Computer Security

Cunsheng Ding, HKUST

COMP4631

Lecture 10: The RSA Public-Key Block Cipher

Objectives of this Lecture

1. To introduce the RSA public-key block cipher.
2. To look at its security issues.

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

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

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

Example: $\phi(7) = 6$ because

$$\{x : 1 \leq x < 7, \gcd(x, 7) = 1\} = \{1, 2, 3, 4, 5, 6\}.$$

Example: $\phi(6) = 2$ because

$$\{x : 1 \leq x < 6, \gcd(x, 6) = 1\} = \{1, 5\}.$$

Question: What is $\phi(8)$?

Formula for Euler's Totient Function ϕ

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: The first conclusion is straightforward. We now prove the second. Note that pq has only divisors $1, p, q, pq$. The following is the set of integers a such that $1 \leq a < pq$ and $\gcd(a, pq) \neq 1$:

$$\{1p, 2p, \ldots, (q-1)p, \quad 1q, 2q, \ldots, (p-1)q\}$$

which has $(q-1) + (p-1)$ elements. Hence,
$\phi(pq) = pq - 1 - (q-1) - (p-1) = (p-1)(q-1)$.

Fermat's and Euler's Theorem

Euler's Theorem: For every integer a and n 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: See, e.g., W. Stallings, Cryptography and Network Security, pp. 239–241.

Example: Let $a = 3$ and $n = 10$. Then $\phi(10) = 4$ and

$$a^{\phi(n)} \bmod 10 = 3^4 \bmod 10 = 81 \bmod 10 = 1.$$

The RSA Public-key Block Cipher

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

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

Binary representation and integers:

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

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

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

The RSA Public-key Block Cipher

Choose two distinct primes p and q. Define $n = pq$.

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

The RSA Public-key Block Cipher

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

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

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

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

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

Proof: Case I $\gcd(M, n) = 1$.

By Euler's theorem,

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

where u is some integer.

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

Proof: Case II $\gcd(M, n) = p$**.**

We have $M = tp$, $0 < t < q$. So $\gcd(M, q) = 1$. Since $ed = u\phi(n) + 1$ for some u, by Fermat's

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

Whence

$$\left(M^{ed} - M\right) \bmod n = M\left(M^{ed-1} - 1\right) \bmod n = tp\left(M^{u\phi(n)} - 1\right) \bmod pq = 0.$$

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 Block Cipher: Example

Parameters:

p	q	n	ϕ	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	ϕ	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

Brute force attack: Trying all possible private keys.

The number of decryption keys:

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

Comment: As long as p and q are large enough, this attack does not work as $\phi((p-1)(q-1)) - 1$ will be large! But the larger the n, the slower the system.

Attacking the RSA Using Mathematical Structures

Attack: Factor n into pq. Thus $\phi(n)$ and d are known.

Attack: Determine $\phi(n)$ directly, without first determining p and q.

Attack: Determine d directly, without first determining $\phi(n)$.

Attacking the RSA Using Mathematical Structures

Comment: It is believed that determine $\phi(n)$ given n is equivalent to factoring n.

Comment: With presently known algorithms, determining d given e and n, **appears** to be at least as time-consuming as the factoring problem.

Claim: We may use factoring as the benchmark for security evaluation.

RSA Security: Factoring

Security of RSA with respect to factoring depends on:

(1) development of algorithms for factorization;

(2) increase in computing power.

Comment: A number of algorithms for factorization. Most of them involve too much number theory and cannot be introduced here.

Comment: Computing power increases dramatically each year due to advances in hardware technology.

RSA Security: Advance in Factoring

Measure: in MIPS-years, a million-instructions-per-second processor running for one year.

No. of digits	100	110	120	129	130
No. of bits	332	365	398	428	431
Year	1991	1992	1993	1994	1996
MIPS-Years	7	75	830	5000	500

Key size: 1024 to 2048 bits for the near future, due to advance in factorization.

How to Choose p and q

Remark: There are some suggestions for choosing p and q. See the following reference for details.

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

- They should not be too close to each other. Why?

Further Comments on the RSA

- We may define the message and ciphertext spaces as $\mathcal{M} = \mathcal{C} = \mathbf{Z}_{pq}$.
- RSA can be used for both encryption and digital signature. It can be used for sigining messages, because the function $E_{k_e}(x)$ has the same domain and range!