CSEN1001

Computer and Network Security

Mervat AbuElkheir
Ahmad Helmy
Mohamed Abdelrazik

Lecture (7)

Public Key Cryptography and Key Management

RSA

- By Rivest, Shamir & Adleman of MIT in 1977
- Best known & widely used public-key scheme
- Based on exponentiation over integers modulo a prime
  - N.B. exponentiation takes $O((\log n)^3)$ operations (easy)
- Uses large integers (e.g. 1024 bits)
- Security due to cost of factoring large numbers
  - N.B. factorization takes $O(e^{\log n \log \log n})$ operations (hard)

RSA Key Setup

Each user generates a public/private key pair by:

1. Selecting two large primes at random: p, q
2. Computing their system modulus $n = p \cdot q$
   - Note that Euler's Totient $\phi(n) = (p-1) \cdot (q-1)$
3. Selecting at random the encryption key e
   - where $3 \leq e < \phi(n)$, $\gcd(e, \phi(n)) = 1$
4. Solve the following equation to find decryption key d
   - $e \times d \equiv 1 \, mod \, \phi(n) and \, d < \phi(n)$ $[e \times d = 1 + k \times \phi(n) \quad for \, some \, k]$
5. Publish the public encryption key: $PU = \{e, n\}$
6. Keep secret the private decryption key: $PR = \{d, n\}$

RSA Use

- To encrypt a message M the sender:
  - obtains **public key** of recipient $PU = \{e, n\}$
  - computes: $C = M^e \bmod n$, where $0 \leq M < n$
- To decrypt the ciphertext C the owner:
  - uses their private key $PR = \{d, n\}$
  - computes: $M = C^d \bmod n$
- Note that the message M must be smaller than the modulus n (block if needed)

Why RSA Works

- Because of Euler's Theorem:
  - $a^{\phi(n)} mod\ n\ =\ 1$ where $\gcd(a, n) = 1$
- In RSA have:
  - $n = p \cdot q$
  - $\phi(n) = (p-1)(q-1)$
  - carefully choose e & d to be inverses $mod\ \phi(n)$
  - hence $e \cdot d = 1 + k \cdot \phi(n)$ for some k
    - (Recall example) $4 \times 7 = 28 = 1\ mod\ 9$ ➔ $1 + 3 \times 9 = 28$
- Hence :

$$
\begin{aligned}
M\ &=\ C^d\, mod\ n\ =\ (M^e)^d\ mod\ n \\
&=\ M^{e \cdot d}\ mod\ n\ =\ M^{1+k\cdot\phi(n)}\ mod\ n\ =\ M^1 \cdot \left(M^{\phi(n)}\right)^k mod\ n \\
&\equiv\ M^1 \cdot (1)^k\ \equiv\ M
\end{aligned}
$$

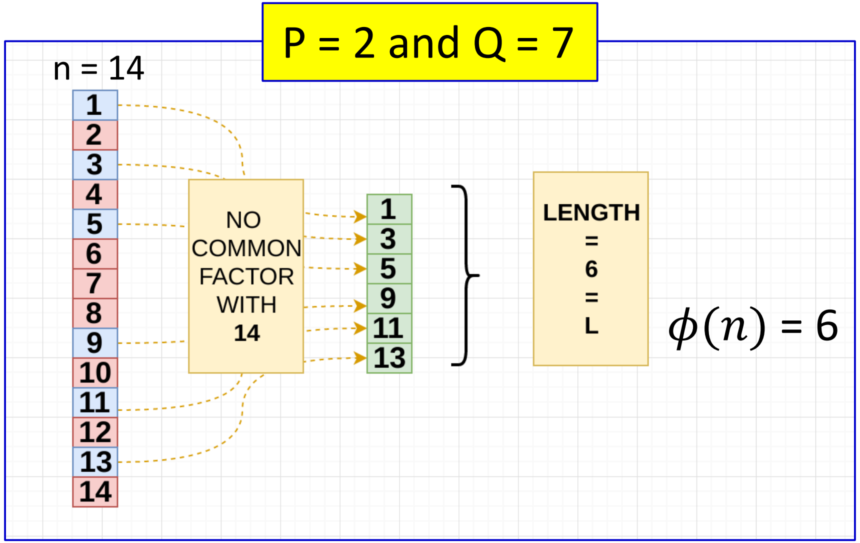
RSA Example – Key Setup

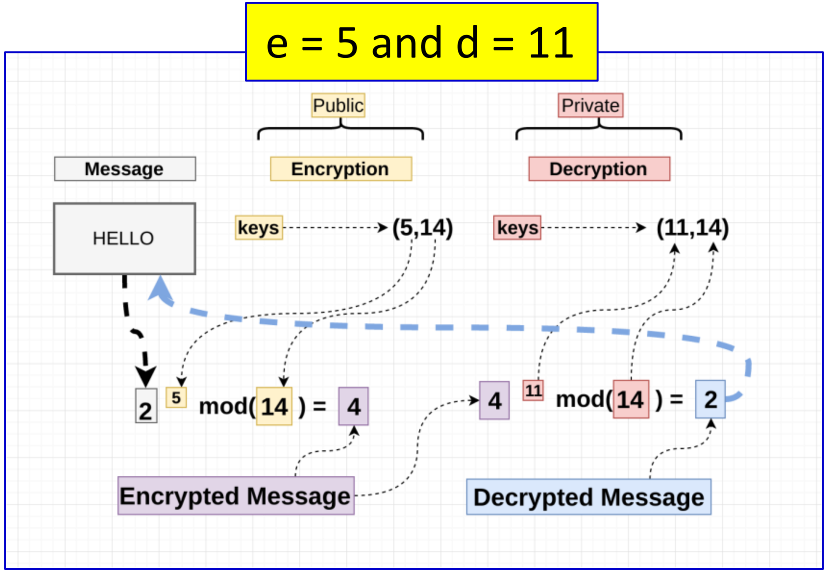
1. Select primes: $p = 17 \,\&\, q = 11$
2. Compute $n = p \times q = 17 \times 11 = 187$
3. Compute $\phi(n) = (p-1) \times (q-1) = 16 \times 10 = 160$
4. Select e: $\gcd(e, 160) = 1$; choose $e = 7$
5. Determine d: $d \times e = 1\, mod\, 160$ and $d < 160$
   Value is $d = 23$ since $23 \times 7 = 161 = 10 \times 160 + 1$
6. Publish public key `PU={7,187}`
7. Keep secret private key `PR={23,187}`

RSA Example – En/Decryption

- Sample RSA encryption/decryption is:
- Given message $M = 88$ (N.B. 88<187)
- Encryption:
  - $C = 88^7 mod\, 187 = 11$
- Decryption:
  - $M = 11^{23} mod\, 187 = 88$

RSA Example – En/Decryption





e has to be coprime with the totient (6) and less than the totient

Candidates are {2, 3, 4, 5}

Only 5 is coprime ➔ e = 5

Compute d = 11

RSA Example – En/Decryption

$$
\begin{aligned}
88^7 \bmod 187 &= [(88^4 \bmod 187) \times (88^2 \bmod 187) \times (88^1 \bmod 187)] \bmod 187 \\
88^1 \bmod 187 &= 88 \\
88^2 \bmod 187 &= 7744 \bmod 187 = 77 \\
88^4 \bmod 187 &= 59{,}969{,}536 \bmod 187 = 132 \\
88^7 \bmod 187 &= (88 \times 77 \times 132) \bmod 187 = 894{,}432 \bmod 187 = 11
\end{aligned}
$$

$$
\begin{aligned}
11^{23} \bmod 187 &= [(11^1 \bmod 187) \times (11^2 \bmod 187) \times (11^4 \bmod 187) \times \\
&\quad (11^8 \bmod 187) \times (11^8 \bmod 187)] \bmod 187 \\
11^1 \bmod 187 &= 11 \\
11^2 \bmod 187 &= 121 \\
11^4 \bmod 187 &= 14{,}641 \bmod 187 = 55 \\
11^8 \bmod 187 &= 214{,}358{,}881 \bmod 187 = 33 \\
11^{23} \bmod 187 &= (11 \times 121 \times 55 \times 33 \times 33) \bmod 187 = 79,720,245 \\
&\quad \bmod 187 = 88
\end{aligned}
$$

Exponentiation

- Can use the Square and Multiply Algorithm
- A fast, efficient algorithm for exponentiation
- Concept is based on repeatedly squaring base
- And multiplying in the ones that are needed to compute the result
- Look at binary representation of exponent
- Only takes $O(\log_2 n)$ multiples for number n
  - e.g. $7^5 = 7^4 \times 7^1 = 10 \bmod 11$
  - e.g. $3^{129} = 3^{128} \times 3^1 = 3^{64} \times 3^{64} \times 3^1 = 4 \bmod 11$

Efficient Encryption

- Encryption uses exponentiation to power e
- Hence if e small, this will be faster
  - often choose e=65537 (2^{16}+1)
  - also see choices of e=3 or e=17

Primality Testing

- Often need to find large prime numbers
- Use statistical primality tests based on properties of primes
  - for which all prime numbers satisfy property
  - but some composite numbers, called pseudo-primes, also satisfy the property
- Can use a slower deterministic primality test

The foundation of RSA's security relies on the fact that given a composite number n that is produced through the multiplication of two prime numbers p and q, it is considered a hard problem to factorize n to determine it's prime factors p and q

RSA Security

- Possible approaches to attacking RSA are:
  - Brute force key search (infeasible given size of numbers)
  - Mathematical attacks (based on difficulty of computing $\phi(n)$, by factoring modulus n)
  - Timing attacks (on running of decryption)

Diffie-Hellman Key Exchange

- First public-key type scheme proposed
- By Diffie & Hellman in 1976 along with the exposition of public key concepts
- Is a practical method for public exchange of a secret key
- Used in a number of commercial products

Diffie-Hellman Key Exchange

- A public-key distribution scheme
  - cannot be used to exchange an arbitrary message
  - rather it can establish a common key
  - known only to the two participants
- Value of key depends on the participants (and their private and public key information)
- Based on exponentiation in a finite (Galois) field (modulo a prime or a polynomial) - easy
- Security relies on the difficulty of computing discrete logarithms (similar to factoring) – hard

Diffie-Hellman Setup

- All users agree on global parameters:
  - large prime integer or polynomial q
  - a being a primitive root $\mathrm{mod}\ q$
  - a is a **primitive root modulo** q if the powers of $a\ mod\ q$ generate all the integers from 1 to $q-1$
  - *Ex:* 3 is a primitive root mod 7 because $3^1 \equiv 3$ mod 7, $3^2 \equiv 2$ mod 7, $3^3 \equiv 6$ mod 7, $3^4 \equiv 4$ mod 7, $3^5 \equiv 5$ mod 7, $3^6 \equiv 1$ mod 7
- Each user (e.g. A) generates their key
  - chooses a secret key (number): $x_A < q$
  - compute their **public key**: $y_A = a^{x_A}\ mod\ q$
  - The exponent x_A is referred to as the discrete logarithm of y_A for the base a, mod q
- Each user makes public that key y_A

Diffie-Hellman Key Exchange

- Shared session key for users A & B is K_{AB}:
  - $K_{AB} = a^{x_A x_B} \, mod \; q$

    $= {y_A}^{x_B} \, mod \; q$ (which ***B*** can compute)

    $= {y_B}^{x_A} \, mod \; q$ (which ***A*** can compute)
- K_{AB} is used as session key in private-key encryption scheme between Alice and Bob
- If Alice and Bob subsequently communicate, they will have the **same** key as before, unless they choose new public-keys
- Attacker needs an x, must solve discrete log $x_B = \mathrm{dlog}_{a,q}(y_B)$

Diffie-Hellman Example

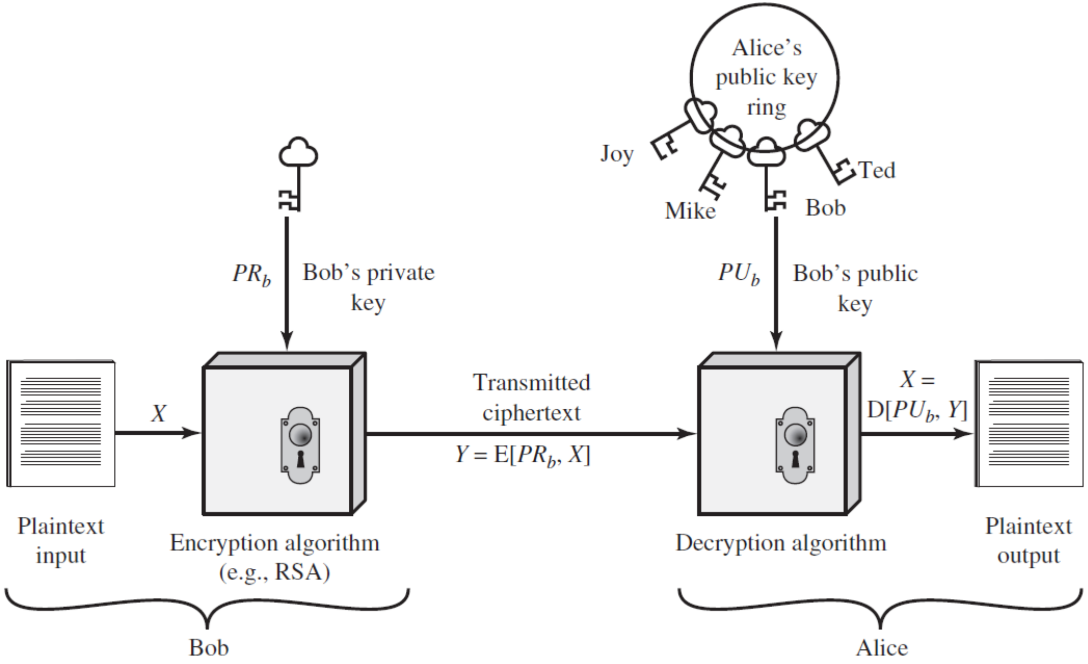
- Users Alice & Bob who wish to swap keys:
- Agree on prime $q = 353$ and $a = 3$
- Select random secret keys:
  - A chooses $x_A = 97$, B chooses $x_B = 233$
- Compute respective public keys:
  - $y_A = 3^{97} \bmod 353 = 40$ (Alice)
  - $y_B = 3^{233} \bmod 353 = 248$ (Bob)
- Compute shared session key as:
  - $K_{AB} = y_B{}^{x_A} \bmod 353 = 248^{97} \bmod 353 = 160$ (Alice)
  - $K_{AB} = y_A{}^{x_B} \bmod 353 = 40^{233} \bmod 353 = 160$ (Bob)

Applications of Public Key Algorithms

1. Digital Signatures

- Key Management and Distribution
  2. The distribution of symmetric keys
  3. The use of public-key encryption to create temporary keys for message encryption
  4. The use of public-key encryption to distribute secret keys
  5. The secure distribution of public keys

1- Digital Signatures



(b) Encryption with private key

2- Symmetric Key Distribution

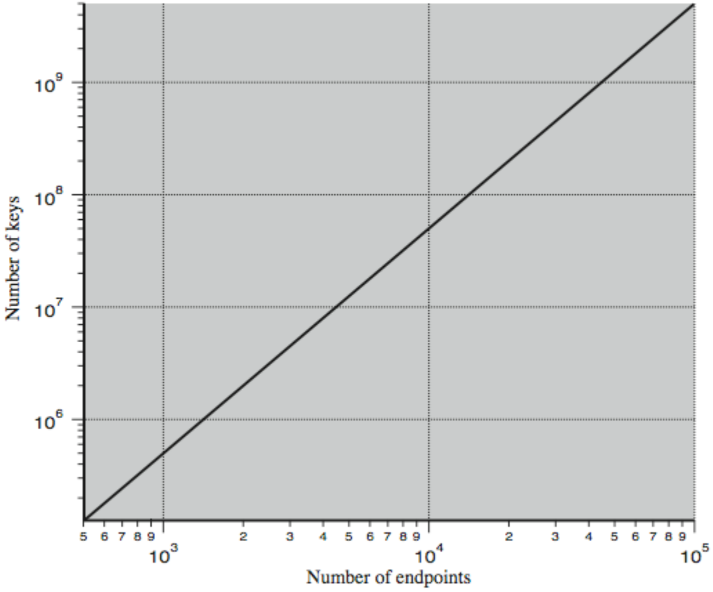
- Symmetric schemes require both parties to share a common secret key
- Issue is how to securely distribute this key
- whilst protecting it from others
- Frequent key changes can be desirable
- Often secure system failure due to a break in the key distribution scheme

Symmetric Key Distribution

- Parties A and B have various **key distribution** alternatives:
  1. A can select key and physically deliver to B
  2. Third party can select & deliver key to A & B
  3. if A & B have communicated previously can use previous key to encrypt a new key
  4. if A & B have secure communications with a third party C, C can relay key between A & B

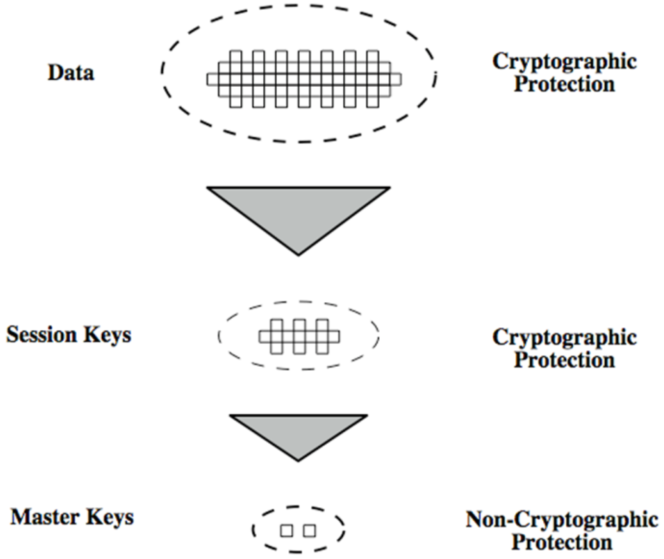
Key Distribution Task

- scale depends on the number of communicating pairs
- If encryption is done at a network or IP level, then a key is needed for each pair of hosts on the network that wish to communicate
- For *N* hosts, the number of required keys is *[N(N – 1)]/2*
- If encryption is done at the application level, then a key is needed for every pair of users or processes that require communication. Thus, a network may have hundreds of hosts but thousands of users and processes.
- A network using node-level encryption with 1000 nodes would need to distribute as many as half a million keys.
- same network supports 10,000 applications, then as many as 50 million keys may be required for application-level encryption

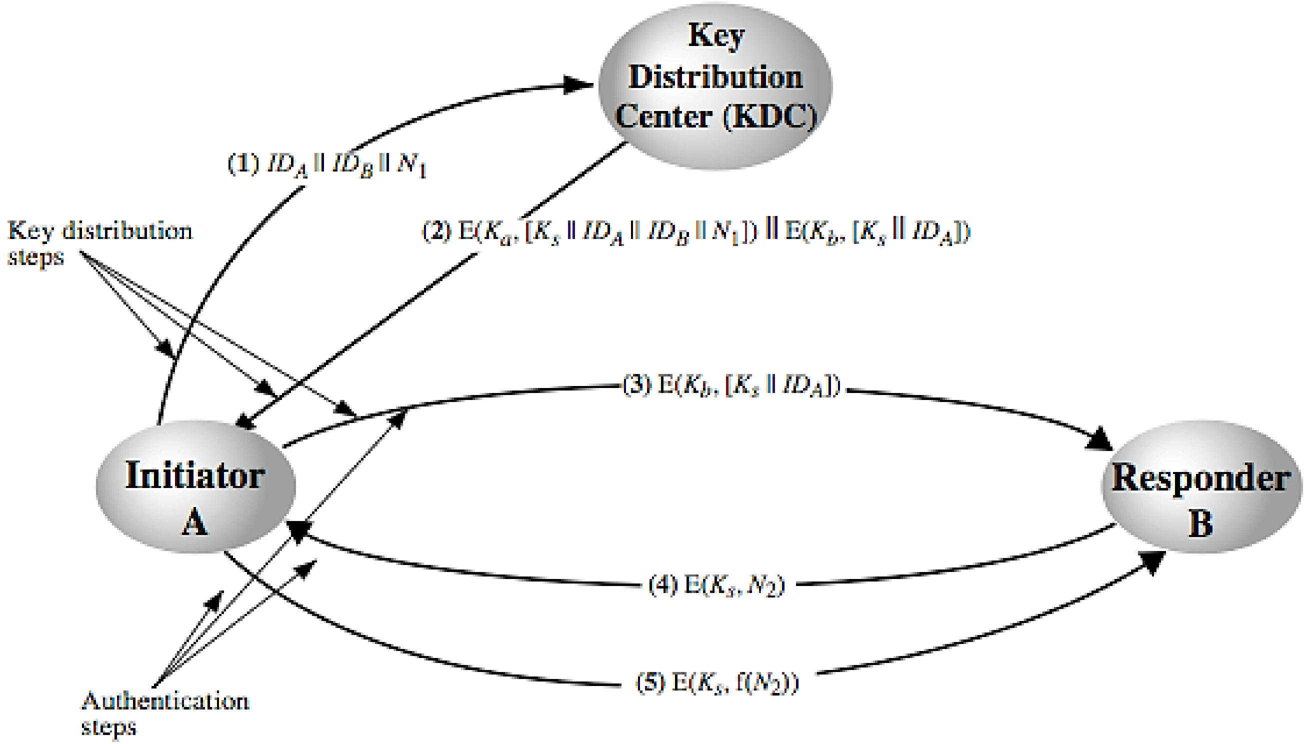


Key Hierarchy

- typically have a hierarchy of keys
- session key
  - temporary key
  - used for encryption of data between users
  - for one logical session then discarded
- master key
  - used to encrypt session keys
  - shared by user & key distribution center
  - reduces scale of problem as only *N* master keys are required



Key Distribution – The Needham-Schroeder Protocol



Key Distribution Issues

- hierarchies of KDC's required for large networks, but must trust each other
- session key lifetimes should be limited for greater security (connection-oriented vs. connection-less communication)
- use of automatic key distribution on behalf of users, but must trust system
- use of decentralized key distribution
- controlling key usage

Decentralized Secret Key Distribution