CSE 632: Analysis of Algorithms II: Randomized Algorithms **Spring 2024**

Lecture 14: RSA Cryptosystem

Lecturer: Zongchen Chen

1 RSA Cryptosystem

RSA (Rivest–Shamir–Adleman) is a public-key cryptosystem, one of the oldest widely used for secure data transmission.

Algorithm 1 Key generation

1. Choose two random n-bit primes p and q
2. Compute $N = pq$
3. Find e which is relatively prime to $(p-1)(q-1)$ (i.e., $\gcd(e, (p-1)(q-1)) = 1$) ▷ Typically by trying $e = 3, 5, 7, 11, \ldots$ The most commonly chosen value for e is $2^{16} + 1 = 65537$
4. Compute $d \equiv e^{-1} \bmod (p-1)(q-1)$ ▷ Using the extended Euclidean algorithm
5. Public Key: (N, e) ▷ Release them
   Private Key: d ▷ Keep it secrete

Algorithm 2 Encryption

Input: a message represented as an integer m where $0 \le m < N$, public key (N, e)

1. Compute the ciphertext $c \equiv m^e \bmod N$ ▷ Using fast modular exponentiation algorithm
   return c

Algorithm 3 Decryption

Input: a ciphertext c where $0 \le c < N$, private key d

1. Compute the message $m \equiv c^d \bmod N$ ▷ Using fast modular exponentiation algorithm
   return m

Security. Given the public key (N, e), we believe it is computationally intractable to factor $N = pq$ and find the private key d. See also the RSA problem.

Correctness. The correctness of RSA is guaranteed by the following results from number theory.

Theorem 1 (Fermat's Little Theorem)**.** *Let p be a prime number. For all $a \in \mathbb{N}$ such that $a \not\equiv 0 \bmod p$, it holds*

$$a^{p-1} \equiv 1 \bmod p.$$

Lemma 2. *Suppose $N = pq$ where p, q are distinct primes. Let $e \in \mathbb{N}$ be relatively prime to $(p-1)(q-1)$ and $d \equiv e^{-1} \bmod (p-1)(q-1)$. For any $m \in \mathbb{N}$, it holds*

$$m^{ed} \equiv m \bmod N.$$

Remark 3. If $c \equiv m^e \bmod N$ is the ciphertext, then Lemma 2 shows that

$$c^d \equiv (m^e)^d \equiv m^{ed} \equiv m \bmod N.$$

This shows the correctness of RSA.

Proof of Theorem 1. Let $a_i = (ia \bmod p)$ where $i = 1, \ldots, p-1$. We claim that

$$\{a_1, \ldots, a_{p-1}\} = \{1, \ldots, p-1\}$$

as two sets. To see this, notice that $a_i = a_j$ iff $ia \equiv ja \bmod p$, which happens iff $i = j$ since $a \not\equiv 0 \bmod p$ and a^{-1} exists. Thus, $a_i \neq a_j$ for all $i \neq j$. Moreover, $a_i = (ia \bmod p) \neq 0$ since both $i, a \not\equiv 0 \bmod p$. This shows the claim.

Taking product in both sets, we deduce that

$$(p-1)! = \prod_{i=1}^{p-1} a_i \equiv \prod_{i=1}^{p-1} ia \equiv (p-1)! \cdot a^{p-1} \bmod p.$$

Since p is prime, we have $(p-1)! \not\equiv 0 \bmod p$ and hence its inverse exists. It follows that

$$a^{p-1} \equiv 1 \bmod p,$$

as desired. □

Proof of Lemma 2. Notice that $m^{ed} \equiv m \bmod pq$ iff $m^{ed} \equiv m \bmod p$ and $m^{ed} \equiv m \bmod q$ (in other words, $pq \mid m^{ed} - m$ iff $p \mid m^{ed} - m$ and $q \mid m^{ed} - m$). We shall prove $m^{ed} \equiv m \bmod p$ and the argument for q is the same.

If $m \equiv 0 \bmod p$, then we trivially have $m^{ed} \equiv 0 \equiv m \bmod p$. Assume $m \not\equiv 0 \bmod p$. Since $d \equiv e^{-1} \bmod (p-1)(q-1)$, there exists $k \in \mathbb{N}$ such that $ed = k(p-1)(q-1) + 1$. It follows that

$$m^{ed} = m^{k(p-1)(q-1)+1} = (m^{p-1})^{k(q-1)} \cdot m \equiv (1)^{k(q-1)} \cdot m \equiv m \bmod p,$$

where we apply Fermat's little theorem. □

2 Generation of Random Primes

In both fingerprinting and RSA algorithms, we need to generate n-bit prime numbers uniformly at random. Recall that, $\pi(x)$ is the number of prime numbers less than or equal to x.

Theorem 4 (Prime Number Theorem)**.** *For $x \geq 17$, it holds*

$$\pi(x) \geq \frac{x}{\log x}.$$

So, for a random n-bit number x, we have

$$\Pr(x \text{ is prime}) = \frac{\pi(2^n)}{2^n} \geq \frac{1}{n}.$$

Algorithm 4 Generating random primes

1: Choose an n-bit number x u.a.r.
2: Check if x is prime or not
3: **if** Yes **then**
  return x
4: **else**
  return Failure
5: **end if**

When Algorithm 4 outputs a prime number, it is a uniformly random one. We need to run $O(n)$ trials of Algorithm 4 in expectation to find a random prime, and $O(n \log n)$ trials to have failure probability at most $1/\text{poly}(n)$.

3 Primality Test

In Algorithm 4, a crucial step is to determine if a given number x is prime or not. In this section we give two primality tests that can accomplish this goal.

3.1 Fermat test

Recall that Fermat's little theorem states that if x is prime, then $a^{x-1} \equiv 1 \bmod x$ for all $a \in \{1, \ldots, x-1\}$.

Definition 5 (Fermat Witness)**.** We say $a \in \{1, \ldots, x-1\}$ is a *Fermat witness* if $a^{x-1} \not\equiv 1 \bmod x$.

Claim 6.
- *If x is prime, then there is no Fermat witness.*
- *If x is composite, then there exists a Fermat witness.*

Proof. The first claim is due to Fermat's little theorem. For the second claim, observe that for any $a \in \{1, \ldots, x-1\}$ such that $\gcd(a, x) > 1$, we have $a^{x-1} \not\equiv 1 \bmod x$ since $a^{x-1} \bmod x$ is a multiple of $\gcd(a, x)$. $\square$

Algorithm 5 Fermat primality test

Input: an n-bit number x
1: Choose $a \in \{1, \ldots, x-1\}$ u.a.r.
2: Compute $a^{x-1} \bmod x$
3: **if** $a^{x-1} \equiv 1 \bmod x$ **then**
  return Yes
4: **else** ▷ $a^{x-1} \not\equiv 1 \bmod x$ and so a is a Fermat witness
  return No
5: **end if**

Definition 7 (Non-trivial Fermat Witness)**.** We say $a \in \{1, \ldots, x-1\}$ is a *non-trivial* Fermat witness if $a^{x-1} \not\equiv 1 \bmod x$ and $\gcd(a, x) = 1$.

The proof of Claim 6 shows that every composite x has at least one trivial Fermat witness. Those having only trivial Fermat witnesses are called Carmichael numbers.

Definition 8 (Carmichael Number)**.** A composite x is called a *Carmichael number* if it has *no* non-trivial Fermat witness.

Remark 9. Equivalently, a composite x is Carmichael iff $a^x \equiv a \bmod x$ for all a.

Remark 10. There are infinitely many Carmichael numbers, but they are very rare. Smallest Carmichael numbers are 561, 1105, 1729...

Lemma 11. *If x is composite and not Carmichael, then*

$$\Pr\left(a \textit{ is a Fermat witness}\right) \geq \frac{1}{2}.$$

The success probability of Algorithm 5 is summarized in the table below.

	Pr(**return** Yes)	Pr(**return** No)
x is prime	1	0
x is composite & not Carmichael	$\leq \frac{1}{2}$	$\geq \frac{1}{2}$
x is Carmichael	No guarantee (give up)	

Proof of Lemma 11. Let b be a non-trivial Fermat witness, i.e., $b^{x-1} \not\equiv 1 \bmod x$ and $\gcd(b, x) = 1$. Let

$$F = \{f \in \{1, \ldots, x-1\} : f^{x-1} \not\equiv 1 \bmod x\}$$

be the set of Fermat witnesses, and let

$$G = \{1, \ldots, x-1\} \setminus F = \{g \in \{1, \ldots, x-1\} : g^{x-1} \equiv 1 \bmod x\}$$

be the complement. We need to show $|F| \geq |G|$.

Consider a mapping

$$\begin{aligned} \varphi : G &\to F \\ g &\mapsto bg \bmod x. \end{aligned}$$

Notice that,

$$(bg)^{x-1} = b^{x-1} g^{x-1} \equiv b^{x-1} \not\equiv 1 \bmod x.$$

Hence, $\varphi(g) = (bg \bmod x) \in F$; namely, the image of φ is indeed contained in F. We show that φ is injective. Observe that $\varphi(g_1) = \varphi(g_2)$ iff $bg_1 \equiv bg_2 \bmod x$, which happens iff $g_1 = g_2$ since $\gcd(b, x) = 1$ and hence $b^{-1} \bmod x$ exists. Therefore, $\varphi(g_1) \neq \varphi(g_2)$ for all $g_1 \neq g_2$, showing that φ is an injective mapping. Consequently, $|G| \leq |F|$ and the lemma follows. □

3.2 Miller–Rabin test

Definition 12 (Square Root of 1)**.** We say $y \in \{1, \ldots, x-1\}$ is a *square root* of 1 if $y^2 \equiv 1 \bmod x$.

Definition 13 (Non-trivial Square Root of 1)**.** We say $y \in \{1, \ldots, x-1\}$ is a *non-trivial* square root of 1 if $y^2 \equiv 1 \bmod x$ and $y \neq 1, x-1$.

Claim 14. *If x is prime, then there is no non-trivial square root of* 1.

Proof. Notice that $y^2 \equiv 1 \bmod x$ is equivalent to $(y-1)(y+1) \equiv 0 \bmod x$. Since x is prime, this happens iff $y - 1 \equiv 0 \bmod x$ or $y + 1 \equiv 0 \bmod x$. The claim then follows. □

Algorithm 6 Miller–Rabin primality test

Input: an n-bit *odd* number x

Find integers t, u such that $x - 1 = 2^t u$ where $t \geq 1$ and u is odd
Choose $a \in \{1, \ldots, x-1\}$ u.a.r.
for $i = 0$ to t **do**
  Compute $a^{2^i u} \bmod x$ ▷ This can be done recursively
end for
if $a^{2^t u} = a^{x-1} \not\equiv 1 \bmod x$ **then**
  return No ▷ a is a Fermat witness
else if $a^{2^i u} \equiv 1 \bmod x$ and $a^{2^{i-1} u} \not\equiv \pm 1 \bmod x$ for some i **then**
  return No ▷ $a^{2^{i-1} u}$ is a non-trivial square root of 1
else
  return Yes
end if

The success probability of Algorithm 6 is summarized in the table below. The proof is omitted.

	Pr(**return** Yes)	Pr(**return** No)
x is prime	1	0
x is odd composite	$\leq \frac{1}{2}$	$\geq \frac{1}{2}$