Cryptosystem

Traditional Cryptosystems:

• The two parties agree on a **secret** (one to one) function $f$.

• To send a message $M$, the sender sends the message $f(M)$.

• The receiver computes $f^{-1}(f(M))$.

**Advantage:** Cannot be broken if the function $f$ is used only once (or very few times).

**Disadvantage:** The two parties need a **secure** channel to agree on secrete keys.
Public Key Cryptosystem

Bob needs to send a secret message to Alice:

• Alice generates two functions $P_A()$ and $S_A()$, such that
  1. For any legal message $M$, $S_A(P_A(M)) = M$.
  2. $S_A()$ and $P_A()$ are easy to compute.
  3. It is computationally hard to compute $P_A^{-1}()$.

• Alice publishes the function $P_A()$.

• Bob sends Alice the message $P_A(M)$.

• Alice computes $M = S_A(P_A(M))$.

To decrypt the message without the function $S_A()$ one needs to compute $P_A^{-1}()$. 
Digital Signatures

Bob needs to verify (and be able to prove) that Alice sent him a message $M$:

- Alice generates two functions $P_A()$ and $S_A()$, such that
  1. For any legal message $M$, $P_A(S_A(M)) = M$.
  2. $S_A()$ and $P_A()$ are easy to compute.
  3. It is computationally hard to compute $P_A^{-1}()$.

- Alice publishes the function $P_A()$.

- Alice sends the message $(M, S_A(M))$ to Bob.

- Bob verifies that $P_A(S_A(M)) = M$

To forge Alice’s signature one needs to compute $P_A^{-1}()$. 
Challenge

How to generate a pair of functions \((S_A(), P_A())\) such that for any \(M\):

- \(S_A(P_A(M)) = M\) and \(P_A(S_A(M)) = M\) and it is easy to compute.

- Without the function \(S_A()\), the function \(P_A()\) is hard to “invert” (“one-way function”).

Almost all cryptosystems today use public-key.

We’ll study one such method: RSA.
The RSA Cryptosystem

1. Select at random two LARGE prime numbers $p$ and $q$ (100-200 decimal digits).

2. Compute $n = pq$.

3. Select a small odd integer $e$ relatively prime to $\phi(n) = (p - 1)(q - 1)$.

4. Compute $d$ such that $ed = 1 \mod \phi(n)$ ($d$ exists and is unique!!!).

5. Publish the public key function $P_A(M) = M^e \mod n$ (the pair $(e, n)$).

6. Keep secret the secret key function $S_A(C) = C^d \mod n$. 
Theorem 1. *The RSA system is correct, i.e.*

- $S_A(P_A(M)) = M$;
- $P_A(S_A(M)) = M$
Divisibility

Integer $a$ divides integer $b$ iff $\frac{b}{a}$ is an integer.

The greatest common divisor of $a$ and $b$,

$$d = \gcd(a, b)$$

is the largest integer that divides both $a$ and $b$.

Integers $a$ and $b$ are relatively prime if

$$\gcd(a, b) = 1$$

Integer $p$ is a prime number if for any $a < p$, $\gcd(p, a) = 1$. 
Theorem 2. If \( d = \gcd(a, b) \) then there are integers \( x \) and \( y \) such that

\[
d = ax + by
\]

Proof. Let \( s \) be the smallest positive integer such that \( s = ax + by \) for some integers \( x \) and \( y \).

Let \( q = \left\lfloor \frac{a}{s} \right\rfloor \).

\[
a \mod s = a - qs = a - q(ax + by) = a(1 - qx) + b(-qy)
\]

Thus \( a \mod s \) is also a linear combination of \( a \) and \( b \).
Since \(a \mod s < s\) and \(s\) is the smallest linear combination of \(a\) and \(b\), \(a \mod s = 0\), and \(s\) divides \(a\).

Similarly \(s\) divides \(b\), and \(\gcd(a, b) \geq s\).

But \(\gcd(a, b)\) divides \(s\), thus \(s = \gcd(a, b)\). \(\square\)
Theorem 3. If $e$ and $m = \phi(n)$ are relatively prime the equation

$$ed = 1 \mod m$$

has a unique solution for $d$.

Proof. Since $\gcd(e, m) = 1$ there are integers $x$ and $y$ such that

$$ex + my = 1$$

or

$$ex - 1 = 0 \mod m$$
Fermat’s Theorem

Theorem 4. For any integer $a$ and prime $p$

$$a^{p-1} \mod p = 1$$
The Chinese Reminder Theorem

**Corollary 1.** If \( n_1, n_2, \ldots, n_k \) are pairwise relatively prime and \( n = n_1 n_2 \cdot n_k \), then for all integer \( a \) and \( b \),

\[
a = b \mod n_i
\]

for all \( i = 1, \ldots, k \) iff

\[
a = b \mod n
\]
Theorem 5. Let $gcd(p, q) = 1$ and assume that

$$M^{ed} = M \mod p, \quad \text{and} \quad M^{ed} = M \mod q.$$

Let $n = pq$ then

$$M^{ed} = M \mod n$$

Proof. There are integers $k_1$ and $k_2$ such that

$$M^{ed} = M + k_1p, \quad \text{and} \quad M^{ed} = M + k_2q.$$ 

Thus, $k_1p = k_2q$.

If $k_1 = k_2 = 0$, then $M^{ed} = M = M \mod n$

Else, since $gcd(p, q) = 1$, $q$ divides $k_1$ and

$$M^{ed} = M + k_3(pq) = M \mod n.$$ 

□
Theorem 6. The RSA system is correct, i.e. $S_A(P_A(M)) = M$ and $P_A(S_A(M)) = M$.

Proof.

$$P_A(S_A(M)) = S_A(P_A(M)) = M^{ed} \mod n.$$  

We need to show that $M^{ed} \mod n = M$.

Since $ed = 1 \mod \phi(n)$, for some integer $k$

$ed = 1 + k(p-1)(q-1)$.

If $M = 0 \mod p$ then $M^{ed} = M \mod p$,

If $M \neq 0 \mod p$ then

$$M^{ed} = M^{1+k(p-1)(q-1)} \mod p$$
$$= M(M^{p-1})^{k(q-1)} \mod p$$
$$= M \mod p$$
Similarly $M^{ed} = M \mod q$.

We have

\[ M^{ed} = M \mod p \]
\[ M^{ed} = M \mod q \]

\[ n = pq \], thus by the Chinese reminder theorem for all $M$:

\[ M^{ed} = M \mod n \]

\(\square\)
Complexity

**Theorem 7.** Encrypting and decrypting using the RSA method takes $O(\log n)$ multiplication steps.
Security

If an adversary can factor $n$ it can “guess” $S_A()$.

**Conjecture:** Factoring a large number is “hard”.

**Conjecture:** If factoring is hard breaking RSA is hard.