Dobbertin Challenge 2014

1 Challenge

Cold Boot Attacks

It is known for many years that secret cryptographic keys can be recovered from DRAM[1]. One of the main scenarios is a locked computer running a cryptographic software. The idea of the so-called "cold boot attack" is that even after a reboot of the computer, the DRAM data can be retrieved. Experiments showed that in the short time the DRAMs are not supplied with power during the reboot, the bits decay very slowly at a predictable rate. Depending on the "ground state" and the duration of the power outage, bit flipping probabilities can be estimated. If the ground state is `0`, it can be observed that a couple of `1`-bits flip to `0`, but most of the `0`-bits stay `0`. After the DRAM regains its power supply, the erroneous bits are refreshed regularly and stay stable. A special purpose operating system that uses only negligible amounts of DRAM (and thus preserves most of the old DRAM bits) is booted and streams the memory content to an external device.

Attack on NTRU

In this challenge, we want to use "cold boot" side channel information to attack NTRU. NTRU is a public key encryption system that uses a binary polynomial f as its secret key and is described below. Due to implementation reasons of the decryption routine, in our case the polynomial f was stored in memory *three times*. Applying the above described attack, it was possible to restore all three copies of f. Due to the short power outage, some of the bits flipped during the attack. It is estimated that the probability of a `1` flipping to the ground state `0` is 30 %. On the other hand a `0` becomes a `1` with only 0.1%. In your attack, you may also assume that all bits flipped independently.

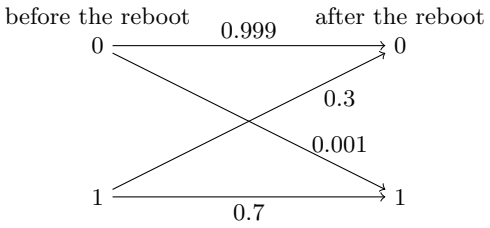

Fig. 1. bit flipping probabilities

Your Task

The attacked implementation uses the following NTRU parameters:

$$N = 1087,\ p = 3,\ q = 2048,\ d_f = d_r = 121,\ d_g = 362.$$

You can find the used public key and the three erroneous versions of f in the attached file. You will also find an encryption of a secret (ASCII encoded) message. Your task is to compute this message using the given side channel information! Please send your solution to

`dc2014@rub.de`

[1] Dynamic Random Access Memory

2 NTRU

There are several versions of NTRUEncrypt. In this section we will describe the version that is attacked in the challenge. NTRU parameters are N, p, q, d_f, d_r and d_g.

Some Math

NTRU operations are performed in the ring $R = \mathbb{Z}_q[X]/(X^N - 1)$. Each element of R can thus be described as a polynomial of maximal degree $N-1$ with coefficients in $\mathbb{Z}_q = \{0, 1, \ldots, q-1\}$. The addition in the ring is the coefficient-wise (modulo q) addition of the polynomials. The ring multiplication is the multiplication of the polynomials followed by a reduction modulo $X^N - 1$. The multiplicative inverse of a ring element a is a ring element a^{-1} such that $a \cdot a^{-1} = 1$. Example with $N = 5, q = 4$ (all coefficient-wise operations are performed modulo 4):

$$a = X^4 + 3X^2 + 1, \quad b = 3X^3 + X^2$$
$$\text{Addition: } a + b = X^4 + 3X^3 + 1$$
$$\text{Multiplication: } a \cdot b = (X^4 + 3X^2 + 1) \cdot (3X^3 + X^2) = 3X^4 + 3X^3 + X + 1 \bmod X^5 - 1$$
$$\text{Inverse: } a^{-1} = X^4 + 3X^3 + X^2, \text{since } a \cdot a^{-1} = 1 \bmod X^5 - 1$$

Remark: Some NTRU operations are performed modulo p (instead of modulo q), with the same rules as above.

Key Generation

In the key generation, two ring elements f, g are chosen uniformly at random with the following restrictions:

- g is chosen such that d_g coefficients are 1 and (the remaining) $N - d_g$ are 0.
- f is chosen such that d_f coefficients are 1 and $N - d_f$ are 0 and such that the inverses $f_q := f^{-1} \bmod q$ (with coefficients mod q) and $f_p := f^{-1} \bmod p$ (coefficients mod p) exist.

Afterwards, a polynomial $h := p \cdot f^{-1} \cdot g \pmod q$ is computed and is used as the public key. The private key is (only) f.

A toy example ($N = 7, q = 16, p = 3, d_f = 5, d_g = 2$): The polynomials $g = X + 1$ (two 1-coefficients) and $f = X^6 + X^4 + X^3 + X^2 + 1$ (five 1-coefficients) are chosen. f is both invertible modulo q and modulo p. The inverses are $f_q = 6X^6 + 7X^5 + 7X^4 + 7X^3 + 6X^2 + 6X + 6$ and $f_p = 2X^6 + 2X^2 + 2X + 2$.

The corresponding public key is $h = p \cdot f_q \cdot g = 7X^6 + 10X^5 + 10X^4 + 7X^3 + 4X^2 + 4X + 4$.

Encryption

Again a uniformly random polynomial r is chosen such that d_r coefficients are 1 and the remaining $N - d_r$ coefficients are 0. The message is encoded as a polynomial m with binary coefficients. The ciphertext is finally computed as $c := h \cdot r + m \bmod q$. Example: $r = X^4 + 1$ and $m = X^3 + X^2$ are chosen. Thus $c = 11X^6 + 14X^5 + 14X^4 + 15X^3 + 15X^2 + 14X + 11$.

Decryption

The message is decrypted as $m := (f \cdot c \bmod q) \cdot f_p \bmod p$. Thus, first $f \cdot c \bmod q$ is computed, reduced modulo p and finally multiplied with the inverse of f modulo p. Example:

- $f \cdot c \bmod q = 2X^6 + 5X^5 + 4X^4 + X^3 + 2X^2 + 4X + 4$
- $(f \cdot c \bmod q) \bmod p = 2X^6 + 2X^5 + X^4 + X^3 + 2X^2 + X + 1$
- $(f \cdot c \bmod q) \cdot f_p \bmod p = X^3 + X^2$