Modular Arithmetic

- GCD with two numbers a and b is as follows:
  $a = q_0 b + r_0$
  $b = q_1 r_0 + r_1$
  $r_0 = q_2 r_1 + r_2$
  etc ...Until we get the last non zero remainder. We can then back substitute in and find $b^{-1} moda$
- Square roots: $x \equiv \sqrt(2) mod 7 \rightarrow x^2 mod 7 = 2 \rightarrow x = 3, x^2 = 9$
- Euler Fermat: $a^{\phi}(n) \mod n = 1$, where $\phi(n)$ is the number of prime values less than n
- Property: for prime p, $a^{p-1} \equiv 1 \mod p$
- Property: If a is a square mod p, $a_{(p-1)/2} \equiv 1 \mod p$
- Property: $x^{p-2} \equiv x^{-1} \mod p$

Chinese Remainder Theorem

- $x \mod (pq) =< \pm x \mod p, \pm x \mod q >$
- Factoring: $4 \mod (3*5) = (2,2),(2,3),(1,2),(1,3)$, basically $4 = \pm 2$, then mod each factor in n

RSA

- asymmetric - has public and private keys
- $ed \equiv 1 mod (p-1)(q-1)$, e is public, d is private
- Encryption: $c = m^e \mod n$, with $n = pq$
- Decryption: $m = c^d \mod n$
- Property: homomorphic $(m_1^e m_2^e = (m_1 m_2)^e)$ can multiply messages, so need to pad and otherwise avoid this.
- Breaking is equivalent to factoring, since n is known.

Diffie-hellman key exchange (w/ elliptic curve)

- Elliptic Curve E mop, $P \in E$
- Alice sends $n_A P$ to Bob
- Bob send $n_B P$ to Alice
- Now have $n_A n_B P$

Man in the Middle attack
Alice sends $g^A \mod p$ which MITM intercepts and sends Bob $g^S \mod B$
Bob sends $g^B \mod p$ which MITM intercepts and sends Alice $g^T \mod p$
MITM now has g^{AT}, $g^{SB} \mod p$ and has an encrypted channel b/w Alice and him and Bob and him
Elgamal cryptosystem Referee
prime p, generator g
Bob
random $x \in 1, 2, ..., (p-2)$
$y = g^x \mod p$
public key (p, g, y); secret key x
Alice
message M, random $k \in 1, 2, ..., (p-2)$
$a = g^k$; $b = My^k \mod p$
transmits $< a, b >$
Bob
$b(a^x)^{-1} = My^k (g^{kx})^{-1} = M(g^x)^k g^{-xk} = M \mod p$

Shamir secret sharing Steps: Make random curve of degree q-1 called f(x) Distribute n points on curve: f(1), f(2), .. , f(n) q points determine curve (not q-1 points!) secret is f(0), which can be any integer mod n
$f(x) = a_{q-1}x^{q-1} + ... + a_1 x + a_0 (modm)$ Share f(1), f(2),...,f(n) q points $\rightarrow$ we can solve for $a_{q-1}, ..., a_1, a_0$ $f(0) = a_0 =$ secret
Shamir secret sharing is (sort-of) homomorphic
$f(x) = a_{q-1}x^{q-1} + ... + a_1 x + a_0 (modm)$
$g(x) = b_{q-1}x^{q-1} + ... + b_1 x + b_0 (modm)$
$h(x) = c_{q-1}x^{q-1} + ... + c_1 x + c_0 (modm)$
We can define: $SUM(x) =$
$(a_{q-1} + b_{q-1} + c_{q-1})x^{q-1} + ... + (a_1 + b_1 + c_1)x + (a_0 + b_0 + c_0)(modm)$
$SUM(0) = a_0 + b_0 + c_0 \mod m$ (sum of secrets)

Elliptic Curves

- Formula $y^2 = x^3 + Ax + b \mod p$
- scalar multiplication same complexity as discrete log problem
- O is special point, infinity
- Number of points is bounded by $|t_p| < 2\sqrt{p}$, where $t_p = p + 1 -$ (# points in E) and p is the prime

Addition rules and properties

- $P \oplus O = P$
- $(x, y) \oplus (x, -y) = O$
- $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$ if $P \neq Q$
- $\lambda = \frac{3x_1^2 + A}{2y_1}$ if $P = Q$
- $P \oplus Q = (x_3, y_3)$
- $x_3 = \lambda^2 - x_1 - x_2$
- $y_3 = \lambda(x_1 - x_3) - y_1$
- $0P = O$, $1P = P$, $2P = P \oplus P$, $3P = P \oplus P \oplus P$, etc...

DES

- feistel, 64 bit blocks, 56 bit keys
- 16 applications of feistel = blocks $(L_0, R_0) \rightarrow (R_0, L_0 \oplus F(R_1, K_1))$
- triple DES more secure, need 2^{57} calculations and a known plaintext attack
- meet in the middle for double des, easily broken

AES

- Rijndael cipher - (DES diagram)

Different modes of Encryption

- CBC - split into blocks, pick init vecotre, XOR vector w/ encrypted block, send. Decryption: XOR decrypted C_i with raw C_{i-1}.
- ECB - codebook. Break msg into blocks, each block has 1:1 map of ciphertext. Good for single values, bad for repetition and if msg aligns on blocks.
- CTR - encrypt counter rather than feedback: $O_i = E_k(i)$, $C_i = P_i xor O_i$
- CFB - stream of cipher feedback. $C_i = P_i \oplus E_k$. $C_{i-1} = IV$
- OFB - output feedback mode. Stream encryption on noise channels. $O_i = E_k(O_{i-1})$, $C_i = P_i xor O_i$, $O_{-1} = IV$

Signatures and Hashes

- To avoid tampering, can send $m, H(m)$, and recipient verifies hash
- If message short enough, can even sign the message itself
- When signing the hash, use known public key, ownership verified via Certificate Authority
- probability of collision needs to be low. if n hash range and k inputs, P(col) $= P(n,k) = 1 - (n!/((n-k)!)n^k) \ = 1 - e^{-k^2/(2n)}$
- preimage resistance - given h, can't find y st $H(y) = h$
- second preimage resistance - given x, can't find $y \neq xstH(y) = H(x)$
- collision resistance - can't find $x \neq ystH(y) = h(x)$

Certificate Authority

- Verifies identity of person, plus their known public key (for encrypting messages and verifying signatures)
- chain of trust - root CA has absolute trust
- can revoke keys when needed or compromised

Rabin Signatures
Encryption:

1. pick p,q,n s.t. $pq = n$
2. publish n as public key
3. pick an m in range $0..(n-1)$ as message
4. $c = m^2 \mod n$. send c

Decryption:

1. Get 4 roots of c, 2 for each factor. $\pm m_p = \sqrt{c} \mod p$ and $\pm m_q = \sqrt{c} \mod q$
2. $https://en.wikipedia.org/wiki/Rabin_cryptosystem$

Elgamal cryptosystem

Exponents Referee prime p, generator g Bob random $x \in 1, 2, ..., (p-2)$ $y = g^x(\mod p)$ public key (p, g, y); secret key x Alice message M, random $k \in 1, 2, ..., (p-2)$ $a = g^k$; $b = My^k($ mod $p)$ transmits $< a, b >$
Bob $b(a^x)^{-1} = My^k(g^{kx})^{-1} = M(g^x)^k g^{-xk} = M(\mod p)$

Elliptic Curves
Referee: elliptic curve $E \mod p$, $P \in E$
Bob random x, $Q = xP$
public key (E, P, Q); secret key x
Alice: message $M \in E$, random k, $A = kP; B = M \oplus kQ$, transmits $< A, B >$
Bob: $B \oplus (-x)A = M \oplus kQ \oplus (-x)kP = M \oplus xkP \oplus (-x)kP = M$

One time Pad

- Need a pre agreed upon pad
- take message, XOR with the pad.
- perfect secrecy, but need huge keys

Pseudo-random number generation

random bits quite valuable

Linear-congruential PRNG Recommended in Knuth p large prime $s_0 \leftarrow$ random seed $s_{i+1} \leftarrow as_i + b \mod p$ $b_i \leftarrow s_i mod2$

Linear-congruential PRNG problems

Linear-congruential PRNG passes most statistical tests of randomness

not good enough for secruity purposes

if we observe $b_1, b_2, \ldots$ can infer constants PRNG equation

Another approach

use encryption:

$s_0 \leftarrow$ random seed

$s_{i+1} \leftarrow Encrypt(s_i)$

$b_i \leftarrow (s_i mod2)$

several technical problems:

computational cost

cycles

Cryptographically strong PRNG

Given sequence of pseudo-random bits, intractable to predict next bit with probability greater than $50\% + o(1/n)$

n is parameter of cryptographic security, such as length of modulus.

Attacks on ciphers

Ciphertext only: Adversary has $E(m_1), E(m_2), \ldots$

Known plaintext: Adversary has $E(m_1)\&m_1, E(m_2)\&m_2, \ldots$

Chosen plaintext (offline): Adversary picks $m_1, m_2, \ldots$, Adversary sees $E(m_1), E(m_2), \ldots$

Chosen plaintext (adaptive): Adversary picks m_1 and sees $E(m_1)$, Then adversary picks m_2 and sees $E(m_2)$

Chosen ciphertext (offline & adaptive): Like chosen-plaintext, but adversary picks $E(m)$

Brute force attacks We can try all possible keys, we can usually recognize valid plaintext. Unicity distance: Minimum number of characters of ciphertext needed for a single intelligible plaintext

Homework 2

$E: y^2 = x^3 + 2x + 1$ List the points of the curve mod 3

x	y^2	y	points
0	1	± 1	(0,1), (0,2)
1	1	± 1	(1,1), (1,2)
2	1	± 1	(2,1), (2,2)

Points: $O, (0,1), (0,2), (1,1), (1,2), (2,1), (2,2)$

Write the addition table for $E \mod 7$

+	O	(0,1)	(0,6)	(1,2)	(1,5)
O	O	(0,1)	(0,6)	(1,2)	(1,5)
(0,1)	(0,1)	(1,5)	O	(0,6)	(1,2)
(0,6)	(0,6)	O	(1,2)	(1,5)	(0,1)
(1,2)	(1,2)	(0,6)	(1,5)	(0,1)	O
(1,5)	(1,5)	(1,2)	(0,1)	O	(0,6)

Let's find our y values:

$p = 1123,\ x = 278$

$y^2 = x^3 + 54x + 87 \mod 1123$

$y^2 = 278^3 + 54(278) + 87$

$y^2 \equiv 216 \mod 1123$

Discussion 2

$E: y^2 = x^3 + 3x + 2 mod 31$

$(2,27) \oplus (3,10) \oplus (3,21)$

By associativity,

$((2,27) \oplus (3,10)) \oplus (3,21) = (2,27) \oplus ((3,10) \oplus (3,21)) = (2,27) \oplus ((3,10) \oplus (3,10)) = (2,27) \oplus O = (2,27)$.

$(3,10) \oplus (2,4) \oplus (3,21)$

By commutativity,

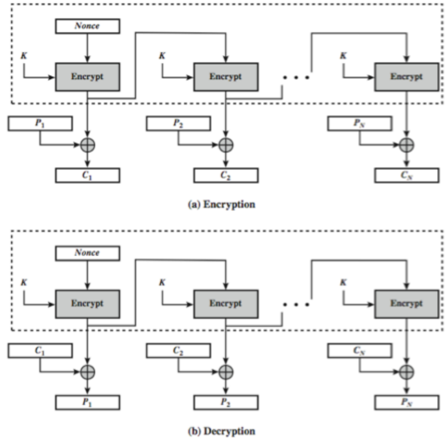
$(3,10) \oplus (2,4) \oplus (3,21) = (2,4) \oplus (3,10) \oplus (3,21)$ By associativity, $(2,4) \oplus (3,10) \oplus (3,21) = (2,4) \oplus ((3,10) \oplus (3,21)) = (2,4)$

1. Relax, GPA does not matter anymore.
2. Think of the cash you'll make after graduation.
3. Do the best you can, and have no regrets!

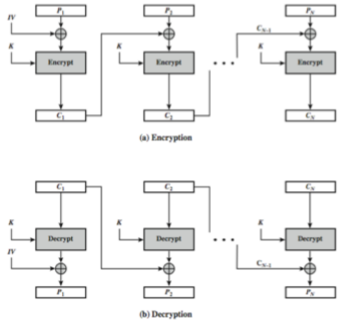
Author: Ivan Smirnov (http://ivansmirnov.name), collaborated with Tsion Behailu (http://tsion.me)

Electronic Codebook Book

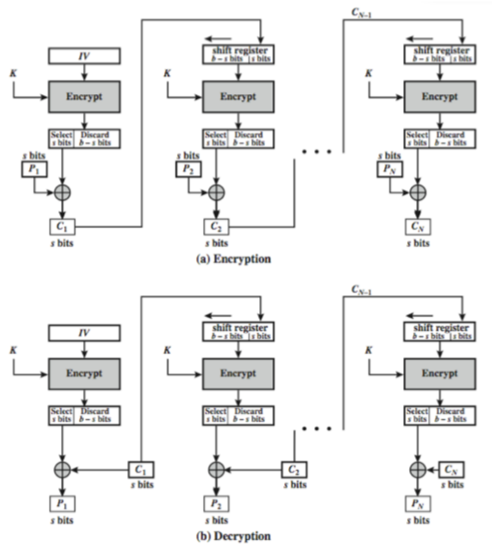
Output FeedBack (OFB)



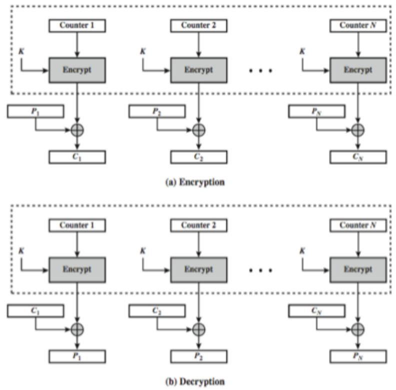
Cipher Block Chaining (CBC)



s-bit Cipher FeedBack (CFB-s)



Counter (CTR)



Man in the Middle