Outline

- Number theory overview
- Public key cryptography
- RSA algorithm
A prime number is an integer that can only be divided without remainder by positive and negative values of itself and 1.

Prime numbers play a critical role both in number theory and in cryptography.
two numbers \(a, b\) are relatively prime if they have no common divisors apart from 1

Example: 8 & 15 are relatively prime since factors of 8 are 1, 2, 4, 8 and of 15 are 1, 3, 5, 15 and 1 is the only common factor

Conversely can determine the Greatest Common Divisor by comparing their prime factorizations and using least powers

Example: \(300 = 2^2 \times 3^1 \times 5^2\)

\(18 = 2^1 \times 3^2\)

hence \(\text{GCD}(18, 300) = 2^1 \times 3^1 \times 5^0 = 6\)
Fermat's Theorem

- Fermat’s theorem states the following: If $p$ is prime and is a positive integer not divisible by $p$, then

$$a^{p-1} = 1 \, (\text{mod} \, p)$$

- also known as Fermat’s Little Theorem
- also have: $a^p = a \, (\text{mod} \, p)$
- useful in public key and primality testing
Public Key Encryption

- **Asymmetric encryption** is a form of cryptosystem in which **encryption** and **decryption** are performed using the different keys
  - a public key
  - a private key.

- It is also known as **public-key encryption**
Public Key Encryption

- Asymmetric encryption transforms *plaintext* into *ciphertext* using a *one of two keys* and *an encryption algorithm*.

- Using the paired key and a decryption algorithm, the plaintext is recovered from the ciphertext.

- Asymmetric encryption can be used for *confidentiality*, *authentication*, or *both*.

- The most widely used public-key cryptosystem is *RSA*.

- The difficulty of attacking RSA is based on the difficulty of finding the prime factors of a composite number.
Why Public Key Cryptography?

- Developed to address **two key issues:**
  - **key distribution** – how to have secure communications in general without having to trust a KDC with your key
  - **digital signatures** – how to verify a message comes intact from the claimed sender

- Public invention due to Whitfield Diffie & Martin Hellman at Stanford University in 1976
  - known earlier in classified community
Public Key Cryptography

- public-key/two-key/asymmetric cryptography involves the use of two keys:
  - a **public-key**, which may be known by anybody, and can be used to encrypt messages, and verify signatures
  - a related **private-key**, known only to the recipient, used to decrypt messages, and sign (create) signatures

- Infeasible to determine private key from public

- is **asymmetric** because
  - those who encrypt messages or verify signatures **cannot** decrypt messages or create signatures
Public Key Cryptography

(a) Encryption with public key

Plaintext input -> Encryption algorithm (e.g., RSA) -> Transmitted ciphertext

Bob

Y = E[PU_a, X]

Alice

Decryption algorithm

X = D(PR_a, Y)

Plaintext output
## Symmetric vs. Public Key

### Conventional Encryption

**Needed to Work:**

1. The same algorithm with the same key is used for encryption and decryption.

2. The sender and receiver must share the algorithm and the key.

**Needed for Security:**

1. The key must be kept secret.

2. It must be impossible or at least impractical to decipher a message if no other information is available.

3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.

### Public-Key Encryption

**Needed to Work:**

1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption.

2. The sender and receiver must each have one of the matched pair of keys (not the same one).

**Needed for Security:**

1. One of the two keys must be kept secret.

2. It must be impossible or at least impractical to decipher a message if no other information is available.

3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.
Public Key Cryptosystems
Public Key Applications

- can classify uses into **3 categories**:  
  - encryption/decryption (provide secrecy)  
  - digital signatures (provide authentication)  
  - key exchange (of session keys)

- some algorithms are suitable for all uses, others are specific to one

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Encryption/Decryption</th>
<th>Digital Signature</th>
<th>Key Exchange</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Elliptic Curve</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Diffie-Hellman</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>DSS</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Public Key Requirements

- **Public-Key algorithms rely on two keys where:**
  - it is computationally *infeasible* to find decryption key knowing only algorithm & encryption key
  - it is computationally *easy to encrypt/decrypt messages* when the relevant (en/decrypt) key is known
  - either of the two related keys can be used for encryption, with the other used for decryption (for some algorithms)
Public Key Requirements

- need a trap-door one-way function
- one-way function has
  - $Y = f(X)$ easy
  - $X = f^{-1}(Y)$ infeasible
- a trap-door one-way function has
  - $Y = f_k(X)$ easy, if $k$ and $X$ are known
  - $X = f_k^{-1}(Y)$ easy, if $k$ and $Y$ are known
  - $X = f_k^{-1}(Y)$ infeasible, if $Y$ known but $k$ not known
- a practical public-key scheme depends on a suitable trap-door one-way function
Like symmetric encryption, a public-key encryption scheme is **vulnerable to a brute-force attack**.

The difference is, keys used are too large (>512bits).

Requires the use of **very large numbers**.

**Slow** compared to private key schemes.
RSA

- by Rivest, Shamir & Adleman of MIT in 1977
- best known & widely used public-key scheme
- based on exponentiation in a finite (Galois) field over integers modulo a prime
  - **Note:** exponentiation takes $O((\log n)^3)$ operations (easy!)
- uses *large integers* (eg. 1024 bits)
- **security** due to cost of factoring large numbers
  - **Note:** factorization takes $O(e^{\log n \log \log n})$ operations (hard!)
RSA En/decryption

- **To encrypt a message** $M$ **the sender:**
  - obtains **public key** of recipient $PU=\{e,n\}$
  - computes: $C = M^e \mod n$, where $0 \leq M < n$

- **To decrypt the ciphertext** $C$ **the owner:**
  - uses their private key $PR=\{d,n\}$
  - computes: $M = C^d \mod n$

- **Note that the message** $M$ **must be smaller than the modulus** $n$ **(block if needed)**
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.q
   - note \(\varphi(n)=(p-1)(q-1)\)
3. selecting at random the encryption key e
   - where \(1 < e < \varphi(n)\), \(\text{GCD}(e,\varphi(n)) = 1\)
4. solve following equation to find decryption key d
   - \(e \cdot d = 1 \mod \varphi(n)\) and \(0 \leq d \leq n\)
5. publish their public encryption key: \(\text{PU} = \{e, n\}\)
6. keep secret private decryption key: \(\text{PR} = \{d, n\}\)

For more details, see references:

[1] pages 278-280
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 chose \( e \) & \( d \) to be inverses mod \( \phi(n) \)
  - hence \( e \cdot d = 1 + k \cdot \phi(n) \) for some \( k \)

- hence:
  \[
  C^d = M^{e \cdot d} = M^{1+k \cdot \phi(n)} = M^1 \cdot (M^\phi(n))^k
  = M^1 \cdot (1)^k = M^1 = M \mod n
  \]
RSA Example - Key Setup

1. Select primes: \( p = 17 \) \& \( q = 11 \)
2. Calculate \( n = pq = 17 \times 11 = 187 \)
3. Calculate \( \varphi(n) = (p-1)(q-1) = 16 \times 10 = 160 \)
4. Select \( e: \gcd(e, 160) = 1 \); choose \( e = 7 \)
5. Determine \( d: de = 1 \text{ mod } 160 \) and \( d < 160 \)
   
   Value is \( d = 23 \) since \( 23 \times 7 = 161 = 10 \times 160 + 1 \)

1. Publish public key \( PU = \{7, 187\} \)
2. Keep secret private key \( PR = \{23, 187\} \)
Efficient Operation using Public Key

- To speed up the operation of the RSA algorithm using the public key, a specific choice of $e$ is usually made.
  - The most common choice is $65537 = 2^{16} + 1$;
  - Two other popular choices are 3 and 17.
- Each of these choices has only two 1 bits, so the number of multiplications required to perform exponentiation is minimized.
- However, with a very small public key, such as $e = 3$, RSA becomes vulnerable to a simple attack.
- Suppose we have three different RSA users who all use the value $e = 3$ but have unique values of $n$, namely $(n_1, n_2, n_3)$.
- If user A sends the same encrypted message $M$ to all three users, then the three ciphertexts are $C_1 = M^3 \mod n_1$, $C_2 = M^3 \mod n_2$, and $C_3 = M^3 \mod n_3$. It is likely that $n_1$, $n_2$, and $n_3$ are pairwise relatively prime.
Suppose we have three different RSA users who all use the value $e = 3$ but have unique values of $n$, namely $(n_1, n_2, n_3)$.

If user A sends the same encrypted message $M$ to all three users, then the three ciphertexts are:

- $C_1 = M^3 \mod n_1$,
- $C_2 = M^3 \mod n_2$, and
- $C_3 = M^3 \mod n_3$.

It is likely that $n_1$, $n_2$, and $n_3$ are pairwise relatively prime.

Therefore, one can use the Chinese remainder theorem (CRT) to compute $M^3 \mod (n_1n_2n_3)$. 

Efficient Operation using Public Key
Four possible approaches to attacking the RSA algorithm are

1. **Brute force**: This involves trying all possible private keys.
2. **Mathematical attacks**: There are several approaches, all equivalent in effort to factoring the product of two primes.
3. **Timing attacks**: These depend on the running time of the decryption algorithm.
4. **Chosen ciphertext attacks**: This type of attack exploits properties of the RSA algorithm.
Summary

- Definition of prime number
- Relatively prime numbers
- Public key cryptography
  - Public key
  - Private key
- RSA algorithm
  - Key setup
  - Security
References
