# RSA

October 23, 2014

# Euclidean Algorithm

- For $a, b \in \mathbb{N}$, Euclid's algorithm computes $d = gcd(a, b)$.

# Euclidean Algorithm

- For $a, b \in \mathbb{N}$, Euclid's algorithm computes $d = gcd(a, b)$.
- A simple way to express Euclid's algorithm is by the recursive formula:

$$gcd(a, b) = \begin{cases} gcd(a, 0) = a & \text{if } b = 0 \\ gcd(b, a(mod\ b)) & \text{if } b \neq 0. \end{cases}$$

# Euclidean Algorithm

- For $a, b \in \mathbb{N}$, Euclid's algorithm computes $d = gcd(a, b)$.
- A simple way to express Euclid's algorithm is by the recursive formula:

$$gcd(a,b) = \begin{cases} gcd(a,0) = a & \text{if } b = 0 \\ gcd(b, a(mod\ b)) & \text{if } b \neq 0. \end{cases}$$

```
EUCLID(a, b)
1.  if b=0
2.     then return a
3.     else return EUCLID(b, a mod b)
```

# Extended Euclidean Algorithm

In practice, we often want to compute integers $(x, y)$ such that $d = gcd(a, b) = ax + by$ in which case we use the extended Euclidean algorithm (due to Lagrange).

# Extended Euclidean Algorithm

In practice, we often want to compute integers $(x, y)$ such that $d = gcd(a, b) = ax + by$ in which case we use the extended Euclidean algorithm (due to Lagrange).

```
EXTENDED-EUCLID(a, b)
1.  if b=0
2.     then return a
3.  (d', x', y') ← EXTENDED-EUCLID(b, a mod b)
4.  (d, x, y) ← (d', y', x' − ⌊a/b⌋y')
5.  return (d, x, y)
```

# Modular exponentiation

$$x \, mod \, N \rightarrow x^2 \, mod \, N \rightarrow x^4 \, mod \, N \rightarrow x^8 \, mod \, N \rightarrow \ldots \rightarrow x^{2^{\lfloor log y \rfloor}} \, mod \, N$$

# Modular exponentiation

$$x \, mod \, N \rightarrow x^2 \, mod \, N \rightarrow x^4 \, mod \, N \rightarrow x^8 \, mod \, N \rightarrow \ldots \rightarrow x^{2^{\lfloor log y \rfloor}} \, mod \, N$$

$$x^y = \begin{cases} (x^{\lfloor y/2 \rfloor})^2 & \text{if } y \text{ is even} \\ x \cdot (x^{\lfloor y/2 \rfloor})^2 & \text{if } y \text{ is odd.} \end{cases}$$

# Modular exponentiation

$$x \bmod N \to x^2 \bmod N \to x^4 \bmod N \to x^8 \bmod N \to \ldots \to x^{2^{\lfloor \log y \rfloor}} \bmod N$$

$$x^y = \begin{cases} (x^{\lfloor y/2 \rfloor})^2 & \text{if } y \text{ is even} \\ x \cdot (x^{\lfloor y/2 \rfloor})^2 & \text{if } y \text{ is odd.} \end{cases}$$

MODULAR-EXPONENTIATION($x, y, N$)
1. if $y$=0: return 1
2. z=MODULAR-EXPONENTIATION($x, \lfloor y/2 \rfloor, N$)
3. if $y$ is even:
4. return $z^2 \bmod N$
5. else:
6. return $x \cdot z^2 \bmod N$

# Modular multiplicative inverse

In modular arithmetic, the modular multiplicative inverse ($a^{-1}$) of an integer $a \ mod \ m$ is an integer $x$ such that $ax \equiv 1 \ (mod \ m)$.

# Modular multiplicative inverse

In modular arithmetic, the modular multiplicative inverse ($a^{-1}$) of an integer $a \; mod \; m$ is an integer $x$ such that $ax \equiv 1 \, (mod \; m)$.
The multiplicative inverse of $a \; mod \; n$ exists iff $a$ and $m$ are coprime ($gcd(a, m) = 1$).

# Modular multiplicative inverse

In modular arithmetic, the modular multiplicative inverse $(a^{-1})$ of an integer $a \ mod \ m$ is an integer $x$ such that $ax \equiv 1 \ (mod \ m)$. The multiplicative inverse of $a \ mod \ n$ exists iff $a$ and $m$ are coprime $(gcd(a, m) = 1)$.

### Example

- Suppose we wish to find modular multiplicative inverse $x$ of $3 \ mod \ 11$: $3^{-1} \equiv x \ (mod \ 11)$.

# Modular multiplicative inverse

In modular arithmetic, the modular multiplicative inverse ($a^{-1}$) of an integer $a \; mod \; m$ is an integer $x$ such that $ax \equiv 1 \; (mod \; m)$. The multiplicative inverse of $a \; mod \; n$ exists iff $a$ and $m$ are coprime ($gcd(a, m) = 1$).

## Example

- Suppose we wish to find modular multiplicative inverse $x$ of $3 \; mod \; 11$: $3^{-1} \equiv x \; (mod \; 11)$.
- This is the same as finding x such that $3x \equiv 1 \; (mod \; 11)$.

# Modular multiplicative inverse

In modular arithmetic, the modular multiplicative inverse $(a^{-1})$ of an integer $a \, mod \, m$ is an integer $x$ such that $ax \equiv 1 \, (mod \, m)$. The multiplicative inverse of $a \, mod \, n$ exists iff $a$ and $m$ are coprime $(gcd(a, m) = 1)$.

## Example

- Suppose we wish to find modular multiplicative inverse $x$ of $3 \, mod \, 11$: $3^{-1} \equiv x \, (mod \, 11)$.
- This is the same as finding x such that $3x \equiv 1 \, (mod \, 11)$.
- Working in $\mathbb{Z}_{11}$ we find that the only value of $x$ that satisfies this congruence is 4 because $3(4) = 12 \equiv 1(mod \, 11)$. Therefore, the modular inverse of 3 modulo 11 is 4.

# Modular multiplicative inverse

In modular arithmetic, the modular multiplicative inverse ($a^{-1}$) of an integer $a \ mod \ m$ is an integer $x$ such that $ax \equiv 1 \ (mod \ m)$.
The multiplicative inverse of $a \ mod \ n$ exists iff $a$ and $m$ are coprime ($gcd(a, m) = 1$).

## Example

- Suppose we wish to find modular multiplicative inverse $x$ of $3 \ mod \ 11$: $3^{-1} \equiv x \ (mod \ 11)$.
- This is the same as finding x such that $3x \equiv 1 \ (mod \ 11)$.
- Working in $\mathbb{Z}_{11}$ we find that the only value of $x$ that satisfies this congruence is 4 because $3(4) = 12 \equiv 1(mod \ 11)$. Therefore, the modular inverse of 3 modulo 11 is 4.
- Generalizing in $\mathbb{Z}$, all possible solutions for this example can be formed from $4 + (11 \cdot z), z \in \mathbb{Z}$, yielding $\{..., -18, -7, \mathbf{4}, 15, 26, ...\}$.

# Fermat Primality Test

Fermat's little theorem states that if $p$ is prime and $1 \leq a < p$ then $a^{p-1} \equiv 1 (mod\ p)$.

## Fermat Primality Test

Fermat's little theorem states that if $p$ is prime and $1 \leq a < p$ then $a^{p-1} \equiv 1 (mod\ p)$.

- If the equality does not hold for a value of $a$, then $p$ is **composite**. If the equality hold for many values of $a$, then we can say that $p$ is **probable prime**.

# Fermat Primality Test

Fermat's little theorem states that if $p$ is prime and $1 \leq a < p$ then $a^{p-1} \equiv 1(mod\ p)$.

- If the equality does not hold for a value of $a$, then $p$ is **composite**. If the equality hold for many values of $a$, then we can say that $p$ is **probable prime**.
- It is possible for a composite number $N$ to pass Fermat's test for certain choices of $a$.

# Fermat Primality Test

Fermat's little theorem states that if $p$ is prime and $1 \leq a < p$ then $a^{p-1} \equiv 1 (mod\ p)$.

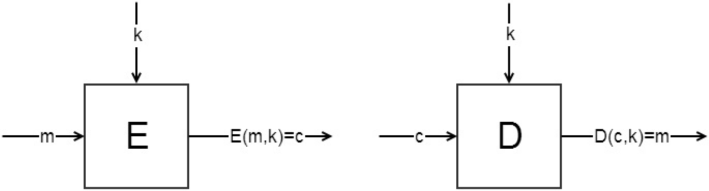
- If the equality does not hold for a value of $a$, then $p$ is **composite**. If the equality hold for many values of $a$, then we can say that $p$ is **probable prime**.
- It is possible for a composite number $N$ to pass Fermat's test for certain choices of $a$.
- Carmichael numbers: rare composite numbers that pass Fermat's test for *all* $a$ relatively prime to $N$.

# Fermat Primality Test

Fermat's little theorem states that if $p$ is prime and $1 \leq a < p$ then $a^{p-1} \equiv 1 (mod\ p)$.

- If the equality does not hold for a value of $a$, then $p$ is **composite**. If the equality hold for many values of $a$, then we can say that $p$ is **probable prime**.
- It is possible for a composite number $N$ to pass Fermat's test for certain choices of $a$.
- Carmichael numbers: rare composite numbers that pass Fermat's test for *all* $a$ relatively prime to $N$.
- For composite $N$, *most* values of $a$ will fail the test.

# Symmetric Cryptography (1)



# Symmetric Cryptography (2)

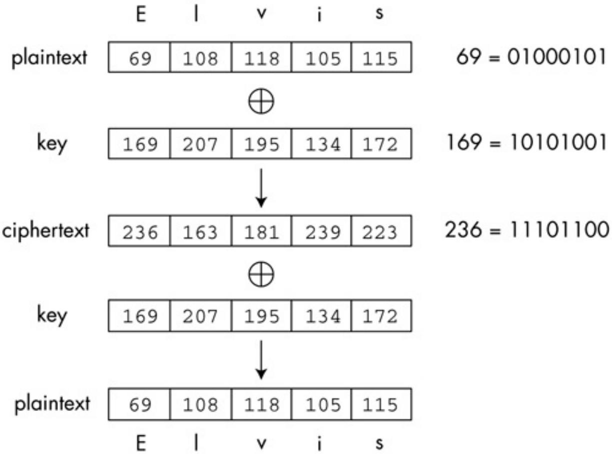
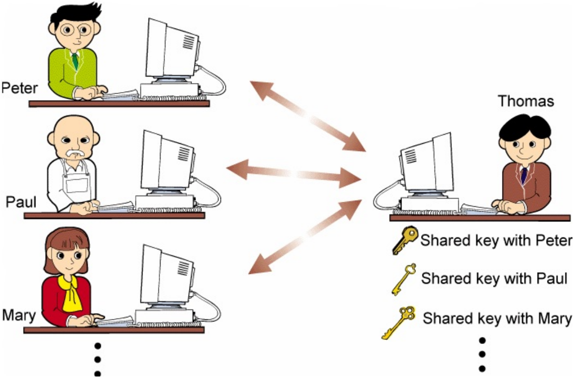
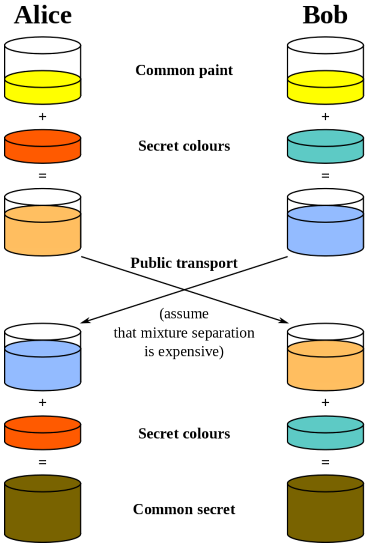



Figure : One-time pad

# Symmetric Cryptography (3)



# Diffie-Hellman-Merkle key exchange



# Asymmetric Cryptography

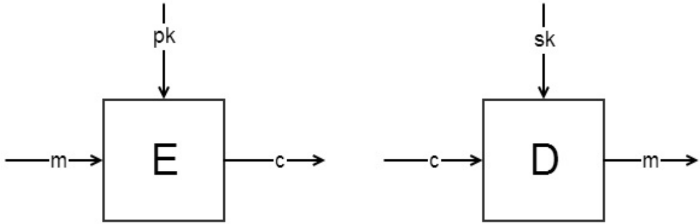



Figure : Public Key Cryptography

# Rivest, Shamir, Adleman (1977)

# 'textbook' RSA: Key Generation

Alice:

# 'textbook' RSA: Key Generation

Alice:

- chooses two large primes $p$ and $q$ of similar size and computes $N = pq$,

# 'textbook' RSA: Key Generation

Alice:

- chooses two large primes $p$ and $q$ of similar size and computes $N = pq$,
- chooses $e \in \mathbb{N}$ coprime to $\phi(N) = (p-1)(q-1)$,

# 'textbook' RSA: Key Generation

Alice:

- chooses two large primes $p$ and $q$ of similar size and computes $N = pq$,
- chooses $e \in \mathbb{N}$ coprime to $\phi(N) = (p-1)(q-1)$,
- computes $d \in \mathbb{N}$ such that $ed \equiv 1 (mod\ \phi(N))$.

## 'textbook' RSA: Key Generation

Alice:

- chooses two large primes $p$ and $q$ of similar size and computes $N = pq$,
- chooses $e \in \mathbb{N}$ coprime to $\phi(N) = (p-1)(q-1)$,
- computes $d \in \mathbb{N}$ such that $ed \equiv 1 (mod\ \phi(N))$.

Alice's **public key** is the pair of integers $(N, e)$ and her **private key** is the integer $d$.

## 'textbook' RSA: Encryption & Decryption

To **encrypt** a message to Alice, Bob does the following:

## 'textbook' RSA: Encryption & Decryption

To **encrypt** a message to Alice, Bob does the following:

- obtains an authentic copy of Alice's public key $(N, e)$,

## 'textbook' RSA: Encryption & Decryption

To **encrypt** a message to Alice, Bob does the following:

- obtains an authentic copy of Alice's public key $(N, e)$,
- encodes the message as an integer $1 \leq m < N$,

# 'textbook' RSA: Encryption & Decryption

To **encrypt** a message to Alice, Bob does the following:

- obtains an authentic copy of Alice's public key $(N, e)$,
- encodes the message as an integer $1 \leq m < N$,
- computes and trasmits the ciphertext $c = m^e (mod\ N)$.

# 'textbook' RSA: Encryption & Decryption

To **encrypt** a message to Alice, Bob does the following:

- obtains an authentic copy of Alice's public key $(N, e)$,
- encodes the message as an integer $1 \leq m < N$,
- computes and trasmits the ciphertext $c = m^e (mod\ N)$.

To **decrypt** the ciphertext, Alice computes $m = c^d (mod\ N)$ and decodes this to obtain the original message.

# RSA Correctness (1)

We will list the tools we need to prove the correctness of RSA:

# RSA Correctness (1)

We will list the tools we need to prove the correctness of RSA:

**Theorem (Fermat's Little Theorem)**

*If $p$ is a prime number and $a$ an integer such that $a$ and $p$ are relatively prime, then $a^{p-1} - 1$ is an integer multiple of $p$ or equivalently $a^{p-1} \equiv 1 (mod\ p)$.*

# RSA Correctness (1)

We will list the tools we need to prove the correctness of RSA:

**Theorem (Fermat's Little Theorem)**

*If $p$ is a prime number and $a$ an integer such that $a$ and $p$ are relatively prime, then $a^{p-1} - 1$ is an integer multiple of $p$ or equivalently $a^{p-1} \equiv 1 (mod\ p)$.*

**Lemma (Euclid's Lemma)**

Let $a, b$ and $d$ be integers where $d \neq 0$. Then if $d$ divides $a \cdot b$ (symbolically $d|a \cdot b$), then either $d|a$ or $d|b$.

# RSA Correctness (1)

We will list the tools we need to prove the correctness of RSA:

**Theorem (Fermat's Little Theorem)**

*If $p$ is a prime number and $a$ an integer such that $a$ and $p$ are relatively prime, then $a^{p-1} - 1$ is an integer multiple of $p$ or equivalently $a^{p-1} \equiv 1(mod\ p)$.*

**Lemma (Euclid's Lemma)**

Let $a, b$ and $d$ be integers where $d \neq 0$. Then if $d$ divides $a \cdot b$ (symbolically $d | a \cdot b$), then either $d | a$ or $d | b$.

**Lemma (2)**

Let $M$ be an integer. Let $p$ and $q$ be prime numbers with $p \neq q$. Then if $a \equiv M(mod\ p)$ and $a \equiv M(mod\ q)$, then $a \equiv M(mod\ p \cdot q)$.

# RSA Correctness (2)

We need to prove that $(M^e)^d \equiv M^{ed} \equiv M(mod\ N)$.

**Proof.**

We first show that $M^{ed} \equiv M(mod\ p)$ and $M^{ed} \equiv M(mod\ q)$. The desired result follows from lemma 2.
To show $M^{ed} \equiv M(mod\ p)$, we consider two cases:
$M \equiv 0(mod\ p)$, or $M \not\equiv 0(mod\ p)$.

**Case 1.** $M \equiv 0(mod\ p)$. Then $M$ is an integer multiple of $p$, say $M = p \cdot w, w \in \mathbb{Z}$. Then $M^{ed} = (p \cdot w)^{ed} = p \cdot p^{ed-1} \cdot w^{ed}$. So both $M$ and $M^{ed}$ are integer multiples of $p$. Thus $M^{ed} \equiv M(mod\ p)$.

# RSA Correctness (2)

We need to prove that $(M^e)^d \equiv M^{ed} \equiv M(mod\ N)$.

**Proof.**

**Case 2.** $M \not\equiv 0(mod\ p)$. This means that $p$ and $M$ are relatively prime. Thus we can use Fermat's Little Theorem. We have $M^{p-1} \equiv 1(mod\ p)$.
From the way the decryption key $d$ is defined above, we have $ed - 1 = (p-1)\cdot(q-1)\cdot k,\ k \in \mathbb{Z}$. We then have:

$$\begin{aligned} M^{ed} &= M^{ed-1}\cdot M \\ &= M^{(p-1)\cdot(q-1)\cdot k}\cdot M \\ &= (M^{p-1})^{(q-1)\cdot k}\cdot M \\ &\equiv (1)^{(q-1)\cdot k}\cdot M \pmod p \qquad \text{(apply Fermat's Little Theorem)} \\ &\equiv M \pmod p \end{aligned}$$

# RSA Correctness (2)

We need to prove that $(M^e)^d \equiv M^{ed} \equiv M(mod\ N)$.

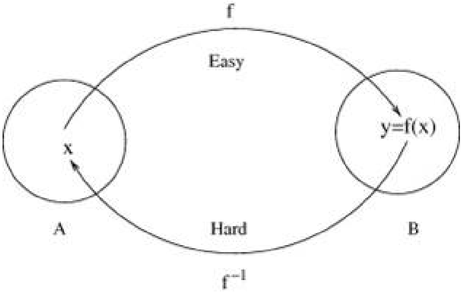
**Proof.**

In a similar way we can show that $M^{ed} = M(mod\ q)$.

By Lemma 2, it follows that $M^{ed} \equiv M(mod\ N = p \cdot q)$.

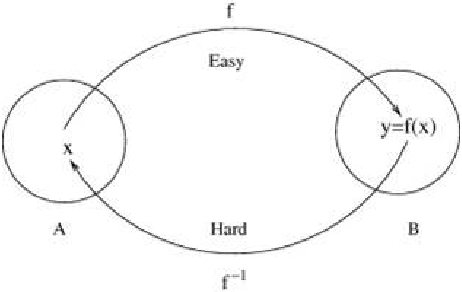
$\square$

# One-way & Trapdoor fuctions (1)




A *one-way function* is a function that is easy to compute on every input, but hard to invert given the image of a random input.

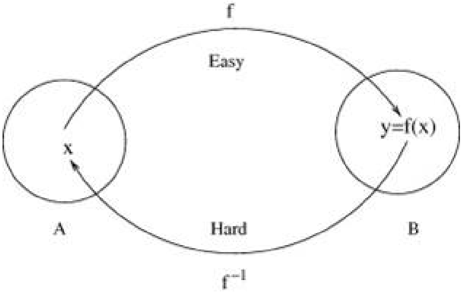
# One-way & Trapdoor fuctions (1)




A *one-way function* is a function that is easy to compute on every input, but hard to invert given the image of a random input.

Do one-way functions exist?

# One-way & Trapdoor fuctions (1)




A *one-way function* is a function that is easy to compute on every input, but hard to invert given the image of a random input.

Do one-way functions exist?
Yes, if $\mathbf{P} \neq \mathbf{NP}$.

# One-way & Trapdoor fuctions (2)

**Candidates for one-way functions:**

- Multiplication and factoring
- The Rabin function (modular squaring)
- Discrete exponential and logarithm
- Cryptographically secure hash functions
- Elliptic curves

# One-way & Trapdoor fuctions (3)

# One-way & Trapdoor fuctions (3)

A trapdoor function is a function that is easy to compute in one direction, yet believed to be difficult to compute in the opposite direction without **special information**, called the "trapdoor".

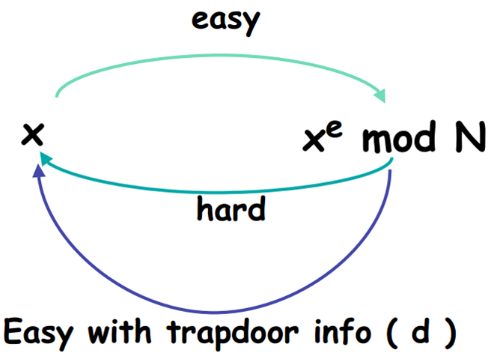
# One-way & Trapdoor fuctions (3)

A trapdoor function is a function that is easy to compute in one direction, yet believed to be difficult to compute in the opposite direction without **special information**, called the "trapdoor".

As of 2004, the best known trapdoor function candidates are the RSA and Rabin functions. Both are written as exponentiation modulo a composite number, and both are related to the problem of prime factorization.

# One-way & Trapdoor fuctions (4)

Indeed, exponentiation modulo $N$ is a one-way permutation on $(\mathbb{Z}/N\mathbb{Z})^\times$ when $e$ is co-prime to $\phi(N)$. The private key allows the permutation to be efficiently inverted and it is the trapdoor.



# Key Length and Encryption Strength

| **p,q** | **N** | **time to crack** |
|---|---|---|
| 256 bits | 512 bits | few weeks |
| 512 bits | 1024 bits | 50-100 years |
| 1024 bits | 2048 bits | $>$100 years |
| 2048 bits | 4096 bits | $\approx$ age of the universe |