RSA Attacks and Implementation Failures

Ahto Buldas   Aleksandr Lenin

Nov 11, 2019

Small Modulus and Factoring

Let $n = pq$ be the RSA modulus and $p < q$ are prime numbers.

Trial Division runs in time $O(\sqrt{n})$.

Pollard's rho algorithm: $O(\sqrt[4]{n})$.

Lenstra's elliptic curve factorization:

$$e^{(1+o(1))\sqrt{\ln n \ln \ln n}}$$

General Number Field Sieve (GNFS):

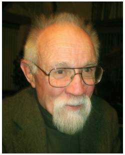
$$e^{\left(\sqrt[3]{\frac{64}{9}}+o(1)\right)\sqrt[3]{\ln n(\ln \ln n)^2}}$$

Common Modulus and Simmons Attack

A has e_A and d_A such that $e_A d_A \equiv 1 \pmod{\varphi(n)}$.

B has e_B and d_B such that $e_B d_B \equiv 1 \pmod{\varphi(n)}$.

Let $\gcd(e_A, e_B) = 1$, which is a very likely case



A and B are sent ciphertexts $y_A = m^{e_A} \bmod n$ and $y_B = m^{e_B} \bmod n$ of the same message m.

Simmons attack:

Find $\alpha, \beta \in \mathbb{Z}$ so that $\alpha e_A + \beta e_B = 1$ and $\alpha < 0$, i.e. $\alpha = -|\alpha|$.

Compute $y_A^{-1} \bmod n$ and

$$\left[y_A^{-1}\right]^{|\alpha|} \cdot [y_B]^{\beta} = m^{\alpha e_A} \cdot m^{\beta e_B} = m^{\alpha e_A + \beta e_B} = m \ .$$

Factoring with Square Roots of 1

Suppose we know $b \neq \pm 1$ such that $b^2 \equiv 1 \pmod{n}$ (where $n = pq$).

From $b^2 = 1$ it follows that $(b+1)(b-1) \equiv 0 \pmod{n}$.

As $b \neq \pm 1$, we have that $b + 1 \not\equiv 0 \pmod{n}$ and $b - 1 \not\equiv 0 \pmod{n}$.

As $p|(p+1)(p-1)$ then either $p|(b+1)$ or $p|(b-1)$.

Hence, $\gcd(b+1, n) \in \{p, q\}$ and we can factor n.

Finding Square Roots of 1 from Key-Pairs (e, d)

As $ed \equiv 1 \pmod{\varphi(n)}$, we have $ed - 1 = c \cdot \varphi(n) = 2^s \cdot \lambda$, where $\lambda \in \mathbb{N}$ is odd.

Finding Square Roots:

- Pick random $a \in \{2, \ldots, n-2\}$ so that $\gcd(a, n) = 1$.
- Find the smallest $j > 0$ such that $a^{2^j \lambda} = 1$ [exists, because $\varphi(n) \mid 2^s \lambda$]
- If $a^{2^{j-1}\lambda} \equiv -1 \pmod{n}$, output $a^{2^{j-1}\lambda} \mod n$, otherwise try again.

It can be shown that a non-trivial $\sqrt{1}$ is found with probability $\frac{1}{2}$.

Deterministic procedure discovered in 2004.

Correctness Proof for the Square Root Algorithm

Lemma 1: For any prime numbers $p, q \geq 3$, there exists $t \in \mathbb{N}$, such that $\frac{p-1}{2^t}$ and $\frac{q-1}{2^t}$ are integers and at least one of them is odd. (Obvious)

Lemma 2: For any prime $p \geq 3$, there are $\frac{p-1}{2}$ elements $x \in \mathbb{Z}_p^*$ with $x^{\frac{p-1}{2}} \equiv 1 \pmod{p}$ and $\frac{p-1}{2}$ elements with $x^{\frac{p-1}{2}} \equiv -1 \pmod{p}$.

Proof: Fermat's theorem implies that all $p-1$ elements of $\mathbb{Z}_p^*$ are roots of the polynomial $X^{p-1} - 1$. Hence, $y^2 - 1 \equiv 0 \pmod{p}$ for any $y = x^{\frac{p-1}{2}}$. As $\mathbb{Z}_p$ is a field, we have $y \equiv \pm 1 \pmod{p}$.

Therefore, every $x \in \mathbb{Z}_p^*$ is a root of $X^{\frac{p-1}{2}} - 1$ or a root of $X^{\frac{p-1}{2}} + 1$. As $\mathbb{Z}_p$ is a field, none of these polynomials has more than $\frac{p-1}{2}$ roots, which means that they both have exactly $\frac{p-1}{2}$ roots, because $|\mathbb{Z}_p^*| = p - 1$. $\square$

Theorem: Let $p > q \geq 3$ be primes, $n = pq$, and $ed \equiv 1 \pmod{\varphi(n)}$. There exists $k \in \mathbb{N}$ so that $\frac{ed-1}{2^k} \in \mathbb{N}$ and $x^{\frac{ed-1}{2^k}}$ is a non-trivial $\sqrt{1}$ in $\mathbb{Z}_n$ with probability $\frac{1}{2}$ for random $x \leftarrow \mathbb{Z}_n^*$.

Proof: Let $\sim$ be the equivalence relation between $\mathbb{Z}_n$ and $\mathbb{Z}_p \times \mathbb{Z}_q$ from Chinese remainder theorem, and $\alpha p + \beta q = 1$, where $\alpha, \beta \in \mathbb{Z}$. Then for every $x \in \mathbb{Z}_n$, $x_p \in \mathbb{Z}_p$ and $x_q \in \mathbb{Z}_q$:

$$\begin{aligned} x &\sim (x \bmod p,\ x \bmod q) \\ \beta q x_p + \alpha p x_q \bmod n &\sim (x_p,\ x_q)\ . \end{aligned}$$

Non-trivial $\sqrt{1}$ correspond to pairs $(1, q-1)$ and $(p-1, 1)$.

Let $ed - 1 = c \cdot \varphi(n)$ where $c = 2^m \cdot \ell \in \mathbb{N}$ and ℓ is odd.

Let $ed - 1 = 2^s \lambda$, where λ is odd.

Let $k = t + m + 1$, where $t \in \mathbb{N}$ is chosen according to Lemma 1, which means that $\frac{p-1}{2^t}$ and $\frac{q-1}{2^t}$ are integers. By $p, q \geq 3$ we have $t \geq 1$.

As $2^{2t} \mid \varphi(n)$, we have from $2^s\lambda = ed - 1 = 2^m \ell \cdot \varphi(n)$ that

$$s \geq m + 2t \geq m + t + 1 = k$$

and hence $\frac{ed-1}{2^k} \in \mathbb{N}$. Therefore, $\frac{ed-1}{2^k} = \frac{\varphi(n)\ell}{2^{t+1}} = \frac{(p-1)(q-1)\ell}{2 \cdot 2^t}$ and:

$$x^{\frac{ed-1}{2^k}} \equiv x^{\frac{(p-1)(q-1)\ell}{2\cdot 2^t}} \sim \left(\left(x_p^{\frac{p-1}{2}} \right)^{\ell \frac{q-1}{2^t}}, \left(x_q^{\frac{q-1}{2}} \right)^{\ell \frac{p-1}{2^t}} \right) .$$

As $x_p^{\frac{p-1}{2}}$ and $x_q^{\frac{q-1}{2}}$ are congruent to -1 or 1 with equal probability, and at least one of $\frac{q-1}{2^t}$ and $\frac{p-1}{2^t}$ is odd, the probability that the components of the pair are different (i.e. exactly one is 1), is $\frac{1}{2}$ and hence $x^{\frac{ed-1}{2^k}}$ is a non-trivial $\sqrt{1}$ with probability $\frac{1}{2}$. $\square$

Small e: Hastad Broadcast Attack

Users A, B, C have RSA moduli n_1, n_2, n_3. Say $e = 3$ and the moduli have no common divisors. Say, m is broadcasted to A, B, C. Having the ciphertexts:

$$y_A = m^3 \mod n_1, \qquad y_B = m^3 \mod n_2, \qquad y_C = m^3 \mod n_3 \ ,$$

the attacker uses CRT to find the unique $x \in \mathbb{Z}_{n_1 n_2 n_3}$ such that

$$\begin{cases} x \equiv y_A \pmod{n_1} \\ x \equiv y_B \pmod{n_2} \\ x \equiv y_C \pmod{n_3} \end{cases}$$

As $m < \min\{n_1, n_2, n_3\}$, then $m^3 < n_1 n_2 n_3$, which means that m^3 is also the solution of the congruences and hence $x = m^3$ by the uniqueness of the solution. The attacker just computes $m = \sqrt[3]{x}$

Homomorphity

RSA encryption has the following property:

$$\begin{aligned} \mathsf{E}(m_1 m_2) &= (m_1 m_2)^e \mod n = m_1^e \cdot m_2^e \mod n \\ &= \mathsf{E}(m_1) \cdot \mathsf{E}(m_2) \mod n \ . \end{aligned}$$

For example:

$$\mathsf{E}(2m) = \mathsf{E}(2) \cdot \mathsf{E}(m) \mod n \ ,$$

which means that given the ciphertext $\mathsf{E}(m)$, one can compute the ciphertext $\mathsf{E}(2m)$ without using secret key.

Abusing the Homomorphity

Assume that a server has RSA public key (e, n).

Users send encrypted messages $\mathsf{E}(m)$ to the server, where m is assumed to be odd.

Otherwise (if m is even), the server replies with an error message.

Weakness: By communicating with the server, we can decrypt any ciphertext $\mathsf{E}(m)$.

Abusing the Homomorphity

By sending $\mathsf{E}(m)$ to the server, we learn if m is odd or even.

Compute $\mathsf{E}(2m) = \mathsf{E}(2) \cdot \mathsf{E}(m)$ and send it to the server.

If $m < \frac{n}{2}$, then $2m < n$ and as $2m \mod n$ is even, we get an error message.

If $\frac{n}{2} \leq m < n$, then $n \leq 2m < 2n$ and as $2m \mod n = 2m - n$ is odd, we do not get error messages.

Hence, we learn if $m < \frac{n}{2}$.

Secure Encryption

Semantic security: Ciphertext C must not reveal any information about the plaintext M

The textbook RSA is not semantically secure

Example, encrypting yes/no votes. Given an encrypted vote

$$C = v^e \mod N \ ,$$

an attacker can encrypt both votes and compare the results to C.

Random padding has to be applied before encryption

Bleichenbacher's Attack

The PKCS 1 padding looks like this:

`02 | Random | 00 | Message`

Say a server receives encrypted messages and returns an invalid ciphertext error message if the decrypted message has an incorrect padding

So, sending a random ciphertext C to the server, an attacker will know if the corresponding plaintext has 02 in the beginning

Bleichenbacher showed in 1998 that if an attacker who has access to such a server, can decrypt any ciphertext

Partial Key Exposure

Given an n-bit RSA modulus N, and $n/4$ least significant bits of the secret modulus d, it is easy to compute d

Given an n-bit RSA modulus $N = pq$, and $n/4$ least/most significant bits of p, the modulus N can be factored (Coppersmith 1996)

Timing Attacks

Let $d_n d_{n-1} \ldots d_1 d_0$ be the bit-representation of d. The computation of $M^d \mod N$ is performed as follows:

$z := M$, $C := 1$
For $i = 0 \ldots N-1$ do:
    if $d_i = 1$, then $C := C \cdot z \mod N$
    $z := z^2 \mod N$

The attacker asks the smartcard to compute a large number of exponents, measures the times and reconstructs d using statistical analysis.

Random Faults in Hardware

Smartcard applications of RSA frequently use CRT to speed up $m^d \bmod n$ where $n = pq$:

$$d_p \leftarrow d \bmod p-1 \qquad\qquad d_q \leftarrow d \bmod q-1$$
$$C_p \leftarrow m^{d_p} \bmod p \qquad\qquad C_q \leftarrow m^{d_q} \bmod q$$
$$C \leftarrow \beta q C_p + \alpha p C_q \bmod n \ ,$$

where $\alpha, \beta \in \mathbb{Z}$ are constants such that $\alpha p + \beta q = 1$

Say, an error occurs when computing C_q and $\underline{C}$ is the erroneous version of C. Then

$$\underline{C}^e \equiv m \pmod{p} \qquad \underline{C}^e \not\equiv m \pmod{q}$$

Hence, attacker can compute $\gcd(n, \underline{C}^e - m) = p$ and factorize n

Shor's Factoring Attack on a Quantum Computer

Peter Shor showed in 1994, that quantum computers can find the period of a wide class of functions $f\colon \mathbb{Z} \to \mathbb{Z}_{2^m}$ in time $O(m^2)$.

By the period of f we mean the smallest positive integer λ, such that $f(x+\lambda) = f(x)$ for every x.

1. Random $a \leftarrow \mathbb{Z}_n^*$ is chosen
2. The order $r = \mathrm{ord}_n(a)$ of a is the period of $f(x) = a^x \mod n$ that is found by a quantum computer with probability $\frac{1}{\ln n}$
3. Using a and r a non-trivial $\sqrt{1}$ is found with probability $\frac{1}{2}$

In-Device Private Key Generation

Keys are generated inside smart-cards.

Pros: Improved trust model, compared to other generation options

Cons: Slow due to the small computational power

Current Practice in Prime Number Generation

- *Random candidate* p is chosen
- *Trial division*: It is ensured that p is not divisible by any members of a fixed set Π of small prime numbers
- *Exponential tests*, like the Fermat' test is applied:

  $$a^{p-1} \mod p = 1$$

  for a random $a \leftarrow \{2, 3, \ldots, p-1\}$

The density of n-bit primes is approximately $\frac{1}{n}$.

Division is thousands of times faster than exponentiation.

Trial division eliminates bad candidates fast. Trial division diminishes the average number of exponential tests.

Fast-Prime Methods

Chooses candidates in a way that trial division is not needed.

Choose a first candidate a_0 so that $a_0 \bmod M = 1$, where M is the product of all small primes in Π.

Choose the next candidates a_k by $a_k \leftarrow a_0 + kM$.

Pros: Faster generation of prime numbers

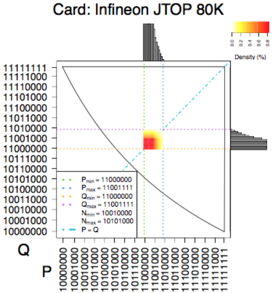
Cons: Lower entropy of prime numbers

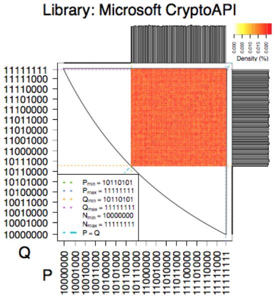
Output Anomalies of Prime Generators

The paper:

"The Million-Key Question – Investigating the Origins of RSA Public Keys" by Svenda, Nemec, Sekan, Kvasnovsky, Formanek, Komarek, and Matyas

analyses the output of several smart-card prime generators. Anomalies in the Infineon's output distribution were discovered.





Formula for the Infineon Primes

The paper:

"The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli." by Nemec, Sys, Svenda, Klinec, and Matyas.

revealed that the Infineon chip's prime numbers are in the form:

$$p = 65537^a \mod M + kM \ ,$$

where M is constant and the same for all chips.

For 2048-bit modulus N, M is the product of the first 126 primes.

All public moduli N satisfy $(65537^c - N) \bmod M = 0$ for some c.

Such c is found in microseconds by the Pohlig-Hellman algorithm

This test was disclosed by the authors in spring 2017, and reached Estonia in August 2017.

Naive Attack

Try all $\mathrm{ord}_M(65537)$ possible a-s and try to find k by Coppersmith's attack

Here, $\mathrm{ord}_M(65537)$ is the order of 65537 in the multiplicative group $\mathbb{Z}_M^*$

Naive search is infeasible: the number of a-s to examine is 2^{254}.

Making the Naive Attack Efficient

Main idea: Use a divisor M' of M, such that $\mathrm{ord}_{M'}(65537)$ is feasible, but still the number of bits in M' is larger than 2048/4 (necessary for the Coppersmith's attack).

Then, the prime numbers are still expressible in the form:

$$p = 65537^{a'} \mod M' + k'M'$$

Authors found optimal M' in terms of the overall attack time by brute force search combined with greedy heuristics.

Impact of the Attack

By using optimal M', the number of possible a-s is 2^{34} for 2048-bit RSA modulus

k is found in 200 ms on a desktop computer by using Coppersmith's algorithm

The total costs by key estimated by authors:

- *30000 EUR* in Amazon cloud
- *1000 EUR* for electricity, without taking hardware into account

Conclusions

Certified$\neq$secure: Though the Infineon chip was certified by Common Criteria, it does not mean it is secure against unknown attacks

Vulnerabilities in soft- and hardware are inevitable

IT-Systems design/management must take potential unknown vulnerabilities into account