Public-Key Cryptography

Data and Information Management: ELEN 3015

School of Electrical and Information Engineering,
University of the Witwatersrand

Overview

Public-Key Cryptosystem

Design Principles

RSA

- Key generation
- Encryption
- Decryption

1. Public-Key Cryptography

Public-key algorithms are based on mathematical functions

Public-key/Assymetric Cryptography - Key pair, one for encryption, one for decryption

Misconceptions:

- More secure than symmetric encryption - Security is based on the key (length) and the effort to break it, not on the means of encryption.
- Public-key encryption is general purpose, making other forms of encryption obsolete - encrypt and decrypt data at a much lower rate
- Key distribution is a trivial matter - requires a large amount of effort and is no more efficient than exchange systems for symmetric ciphers.

1. Public-Key Cryptography

Characteristics:

- Computationally infeasible to determine decryption key given only knowledge of the cryptographic algorithm and the encryption key.
- Either of the two keys can be used for encryption with the other for decryption.

1. Public-Key Cryptography

Requirements

Computationally easy for party B to generate the key pair K_{pu} and K_{pr}

Computationally easy to encrypt message given the public key

Computationally easy for receiver to decipher message using private key

Computationally infeasible to derive private key from public key

Computationally infeasible to decipher message knowing only public key and algorithm

The encryption and decryption functions can be applied in any order.

2. RSA

Developed by Rivest, Shamir and Adleman (1977)

Block Cipher

Security rests on the difficulty of factoring large numbers

2. RSA

2.1 Encryption

Message divided into blocks m_i

m_i encrypted to form c_i

Use the public key (n, e):

$$c_i = m_i^e \bmod n$$

Block size must be smaller than the size of n.

For binary data, choose $2^s < n$

2. RSA

2.2 Decryption

Use the private key (n, d)

$$m_i = c_i^d \bmod n$$

2. RSA

2.3 Key Generation

Choose two random large prime numbers p and q (length(p) $\approx$ length(q))

Compute the product $n = pq$

Encryption key e: random number such that e and $(p-1)(q-1)$ are relative prime.

Decryption key d: $ed = 1 \bmod (p-1)(q-1)$ (extended Euclidean algorithm)

Public key: e and n $\leftrightarrow$ Private key: d and n

Destroy p and q

2. RSA

2.4 Example - Key Generation

1.) Choose p and q (both prime) $\Rightarrow p = 47,\ q = 59$

2.) $n = p \times q = 47 \times 59 = 2773$

3.) Choose e (e and $\varphi(n) = (p-1)(q-1)$ must be relative prime)

- Factors of 2668 are $\{2, 23, 29\}$
- $e = 17$

4.) Determine d ($d = e^{-1} \bmod \varphi(2773)$ or $ed = 1 \bmod \varphi(2773)$)

- d $= 157$ $(= 2669/17)$

5.) Public Key $= \{17, 2773\}$, Private Key $= \{157, 2773\}$

2. RSA

2.4 Example - Encryption

Plaintext Message:
$m = 0012$

$$\begin{aligned} c &= m^e \mod n \\ &= 12^{17} \mod 2773 \\ &= 336 \end{aligned}$$

$\therefore c = 336$

2. RSA

2.4 Example - Decryption

Ciphertext Message:

$c = 336$

$$\begin{aligned} m &= c^d \mod n \\ &= 336^{157} \mod 2773 \\ &= 12 \end{aligned}$$

$\therefore m = 12$

2. RSA

2.5 Choosing Large Primes

Question: How do you find a large prime/ensure a number is a prime?

2. RSA

2.5 Choosing Large Primes

Question: How do you find a large prime/ensure a number is a prime?

Find all factors → just as much work as cracking RSA

2. RSA

2.5 Choosing Large Primes

Random number generator + test for primality

Probabilistic primality tests: determine with some probability that a number is prime
Some of these tests:

- Solovay-Strassen test
- Lehmann test
- Rabin-Miller test

Witness - a number that does not indicate that p is definitely not prime.

2.6 Solovay-Strassen test

Probabilistic test

Uses Jacobi symbol to test if p is prime

- Choose a random $a < p$
- If $GCD(a, p) \neq 1$, p is composite
- Calculate $j = a^{(p-1)/2} \bmod p$
- Calculate the Jacobi symbol $J(a, p)$
- If $j \neq J(a, p)$, then p is not prime
- If $j = J(a, p)$, probability p prime $> 50\%$

2. RSA

2.6 Solovay-Strassen test - Jacobi Symbol

$J(a, n)$ - Jacobi symbol - defined for any integer a and any odd integer n

1. If $a \equiv b \mod n$ then $J(a, n) = J(b, n)$
2. $$J(a, n) = \begin{cases} 0 & \text{if} \gcd(a, n) \neq 1 \\ \pm 1 & \text{if} \gcd(a, n) = 1 \end{cases}$$
3. $J(ab, n) = J(a, n) \cdot J(b, n)$, so $J(a^2, n) = 1$ (or 0)
4. $J(a, mn) = J(a, m) \cdot J(a, n)$, so $J(a, n^2) = 1$ (or 0)

2.6 Solovay-Strassen test - Jacobi Symbol

5.

$$\begin{aligned} J(m,n) &= J(n,m)\cdot(-1)^{(m-1)(n-1)/4} \\ &= \begin{cases} J(n,m) & \text{if } n \equiv 1 \mod 4 \text{ or } m \equiv 1 \mod 4 \\ -J(n,m) & \text{otherwise} \end{cases} \end{aligned}$$

6.

$$J(-1,n) = (-1)^{(n-1)/2} = \begin{cases} 1 & \text{if } n \equiv 1 \mod 4 \\ -1 & \text{if } n \equiv 3 \mod 4 \end{cases}$$

7.

$$\begin{aligned} J(2,n) &= (-1)^{(n^2-1)/8} \\ &= \begin{cases} 1 & \text{if } n \equiv 1,7 \mod 8 \\ -1 & \text{if } n \equiv 3,5 \mod 8 \end{cases} \end{aligned}$$

2.6 Jacobi Symbol - Example

J(1001,9907)

2. RSA

2.6 Jacobi Symbol - Example

J(1001,9907)

= J(9907,1001) = J(898,1001)

= J(2,1001) J(449,1001) = (449,1001)

=J(1001,449) = J(103,449) = J(449,103)

=J(37,103) = J(103,37) = J(29,37)

=J(37,29) = J(8,29) = J(4,29)(2,29) = -1

2.6 Rabin-Miller test

Probabilistic test

- Choose p, a random number to test
- Calculate b, where b is the number of times 2 divides $p - 1$ (i.e. 2^b is the largest power of 2 that divides $p - 1$)
- Calculate m such that $p = 1 + 2^b \cdot m$

2.6 Rabin-Miller test (Continued)

1. Choose a random number, a, such that a is less than p.
2. Set $j = 0$ and set $z = a^m \mod p$
3. If $z = 1$, or if $z = p - 1$, then p passes and may be prime
4. If $j > 0$ and $z = 1$, then p is not prime
5. Set $j = j + 1$. If $j < b$ and $z \neq p - 1$, set $z = z^2 \mod p$ and go back to step (4). If $z = p - 1$, then p passes the test and may be prime.
6. If $j = b$ and $z \neq p - 1$, then p is not prime.

Error probability 1 out of $(1/4)^t$, $t =$ number of iterations

2. RSA

2.6 Rabin-Miller test Practical Considerations

- Generate random n-bit number p .
- Set MSB and LSB to 1. (Why?)
- Pre-conditioning → Check that p is not divisible by primes $3, 5, 7, \cdots$. Test for all primes < 256 / 2000
- If $j > 0$ and $z = 1$, then p is not prime
- Perform Rabin-Miller test for some random a. If p passes, generate another random a, etc. Use small values for a in order for quicker calculations.

2.8 Proof

Go through proof on own time

2.9 Use of Public-Key Cryptography

No technical difference between public key and private key

Only difference is in how they are used

Alice can encrypt document with private key, everyone with public key can decrypt - Use?

2.9 Use of Public-Key Cryptography

No technical difference between public key and private key

Only difference is in how they are used

Alice can encrypt document with private key, everyone with public key can decrypt - Use?

Signing a document

3. Summary

Public-Key Cryptosystem

- Based on mathematical function
- Consist of two keys
- Used for confidentiality, key distribution or authentication

Design Principles

- Computationally easy to generate the key pair
- Cannot derive one key given the other
- Encryption and decryption can be applied in any order

RSA

- Makes use of expressions with exponentials.
- Security lies in the fact that it is difficult to factor large primes
- 3 stages:
    - Key generation
    - Encryption
    - Decryption