1
    for i = |e| − 1 downto 0 do
        r = square(r)
        r = mod(r, m)
        if e_i = 1 then
            r = mul(b, r)
            r = mod(r, m)
    return r
```

Figure 2: Pseudocode for the square-and-multiply modular exponentiation. The function takes as input integers b, e, m and returns $b^e \pmod{n}$.

This algorithm iterates over the bits of the exponent e. In particular, during decryption, this exponent corresponds to the secret key of the system. Note that whenever there is a 1 in e, the algorithm performs a (slow) multiplication, which is not performed otherwise. Measuring the total execution time of this function can reveal the total number of multiplications that were performed and thus, the number of 1's in the exponent, i.e., the exponent's *Hamming weight*. Note that this only is one bit of leaked information, but this simple example illustrates the idea of how extra information (that should not be possible to extract) can be obtained from timing measurements.

Padding oracle attack

Many cryptographic primitives operate with fixed-size blocks of information. Therefore, variable-length messages need to be expanded in order to make them compatible with the block-size. This additional part that is added to the messages is called *padding*.

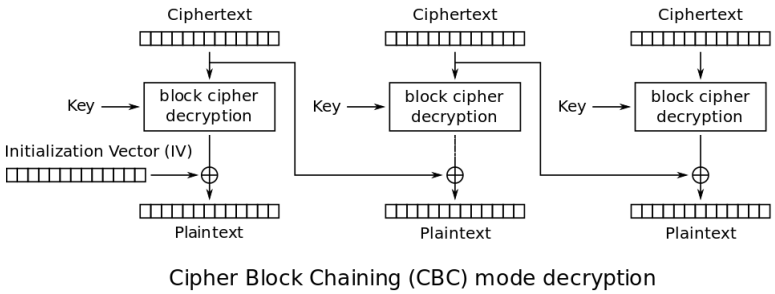

Figure 3: Cipher Block Chaining (CBC) mode decryption (image from Wikipedia)

For example, the Advanced Encryption Standard (AES) is a block cipher that operates with 128-bit blocks. When executed in the CBC mode of operation, if the bit-length of the message is not a multiple of 128, a padding is concatenated to it. Some implementations provide information about the

validity of the padding during decryption and this is a potential source of side-channel information. A padding oracle attack uses this padding validation in order to derive extra information about the primitive. Several works demonstrate how this attacks can be performed in practice [BFK+12, PY04].

The problem

Study, design and implement different techniques to attack very performant cryptographic constructions using side-channel approaches and study what kind of counter-measures could prevent such attacks.

Why the problem can be treated mathematically

From the attackers perspective, finding side-channels for cryptographic primitives is a challenging task since it requires to exploit subtle details about their implementation. Most of this exploits have a mathematical basis and require a detailed analysis of the underlying algorithms. On the other hand, other attacks leverage techniques from machine learning such as *neural networks*, *decision trees*...

From the security point of view, there does not exist a formal model that characterizes the physical properties of the cryptographic systems. Mathematics can be utilized to build a precise model, which would be very helpful for the analysis and prevention of side-channel attacks.

Work plan

The students will be given some highly performance implementations of cryptographic primitives such as RSA or ElGamal (relying on performance optimizations like the Montgomery modular multiplication). The work plan is divided into several tasks:

- Develop algorithms to remotely attack the implemented primitives by using *timing attacks*/*padding oracle attacks* and fully recover the secret key. This task can be performed by exploring ideas from works like the one by Dhem et. al. [DKL+00]
- Perform a comparison between the different proposed and implemented attacks, benchmarking the number of measurements they require, their computational cost...
- Study and propose different counter-measures that can be taken into account in order to prevent these side-channel attacks, without fully compromising the performance/functionality of the primitive (following ideas from works like [Doy16]). If there is time, implement this counter-measures and test their effectiveness.
- Conclusions and exposition. Structure and summarize the obtained results, comparing them to other works from the literature.

References

[BB03] David Brumley and Dan Boneh. Remote timing attacks are practical. In *Proceedings of the 12th Conference on USENIX Security Symposium - Volume 12*, SSYM'03, pages 1–1, Berkeley, CA, USA, 2003. USENIX Association.

[BB07] Andrew Bortz and Dan Boneh. Exposing private information by timing web applications. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 621–628, New York, NY, USA, 2007. ACM.

[BFK+12] Romain Bardou, Riccardo Focardi, Yusuke Kawamoto, Lorenzo Simionato, Graham Steel, and Joe-Kai Tsay. Efficient padding oracle attacks on cryptographic hardware. Cryptology ePrint Archive, Report 2012/417, 2012.

[BT11] Billy Bob Brumley and Nicola Tuveri. Remote timing attacks are still practical. In Vijay Atluri and Claudia Diaz, editors, *Computer Security – ESORICS 2011*, pages 355–371, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[DKL+00] Jean-François Dhem, François Koeune, Philippe-Alexandre Leroux, Patrick Mestré, Jean-Jacques Quisquater, and Jean-Louis Willems. A practical implementation of the timing attack. In Jean-Jacques Quisquater and Bruce Schneier, editors, *Smart Card Research and Applications*, pages 167–182, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[Doy16] Goran Doychev. Tools for the evaluation and choice of countermeasures against side-channel attacks. thesis (doctoral). In *E.T.S. de Ingenieros Informticos (UPM)2*, 2016.

[Koc96] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. pages 104–113, 1996.

[PY04] Kenneth G. Paterson and Arnold Yau. Padding oracle attacks on the iso cbc mode encryption standard. In Tatsuaki Okamoto, editor, *Topics in Cryptology – CT-RSA 2004*, pages 305–323, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.