COMP7120/8120 Cryptography and Data Security

Asymmetric Cryptography: A High-Level Introduction and Applications
Company A’s Problem I

- Company A is a big web service company with over 10,000 employees.

- The president Bob want to make sure that all employees can verify the authenticity of the announcement emails that he sends.

- Q: How to ensure authenticity of these emails.
Company A’s Problem II

• Company A is accepting vulnerability report of their web system from the public.

• They need a design that someone can successfully send the report of a potential vulnerability via email to them.

• Q: How to ensure the confidentiality of reports?
Public Key Cryptography

- Invented and published in 1975
- A public / private key pair is used
- Also known as asymmetric cryptography
- Much slower to compute than secret key cryptography
Public/Private Key

- Public key - encrypt
- Private key - decrypt

How does the secret communication look like?

Alice

wants to send a message

Bob
Public/Private Key

- Alice has her own public and private key pair
- Bob also has his own public and private key pair

- Public key
  - Can be released to the public

- Private Key
  - Must be kept secret.
Message integrity with *digital signatures*

- Alice computes hash, **signs** with her private key (no one else can do this without her key)
- Bob **verifies** hash on receipt using Alice’s public key using the verification equation
Authentication (cont’d)

• Authentication in public key crypto:
  - Hash function to hash the message into a digest
  - The action of sign the digest with (private key)
  - The action of verify the digest with (public key)
Public-Key Requirements

• It must be computationally
  - easy to generate a public / private key pair
  - hard to determine the private key, given the public key

• It must be computationally
  - easy to encrypt using the public key
  - easy to decrypt using the private key
  - hard to recover the plaintext message from just the ciphertext and the public key
Public Key Algorithms

- Public key algorithms covered in this class, and their applications

<table>
<thead>
<tr>
<th>System</th>
<th>Encryption / Decryption?</th>
<th>Digital Signatures?</th>
<th>Key Exchange?</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Diffie-Hellman</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>DSA</td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>
Solving Company A’s Problem I

- Company A is a big web service company with over 10,000 employees.

- The president Bob wants to make sure that all employees can verify the authenticity of the announcement emails that he sends.

Answer:
- Everyone knows Bob’s public key.
- Bob signs the email using his private key.
- Everyone can verify the signed email using Bob’s public key.
Solving Company A’s’s Problem II

• Company A is accepting vulnerability report of their web system from the public.

• They need a design that someone can successfully send the report of a potential vulnerability via email to them.

• Answer:
  - Company A generates a key pair, then release the public key to the public for vulnerability report.
  - Everyone use the public key to encrypt the report.
## Public key vs. Symmetric key

<table>
<thead>
<tr>
<th>Symmetric key</th>
<th>Public key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two parties MUST trust each other</td>
<td>Two parties DO NOT need to trust each other</td>
</tr>
<tr>
<td>Both share same key (or one key is computable from the other)</td>
<td>Two separate keys: a public and a private key</td>
</tr>
<tr>
<td>Typically faster</td>
<td>Typically slower</td>
</tr>
<tr>
<td>Examples: DES, RC5, AES, ...</td>
<td>Examples: RSA, DSA, ECC...</td>
</tr>
</tbody>
</table>
Company A’s Problem III

• Company A provides an on-line chat service for vulnerability report.
  - Requirement 1: confidentiality.
  - Requirement 2: efficiency because there will be a number of message exchanges.

• Q: How to satisfy both requirements?
Digital Envelope: Symmetric+Asymmetric

1. Generate a secret key (called a session key) at random.
2. Encrypt the message using the session key and symmetric algorithm.
3. Encrypt the session key with the recipient’s public key. This becomes the “digital envelope”.
4. Send the encrypted message and the digital envelope to the recipient.
Digital Envelope (cont’d)

Alice (finds a vulnerability)

Bob (company representative)

Knows the public key: $K_p$

Generate a session key $K_s$

Encrypt message $M$ using $K_s$

Encrypted $K_s$ using $K_p$

Send a response using $K_s$

All remaining communication is based on $K_s$

Public key: $K_p$
Private key: $K_i$

1. Use $K_i$ to decrypt $K_s$
2. Use $K_s$ to decrypt $M$
Reading for the Following Classes

• We will discuss the basic theory behind public key cryptography

• Try to get familiar with the following concept before the class:
  - Greatest common divisor (gcd)
    • E.g., gcd(10, 8) = ?
  - Modular Arithmetic
    • E.g., ((10+8) mod 11) * (100000 mod 11) mod 11= ?
  - Prime and relatively prime
  - Fermat's Little Theorem
  - The Totient Function and Euler’s Theorem