ICR – Practical Work #3

Fault Attacks against RSA-CRT

Gugger Joël [1]

May 26, 2016

[1] joel.gugger@master.hes-so.ch

Abstract

A way to accelerate the RSA signature procedure consists in exploiting the fact that one knows the two primes p and q, as it is a private-key operation, and to use the Chinese Remainder Theorem (CRT).

The goal of this practical work consists in implementing a fast RSA signature procedure that exploits the CRT and to study the security of such an implementation at the light of fault attacks. This practical work can be implemented either in C, C++, Java or Python, with the big-numbers arithmetic library of your choice.

Contents

Chapter 1

RSA-CRT

This part is dedicated to the implementation of a RSA-CRT fast signing procedure. One can assume that 1024-bit RSA keys are used and that the digest formatting operation is performed elsewhere.

1.1 Questions

1.1.1 How have you tested that your routines are properly working?

1.1.2 What is the gain in terms of speed that you obtain when using RSA-CRT with respect to a standard RSA signature generation procedure?

1.1.3 What are the values that one could pre-compute and store besides n and d, in order to speed up as much as possible the signature generation procedure?

1.2 Implementation

1.2.1 RSA key generation routine

1.2.2 Standard RSA signature and verification routines

1.2.3 Fast RSA signature procedure

Chapter 2

Boneh-DeMillo-Lipton Attack

In 1997, Boneh, DeMillo and Lipton have demonstrated that if a fault is induced during one of the two partial signature computation steps, that erroneous signature can be exploited in order to factor the public modulus.

2.1 Mathematical description

Task 2. *Describe in mathematical terms how the Boneh-DeMillo-Lipton fault attack against RSA-CRT is working.*

2.2 Questions

2.2.1 In practice, how is it possible to induce faults in cryptographic implementations?

2.2.2 Is this attack working on a non-deterministic padding scheme?

2.3 Simulating Boneh-DeMillo-Lipton attack

Task 3. *Write a program simulating Boneh-DeMillo-Lipton attack that allows to factor $n = pq$ in a very efficient way.*

Chapter 3

Implementing Shamir's Trick

Several countermeasures have been proposed to defend against Boneh-DeMillo-Lipton attack. In this part, we will study and implement the one that is known as *Shamir's trick*. This technique essentially works as follows: the partial signatures are computed modulo rp and rq, where r is a small (i.e., 32-bit) random integer, instead of working modulo p and q, respectively.

3.1 Mathematical description

Task 4. *Describe in mathematical terms how Shamir's trick works.*

3.2 Implementation

Task 5. *Implement an RSA-CRT routine protected against Boneh-DeMillo-Lipton attack thanks to Shamir's trick.*

Abstract

The sources of the project are available on GitHub at the following address:
https://github.com/GuggerJoel/Crypto-ICR-lab003