Multiplication Principle

	Repetition	No Repitition
Ordered	n^r	$\frac{n!}{(n-r)!}$
Unordered	$\binom{n+r-1}{r}$	$\frac{n!}{r!(n-r)!}$

$$P(A \mid B) = \frac{P(A \cup B)}{P(B)}$$

Basic Number Theory

Euclidean Algorithm

$ax + by = d$

$$\begin{aligned} \gcd(6, 10) &= 2 \\ 2 &= 10x + 6y \\ &= 10(-1) + 6(2) \end{aligned}$$

Congruence Theory

Fast Powering Algorithm

$$\begin{aligned} 2^{100} \pmod 5 &\equiv 2^{64} \cdot 2^{32} \cdot 2^4 \pmod 5 \\ &\equiv 1 \cdot 4 \pmod 5 \\ &\equiv 4 \pmod 5 \end{aligned} \qquad \begin{aligned} 2^1 &= 2 \pmod 5 \\ 2^2 &= 4 \pmod 5 \\ 2^4 &= 1 \pmod 5 \end{aligned}$$

Fermat's Little Theorem

If $p \nmid a \wedge p$ is prime $\implies a^{p-1} \equiv 1 \pmod p$

Euler's Theorem

$a^{\phi(n)} \equiv 1 \pmod n$

Primitve Root Theorem

Every prime p has a primitive root

Cryptography

Symmetric

Ideal Requirements

1) With key it should be easy to encrypt/decrpyt.
2) Without key it should be difficult to encrypt/decrypt.
3) Even with lots of plaintexts <-> combinations, it should be difficult to find the key.
4) Choosen plaintext attack: Attacker can choose plaintexts and see the corresponding ciphertexts.

Multiplication

vulnerable to plaintext <-> cyphertext attacks

$$E(x) = x \cdot k \pmod n$$

Primality Testing

Miller-Rabin Test

builds off of Fermat's Test

Probabilistic $\rightarrow$ try 100 candidates (to be witnesses)

If n is composite overwhelmingly likely to find a witness

$$\text{If } n \text{ is prime, } a^{n-1} \equiv 1 \pmod n$$

1. Make a table where $n - 1 = 2^k q, q \in$ Odd
$$a^q, a^{2q}, a^{4q}, \ldots, a^{2^{k-1}q}$$
2. Either first number is 1 (probably prime), or one of the numbers is -1
3. Last number **has** to be 1 (we passed Fermat's test)
4. If second to last number is not 1, then n is composite
5. Consider the first term in the sequence congruent to 1. If the preceding term is *not* congruent to -1, then n is composite.

$$\begin{aligned} n = 252601, n - 1 &= 2^3 \cdot 31575 \\ a &= 85132 \end{aligned}$$

$$\begin{aligned} a^{31575} &\equiv 191102 \pmod n \\ a^{2 \cdot 31575} &\equiv 184829 \pmod n \\ a^{4 \cdot 31575} &\equiv 1 \pmod n \end{aligned}$$

*Conclusion: n is **composite**.*

$$\begin{aligned} n = 104717, n - 1 &= 2^2 \cdot 26179 \\ a &= 96152 \end{aligned}$$

$$a^{26179} \equiv 1 \pmod n$$

*Conclusion: n is **probably prime**.*

$$\begin{aligned} n = 3057601, n - 1 &= 2^6 \cdot 47775 \\ a &= 99908 \end{aligned}$$

$$\begin{aligned} a^{47775} &\equiv 1193206 \pmod n \\ a^{2 \cdot 47775} &\equiv 2286397 \pmod n \\ a^{2^2 \cdot 47775} &\equiv 235899 \pmod n \\ a^{2^3 \cdot 47775} &\equiv 1 \pmod n \end{aligned}$$

*Conclusion: n is **composite**.*

$$\begin{aligned} n = 577757, n - 1 &= 2^2 \cdot 144439 \\ a &= 314997 \end{aligned}$$

$$\begin{aligned} a^{144439} &\equiv 373220 \pmod n \\ a^{2 \cdot 144439} &\equiv -1 \pmod n \end{aligned}$$

*Conclusion: n is **probably prime**.*

Shanks's Algorithm

$$\begin{aligned} g^x &\equiv h \pmod p \\ p &\text{ prime} \\ g &\text{ primitive root} \\ N &= p - 1 \\ \text{Solve for } x : g^x &\equiv h \pmod p \\ n &= \lceil \sqrt{N} \rceil \end{aligned}$$

$$\begin{gathered} g, g^2, g^3, \ldots, g^n \\ g^{-n}, g^{-n+1}, \ldots, g^{-1} \\ hg^{-n}, hg^{-2n}, \ldots, hg^{-(n-1)n} \end{gathered}$$

$$\begin{aligned} p &= 101 \\ g &= 2 \end{aligned}$$

$$\begin{aligned} \text{Once you get } hg^{-jn} &= g^i \pmod p \\ h &= g^{i+jn} \pmod p \end{aligned}$$

Pollart's Rho Algorithm

An improvement only in space.

Randomized Algorithm

Work out random powers of g, and random powers of hg. Compare the two lists, and if you find a match, you can solve for x.

$$\begin{aligned} g^x &\equiv h \pmod p \\ h &= g^x \pmod p \\ h &= g^{x+kn} \pmod p \end{aligned}$$

RSA

Public Key Cryptography

$$p, q \text{ large prime numbers} \sim 2^{1000}$$
$$N = pq$$
$$\phi(N) = (p-1)(q-1)$$

- Encryption exponent e s.t $\gcd(e, \phi(n) = 1)$
- Decryption exponent d s.t $ed \equiv 1 \pmod{\phi(N)}$
- Encrypting: $m \to m^e \equiv c \pmod{N}$
- Decrypting: $c \to c^d \equiv m^{ed} \equiv m \pmod{N}$

Digital Signatures

Private signing key d, public verification key e

$$\text{Signer (Sam) } S \equiv D^d \pmod{N}$$
$$\text{Verifier (Victor) } D \equiv S^e \pmod{N}$$

Size of input

Given N, the size of the input is $\log_2 N$ bits.

Group Theory

Multiplicative Group Mod p

$$\mathbf{F}_p^x = \{1, 2, 3, \ldots, p-1\}, \text{ under multiplication modulo } p.$$

A group G is a set, together with a rule for combining ordered pairs of elements to yield another element in the same set.

(I) $e \times a = a \times e = a$ for all $a \in G$
(II) $a^{-1} \times a = a^{-1} \times a = e$ for all $a \in G$
(III) $a \times (b \times c) = (a \times b) \times c$ for all $a, b, c \in G$

All the numbers in the set must be coprime to p to form a cyclic group.

$$13^{-1} \equiv 1 \pmod{17}$$
$$13x + 17y = 1$$

$$17 = 1(13) + 4$$
$$13 = 3(4) + 1$$

$$\begin{aligned} 1 &= 13 - 3(4) \\ &= 13 - 3(17 - 13) \\ &= 4(13) - 3(17) \\ &\equiv 4 \pmod{17} \end{aligned}$$

Diffie-Hellman

Key Exchange

p large prime ($\sim 2^{1000}$)
$g \in \mathbf{F}_p^x$ has <u>large prime order</u> in $\mathbf{F}_p^x$

1. Alice picks a and sends $A \equiv g^a \pmod{p}$ to Bob
2. Bob picks b and sends $B \equiv g^b \pmod{p}$ to Alice
3. Alice computes $B^a \equiv (g^b)^a \equiv g^{ab} \pmod{p}$
4. Bob computes $A^b \equiv (g^a)^b \equiv g^{ab} \pmod{p}$

El Gamel

- Alice picks a and sends $A \equiv g^a \pmod{p}$ to Bob
- Bob chooses k and computes $c_1 \equiv g^k, g_2 \equiv mA^k$
- Alice receives c_1, c_2 and computes $m \equiv c_2(c_1^{-a}) \pmod{p}$

$$\begin{aligned} &c_2 c_2^{-a} \\ &\equiv m g^{ak} (g^k)^{-a} \\ &\equiv m \pmod{p} \end{aligned}$$

Digital Signatures

- Samantha chooses a (secret), computes $A \equiv g^a \pmod{p}$
- Also chooses k coprime to $p-1$, ie $\gcd(k, p-1) = 1$

$$S_1 = g^k \pmod{p}$$
$$S_2 = (D - aS_1)k^{-1} \pmod{p-1}$$

Verification

$$A^{S_1} S_1^{S_2} \equiv g^D \pmod{p}$$

$$\begin{aligned} A_1^{S_1} S_1^{S_2} &= g^{aS_1} g^{\cancel{k}(D-aS_1)\cancel{k^{-1}}} \\ &= g^{aS_1} g^{D-aS_1} = g^D \end{aligned}$$