RUHR UNIVERSITÄT BOCHUM

RUB

Lehrstuhl für Kryptologie und IT-Sicherheit
Prof. Dr. Alexander May
Elena Kirshanova

Hausübungen zur Vorlesung

Kryptanalyse I

SS 2015

Blatt 5 / 9. Juli 2015

Abgabe bis: 16. Juli 12:00 Uhr, Kasten NA/02

Aufgabe 1 (15 Punkte)**:**

Determining RSA-secret key bit-by-bit

In this exercise, you apply the Heninger-Shacham Algorithm to CRT-RSA secret key. As usual, $N = pq, ed = 1 \bmod \phi(N), d_p = d \mod (p-1), d_q = d \bmod (q-1)$. You know the public parameters (N, e), for simplicity we set $e = 3$. Your goal is to determine (p, q, d, d_p, d_q) by learning their values bitwise. For integer $x = (x[n], \ldots, x[1], x[0])$, $x[i]$ is the i-th LSB (with $x[0]$ - LSB, $x[n]$ -MSB). Recall the following RSA-congruences

$$\begin{aligned} N &= p \cdot q \\ ed &= 1 \bmod \phi(n) \\ ed_p &= 1 \bmod (p-1) \\ ed_q &= 1 \bmod (q-1) \end{aligned} \iff \begin{aligned} N &= p \cdot q \\ ed &= k\phi(n) + 1 \\ ed_p &= k_p(p-1) + 1 \\ ed_q &= k_q(q-1) + 1, \end{aligned}$$

for some $k, k_p, k_q \in \mathbb{Z}$.

1. Show that for our setting, there are at most two possible values for k.
2. Thus, we assume that we know k (via enumeration or simple guess). Show how to determine the values for k_q, k_p. For prime e, there should be only two possibilities for each value. Make use of the above congruences.
3. Moreover, you already know $p[0], q[0]$ - the LSBs of the RSA-primes. What are they? Combine your knowledge of k, k_p, k_q to obtain $d[0], d_p[0], d_q[0]$. Write down the corresponding equations.
4. An this stage, you know $(p[0], q[0], d[0], d_p[0], d_q[0])$. Using the above congruences, write down the system of equations for $(p[1], q[1], d[1], d_p[1], d_q[1])$. Overall, you have 4 equations and 5 unknowns. How many candidate-solutions do you have for $(p[1], q[1], d[1], d_p[1], d_q[1])$?
5. Try out your algorithm on $(N, e) = (2173, 3)$. Determine k, k_p, k_q (without factoring N) and then $(p[i], q[i], d[i], d_p[i], d_q[i])$ for $i = 0, 1$.

Turn over!

Aufgabe 2 (15 Punkte)**:**
Programming assignment: Kocher's timing attack on RSA.

You are given access to the RSA Signature oracle that outputs Sign(m) on message m, where Sign(m) = m^d and the exponentiation is implemented via the Repeated-Squaring algorithm, the description of which is provided in file 'RSAKocher.cpp'. This time you also know the d in clear, your goal is to show how to get a good approximation of d.

To achieve this, you will implement the Kocher's attack ([1]) (or, if you like, a simplified version of it). As a start, you will need to measure the time needed to generate a signature. In the function

```
void RSASign (mpz_t m, mpz_t sign, double *t);
```

the last parameter t outputs the elapsed time in seconds.

With this, you generate many messages M_is and try to determine the second LSB of d: $d[1]$, (you know $d[0]$ via the parity). If ($d[1] == 1$), one more modular squaring of each M_i took place during the signature generation and there should be a correlation between the time to generate Sign(M_i) and $M_i^2 \bmod N$. So you should measure the time of modular squaring.

Once the $d[1]$ is determined (with some probability), you proceed with the next bit by first producing the partial signature evaluation (i.e. $M^{2d[1]+d[0]} \bmod N$) and repeating the correlation measurements.

Note that the actual d is sparse, so the generation of one signature is quite fast and you can allow many M_i's. Thus, if your partial signature evaluation takes on average longer than the generation of the complete signature, some guessed bits are wrong. You might need to repeat the algorithm several times to get a 'good' distribution on M_is.

In your solution to the exercise, specify how you measured the correlation between the two outputs (two timings) providing the output of your program (e.g. expected values, variance, etc.) For instance, you can compute the Pearson product-moment correlation coefficient. In other words, you should show the validity of your approach.

Literatur

[1] P. Kocher, Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems, 1996