[thm]Lemma [thm]Fact
[thm]Example

Week 6: Public Key

Jay Daigle

Occidental College

October 3, 2019

Order of an Integer

Order of an Integer

Theorem (Fermat's Little Theorem)

Let p be a prime. If $\gcd(a, p) = 1$, then $a^{p-1} \equiv 1 \mod p$.

Order of an Integer

Theorem (Fermat's Little Theorem)

Let p be a prime. If $\gcd(a, p) = 1$, *then* $a^{p-1} \equiv 1 \mod p$.

Definition

The *Euler totient function* $\phi(m) = \#\{i : 0 < i < m, \gcd(i, m) = 1\}$.

Order of an Integer

Theorem (Fermat's Little Theorem)

Let p be a prime. If $\gcd(a, p) = 1$, *then* $a^{p-1} \equiv 1 \mod p$.

Definition

The *Euler totient function* $\phi(m) = \#\{i : 0 < i < m, \gcd(i, m) = 1\}$.
$\phi(p) = p - 1$ and $\phi(pq) = (p - 1)(q - 1)$.

Order of an Integer

Theorem (Fermat's Little Theorem)

Let p be a prime. If $\gcd(a, p) = 1$, then $a^{p-1} \equiv 1 \mod p$.

Definition

The *Euler totient function* $\phi(m) = \#\{i : 0 < i < m, \gcd(i, m) = 1\}$.
$\phi(p) = p - 1$ and $\phi(pq) = (p - 1)(q - 1)$.

Theorem (Euler's Theorem)

If $a, m \in \mathbb{N}$ and $\gcd(a, n) = 1$, then $a^{\phi(m)} \equiv 1 \mod m$.

Key Generation

Key Generation

1. p a large prime, $g \in \mathbb{Z}/p\mathbb{Z}$ such that $\text{ord}_p(g)$ is large prime.
2. Alice chooses a *private key* $a \in \mathbb{Z}/p\mathbb{Z}$.
3. Alice computes a *public key* $A \equiv g^a \mod p$.

Key Generation

1. p a large prime, $g \in \mathbb{Z}/p\mathbb{Z}$ such that $\text{ord}_p(g)$ is large prime.
2. Alice chooses a *private key* $a \in \mathbb{Z}/p\mathbb{Z}$.
3. Alice computes a *public key* $A \equiv g^a \mod p$.

Encryption

Bob sends Alice a number $2 < m < p$.

Key Generation

1. p a large prime, $g \in \mathbb{Z}/p\mathbb{Z}$ such that $\text{ord}_p(g)$ is large prime.
2. Alice chooses a *private key* $a \in \mathbb{Z}/p\mathbb{Z}$.
3. Alice computes a *public key* $A \equiv g^a \mod p$.

Encryption

Bob sends Alice a number $2 < m < p$.

1. Bob generates a random *ephemeral key* k.
2. Bob computes $c_1 \equiv g^k \mod p$ and $c_2 \equiv mA^k \mod p$. He sends Alice (c_1, c_2).

Key Generation

1. p a large prime, $g \in \mathbb{Z}/p\mathbb{Z}$ such that $\text{ord}_p(g)$ is large prime.
2. Alice chooses a *private key* $a \in \mathbb{Z}/p\mathbb{Z}$.
3. Alice computes a *public key* $A \equiv g^a \mod p$.

Encryption

Bob sends Alice a number $2 < m < p$.

1. Bob generates a random *ephemeral key* k.
2. Bob computes $c_1 \equiv g^k \mod p$ and $c_2 \equiv mA^k \mod p$. He sends Alice (c_1, c_2).

Decryption

Key Generation

1. p a large prime, $g \in \mathbb{Z}/p\mathbb{Z}$ such that $\text{ord}_p(g)$ is large prime.
2. Alice chooses a *private key* $a \in \mathbb{Z}/p\mathbb{Z}$.
3. Alice computes a *public key* $A \equiv g^a \mod p$.

Encryption

Bob sends Alice a number $2 < m < p$.

1. Bob generates a random *ephemeral key* k.
2. Bob computes $c_1 \equiv g^k \mod p$ and $c_2 \equiv mA^k \mod p$. He sends Alice (c_1, c_2).

Decryption

1. Alice computes $x \equiv c_1^a \mod p$, and then $x^{-1} \equiv c_1^{-a} \mod p$.

Key Generation

1. p a large prime, $g \in \mathbb{Z}/p\mathbb{Z}$ such that $\text{ord}_p(g)$ is large prime.
2. Alice chooses a *private key* $a \in \mathbb{Z}/p\mathbb{Z}$.
3. Alice computes a *public key* $A \equiv g^a \mod p$.

Encryption

Bob sends Alice a number $2 < m < p$.

1. Bob generates a random *ephemeral key* k.
2. Bob computes $c_1 \equiv g^k \mod p$ and $c_2 \equiv mA^k \mod p$. He sends Alice (c_1, c_2).

Decryption

1. Alice computes $x \equiv c_1^a \mod p$, and then $x^{-1} \equiv c_1^{-a} \mod p$.
2. Alice then computes $c_2 x^{-1} \mod p$, which is equivalent to m.

RSA Algorithm

RSA Algorithm

Key Generation

RSA Algorithm

Key Generation

1. Bob chooses two primes p, q and computes $N = pq, M = (p-1)(q-1)$.

RSA Algorithm

Key Generation

1. Bob chooses two primes p, q and computes $N = pq, M = (p-1)(q-1)$.
2. Bob chooses e such that $\gcd(e, M) = 1$.

RSA Algorithm

Key Generation

1. Bob chooses two primes p, q and computes $N = pq, M = (p-1)(q-1)$.
2. Bob chooses e such that $\gcd(e, M) = 1$.
3. The public key is (N, e).

RSA Algorithm

Key Generation

1. Bob chooses two primes p, q and computes $N = pq, M = (p-1)(q-1)$.
2. Bob chooses e such that $\gcd(e, M) = 1$.
3. The public key is (N, e).
4. Bob computes $d \equiv e^{-1} \mod M$, so that $ed \equiv 1 \mod M$. Bob's private key is the pair (M, d).

RSA Algorithm

Key Generation

1. Bob chooses two primes p, q and computes $N = pq, M = (p-1)(q-1)$.
2. Bob chooses e such that $\gcd(e, M) = 1$.
3. The public key is (N, e).
4. Bob computes $d \equiv e^{-1} \mod M$, so that $ed \equiv 1 \mod M$. Bob's private key is the pair (M, d).

Encryption and Decryption

Now Alice wishes to send Bob an integer m with $1 \leq m < N$.

RSA Algorithm

Key Generation

1. Bob chooses two primes p, q and computes $N = pq, M = (p-1)(q-1)$.
2. Bob chooses e such that $\gcd(e, M) = 1$.
3. The public key is (N, e).
4. Bob computes $d \equiv e^{-1} \mod M$, so that $ed \equiv 1 \mod M$. Bob's private key is the pair (M, d).

Encryption and Decryption

Now Alice wishes to send Bob an integer m with $1 \leq m < N$.

1. Alice computes $c \equiv m^e \mod N$. She sends c to Bob.

RSA Algorithm

Key Generation

1. Bob chooses two primes p, q and computes $N = pq, M = (p-1)(q-1)$.
2. Bob chooses e such that $\gcd(e, M) = 1$.
3. The public key is (N, e).
4. Bob computes $d \equiv e^{-1} \mod M$, so that $ed \equiv 1 \mod M$. Bob's private key is the pair (M, d).

Encryption and Decryption

Now Alice wishes to send Bob an integer m with $1 \leq m < N$.

1. Alice computes $c \equiv m^e \mod N$. She sends c to Bob.
2. Bob computes $c^d \mod N$ and receives Alice's message m.

Factoring Algorithms

Factoring Algorithms

The Quadratic Sieve

$$O\left(e^{\sqrt{\log(n)\log\log(n)}}\right).$$

Factoring Algorithms

The Quadratic Sieve

$$O\left(e^{\sqrt{\log(n)\log\log(n)}}\right).$$

The General Number Field Sieve

$$O\left(e^{\sqrt[3]{64/9}\ln(n)^{1/3}\ln(\ln(n))^{2/3}}\right).$$