Cryptography

Cunsheng Ding, HKUST

COMP3631

Lecture 11: The RSA Public-Key Cipher

Objectives of This Lecture

1. Recall Euler's totient function and Euler's theorem.
2. Introduce the RSA public-key cipher.
3. Discuss its security issues.

History: The RSA public-key cipher was invented in 1977 by Ron Rivest, Adi Shamir, and Len Adleman at MIT.

Euler's Totient Function and Euler's Theorem

They were introduced in the previous lecture and will be needed in this lecture.

Euler's Totient Function $\phi(n)$

$\phi(n)$**:** The number of positive integers less than n that is relative prime to n.That is,

$$\phi(n) = |\{1 \leq i \leq n-1 : \gcd(i, n) = 1\}|.$$

Theorem:

- $\phi(p) = p - 1$ for any prime number p.
- $\phi(pq) = (p-1)(q-1)$ for any two distinct primes p and q.

Proof: It was given in the previous lecture.

Formula for Euler's Totient Function ϕ

Theorem: Let $n = p_1^{e_1} p_2^{e_2} \cdots p_t^{e_t}$ for some positive integer t, where these p_i are pairwise distinct primes, and these e_j are positive integers. Then

$$\phi(n) = p_1^{e_1-1} p_2^{e_2-1} \cdots p_t^{e_t-1} (p_1 - 1)(p_2 - 1) \cdots (p_t - 1).$$

Proof: You are not required to learn its proof.

Fermat's and Euler's Theorem

Euler's Theorem: For every integer a and $n \geq 2$ that are relatively prime,

$$a^{\phi(n)} \bmod n = 1.$$

If $n = p$ is prime, we have **Fermat's Theorem**:

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

Proof: It was given in the previous lecture.

The RSA Public-key Cipher

The RSA Public-key Cipher

Plaintext space: $\mathcal{M} = \{0,1\}^*$.

Ciphertext space: $\mathcal{C} = \{0,1\}^*$.

Binary representation of nonnegative integers:

A binary block $M = m_{k-1} \cdots m_1 m_0$ is identified with integer

$$m_{k-1}2^{k-1} + \cdots + m_2 2^2 + m_1 2 + m_0,$$

which is in $\{0, 1, \cdots, 2^k - 1\}$, where k is a positive integer.

The RSA Public-key Cipher

Choose two distinct large primes p and q. Define $n = pq$. Then $\phi(n) = (p-1)(q-1)$.

Select d: $1 \leq d < \phi(n)$ with $\gcd(d, \phi(n)) = 1$.

Compute e: e is the multiplicative inverse of d modulo $\phi(n)$.

Public key: (e, n)

Private key: d

Public-key space: $\mathcal{K}_e = \{1 \leq i < \phi(n) : \gcd(i, \phi(n)) = 1\} \times \{n\}$

Private-key space: $\mathcal{K}_d = \{1 \leq i < \phi(n) : \gcd(i, \phi(n)) = 1\}$.

Remark: The relation between the public key and private key is clear. They form a pair.

The RSA Public-key Cipher

Let $2^k < n < 2^{k+1}$, i.e., $k = \lfloor \log_2 n \rfloor$. Plaintext is broken into binary blocks of length k.

Encryption: For each plaintext block M, $C = M^e \bmod n$.

Decryption: $M = C^d \bmod n$.

Remark: Each plaintext block M, when viewed as an integer, is at most $2^k - 1 \leq n - 2$.

Correctness of Decryption: $M = C^d \bmod n$

Proof. Recall that $n = pq$ and $0 \leq M < n$. Clearly, we have $\gcd(M, n) \in \{1, p, q, n\}$.

Then we are going to prove $M = C^d \bmod n$ for the following cases:

- Case I: $\gcd(M, n) = 1$.
- Case II: $\gcd(M, n) = p$.
- Case III: $\gcd(M, n) = q$.
- Case IV: $\gcd(M, n) = n$ (in this case, $M = 0$).

Correctness of Decryption: $M = C^d \bmod n$

Proof: Case I $\gcd(M, n) = 1$**.** Since $ed = u\phi(n) + 1$ for some u, by Euler's theorem,

$$\begin{aligned} C^d \bmod n &= (M^e \bmod n)^d \bmod n \\ &= M^{ed} \bmod n \\ &= M^{u\phi(n)+1} \bmod n \\ &= (M^{\phi(n)})^u M \bmod n \\ &= (M^{\phi(n)} \bmod n)^u M \bmod n \\ &= M \bmod n \\ &= M, \end{aligned}$$

where the last equality follows from the fact that $M \leq 2^k - 1 < n$.

Correctness of Decryption: $M = C^d \bmod n$

Proof: Case II $\gcd(M, n) = p$**.** In this case, there is some $0 < t < q$ such that $M = tp$. So $\gcd(M, q) = 1$. Since $ed = u\phi(n) + 1$ for some u, by Fermat's theorem, we have

$$\left(M^{u\phi(n)} - 1\right) \bmod q = \left(\left[M^{u(p-1)} \bmod q\right]^{q-1} - 1\right) \bmod q = 0.$$

Whence

$$\begin{aligned}(C^d \bmod n - M) \bmod n &= ((M^e \bmod n)^d \bmod n - M) \bmod n \\ &= \left(M^{ed} - M\right) \bmod n = M\left(M^{ed-1} - 1\right) \bmod n \\ &= tp\left(M^{u\phi(n)} - 1\right) \bmod pq = 0.\end{aligned}$$

The desired equality then follows from the facts that $0 < C^d \bmod n < n$ and $0 < M < n$.

Correctness of Decryption: $M = C^d \bmod n$

Proof: Case III $\gcd(M, n) = q$**.**

Similar to Case II.

Proof: Case IV $\gcd(M, n) = pq$**.**

Trivial because $M = 0$ and $C = 0$.

The RSA Public-key Cipher: Example

Parameters:

p	q	n	$\phi(n)$	e	d
5	11	55	40	7	23

Public key: $(7, 55)$

Private key: 23

Encryption: $M = 28$, $C = M^7 \bmod 55 = 52$.

Decryption: $M = C^{23} \bmod 55 = 28$.

The Parameters of the RSA

Parameters:	p	q	n	$\phi(n)$	e	d

Public key: (e, n)

Private key: d

Other parameters: p, q, $\phi(n)$ must be kept secret.

Question: Why?

The Security of the RSA Public-key Cipher

The Security of the RSA

Brute force attack: Trying all possible private keys.

The cardinality of private key space:

$$|\{1 \leq d < \phi(n) | \gcd(d, \phi(n)) = 1\}| = \phi(\phi(n)) = \phi((p-1)(q-1)).$$

Comment: As long as p and q are large enough, this attack does not work as $\phi((p-1)(q-1))$ will be large!

But the larger the n, the slower the system.

Attacking the RSA Using Mathematical Structures

The factorization problem: You are given a large positive integer n and told that n is the product of two distinct primes. The factorization problem is to find out two primes p and q such that $n = pq$.

Attack: Factor n into pq in some way. Then compute $\phi(n) = (p-1)(q-1)$ and finally compute d from e and $\phi(n)$ with the extended Euclidean algorithm.

Attack: Determine $\phi(n)$ directly without first determining p and q in some way. Finally, compute d from e and $\phi(n)$ with the extended Euclidean algorithm.

Attack: Determine d directly from (e, n) without first factoring n and determining $\phi(n)$ in some way.

Attacking the RSA Using Mathematical Structures

Comment: Determining $\phi(n)$ given n is "equivalent" to factoring n in the following sense.

- Clearly, if you know p and q, you can compute $\phi(n) = (p-1)(q-1)$.
- Given both n and $\phi(n)$, you can determine p and q as follows:

$$p = \frac{n + 1 - \phi(n) - \sqrt{(n+1-\phi(n))^2 - 4n}}{2}, \quad q = n/p.$$

Comment: It is believed that determining d directly from (e, n) is at least as time-consuming as factoring n.

Suggested security evaluation: We may use the difficulty of factorizing n to benchmark the security level of the RSA public-key cipher.

RSA Security Evaluation with the Factorization Problem

Security of RSA with respect to factoring depends on the following factors:

(1) the development of algorithms for factorization; and

(2) the advance in computing power.

Comment: A number of algorithms for factorization are known. None of them is polynomial-time. See https://en.wikipedia.org/wiki/Factorization

Comment: The computing power increases each year due to advances in hardware technology. But the security of the RSA cipher depends **mainly** on the development of algorithms for factorization.

Estimation: If the RSA modulus n is large enough (say, 2048 bits), the factorisation of n is believed to be computationally infeasible.

Security of the RSA Public-Key Cipher

Question: Does the RSA public-key cipher satisfy Conditions C1 and C2 specified in the previous lecture?

Answer: People believe that the answer is positive due to the difficulty of the integer factorisation problem. But no one has proved this belief.

How to Choose p and q?

- They should not be too close to each other. Why?
- Both $(p-1)$ and $(q-1)$ should contain a large prime factor.
- $\gcd(p-1, q-1)$ should be small.

Reference: A. Salomaa, Public-Key Cryptography, 2nd Edition, Springer, 1996, pp. 134–136.

Suggestion: If you wish to learn more, try to work out these problems. However, you are not required to do so.

How to Choose e and d

In theory, e and d could be any integer between 1 and $\phi(n) - 1$ that are relatively prime to $\phi(n)$. However,

- d and e should not be too small. Why?

Reference: A. Salomaa, Public-Key Cryptography, 2nd Edition, Springer, 1996, pp. 134–136.

Suggestion: If you wish to learn more, try to work out this problem. However, you are not required to do so.

An Online Demo of the RSA Cipher

http://www-cs-students.stanford.edu/~tjw/jsbn/rsa2.html