Mathematical Background

Z_n

Defn: $Z_n = \{0, \ldots, n-1\}$

Operations:

① $[a] +_n [b] = (a+b) \bmod n$

② $[a] \cdot_n [b] = (a \cdot b) \bmod n$

Properties:

① $(a+b) \bmod n = a \bmod n + b \bmod n$

② $(a \cdot b) \bmod n = a \bmod n \cdot b \bmod n$

i.e. $a^m \bmod n = [(a \bmod n)^m]$

Z_n^* & $\Phi(n)$

Defn: $Z_n^* : x \in Z_n \wedge \gcd(x, n) = 1$

$\Phi_n : |Z_n^*|$

$\Phi(n)$ calculation rules:

① If p is prime: $\Phi(p) = p-1$

② If p is prime & $k \in \mathbb{Z}^+$: $\Phi(p^k) = p^{k-1}(p-1)$

③ If p & q prime & $\gcd(p, q) = 1$: $\Phi(p \cdot q) = \Phi(p) \cdot \Phi(q)$

④ $x = \prod p_i^{k_i}$ w/ p_i being prime $\forall i$: $\Phi(x) = \prod \Phi(p_i^{k_i})$

Multiplicative inverse

If problem is of the form $ax = 1 \bmod n$ w/ a given, then

① Find $\Phi(n)$

② $x = a^{-1} = a^{\Phi(n)-1} \bmod n$ } Euler

OR

① Use Euclidean algo to find GCD of a, n

i) Write out $a = bq + r$ (r is rem. of $\frac{a}{n}$, $q = a // n$)

ii) Repeat until $r = 0$

② Rewrite 1 as sum of multiples of a & n

③ Take coef. of a, mod w/ $n \rightarrow \underline{\underline{x}}$

Else, we have problem $ax = b \bmod n$

A: If $\gcd(a, n) = 1$

① Find $y = a^{-1}$ using above $\Rightarrow$ Soln of $ay = 1 \bmod n$

② $x = by \bmod n$

B: If $\gcd(a, n) \mid b$

① Find y where y is:

$$\left(\frac{a}{m}\right) \cdot y = 1 \bmod \left(\frac{n}{m}\right) \text{ using above}$$

②

$$x = \left(\frac{b}{m}\right) y + \left(\frac{n}{m}\right) j \bmod n$$

$$\downarrow$$

$$0 \leq j \leq m-1$$

C: Else, no solution

Primitive roots

Defn:

$\text{ord}_n r$: minimum m s.t. $r^m = 1 \bmod n$

Primitive root r: $\text{ord}_n r = \Phi(n)$

Recipe for finding primitive root of n:

① Calculate $\Phi(n)$

② Generate list of possible orders: $m = \{2 \leq i \leq \Phi(n) \text{ s.t. } i \mid \Phi(n)\}$

③ Generate list of possible roots: $r = \{0 \leq i \leq n \text{ s.t. } \gcd(i, n) = 1\}$

④ For each r:

A: Go through each m & see if $r^m = 1 \bmod n$.

B: If $m = \Phi(n)$ & minimum, then r is primitive root.

Galois fields

Defn: $G(q^n)$ is set of polynomials s.t.

$a(x)$ $a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_0$, $a_i \in Z_q$, q is prime
↳ usually 2

Operations:

Addition & subtraction: $c_i = a_i \oplus b_i$

Multiplication: $a(x) \cdot b(x) = \ldots ?$

① Convert a & b to algebraic forms → multiply

② Take result & mod coef. by 2 → $C(x)$

③ Reduce by $p(x)$

$$p(x) \overline{)\, C(x)}$$
$$r(x)$$

OR

i) XOR $c(x)$ by $p(x)$ by aligning w/ left
ii) Repeat until deg $<$ deg$(p(x))$

④ Mod $r(x)$ by 2 → ans.

Totient:

$$\Phi(G(2^n)) = 2^n - 1$$

Multiplicative inverse:

① Find $\Phi(G(2^n))$

② $a^{-1} = a(x)^{\Phi(G(2^n))-1} \mod p(x)$

③ Break up resulting polynomial into degrees $<$ deg$(p(x))$

④ Multiply & reduce

Elliptic curves

Operations: Assume points $P(x,y)$ & $Q(a,b)$

① $-P = (x, -y)$

② $P+Q$:

i) Draw line connecting P & Q

ii) Find intersection point $\rightarrow -A$

iii) Find mirror reflection across x-axis $\rightarrow A$

③ $2P$:

i) Take tangent to P

ii) Find intersection $\Rightarrow 2P$

CLASSICAL CRYPTOGRAPHY

Transposition

① Rail fence

1. Decide on # of lines
2. Write mss in zigzag going up & down
3. Encrypted mss: read each line L→R

② Group transposition

1. Take M & write each letter w/ corr. #
2. Rearrange letters in M according to order set in Π_M

③ Transposition by series

1. Step 1 of group transposition
2. For each Π_{M_i}, create group transposition
3. Join all together

④ Column transposition

1. Write out M in matrix w/ # of col = |key|
2. Rearrange each column by sorting key alphabetically
3. Read each column + join → ciphertext

Substitution

① Monoalphabetic sub.

$$E(m_i) = (a \cdot m_i + b) \bmod n$$

② Vigenère sub.

1. Write out M w/ K right below.
2. For each M_i & K_i, look at Vigenère table to find C_i
3. Join $C_i \rightarrow C$

③ Vernam

$$E(m_i) = m_i \oplus k_j \quad , \quad D(m_i) = c_i \oplus k_j$$

(in bits)

SYMMETRIC ENCRYPTION

Stream ciphers

① LFSR:

1) Take seed and write it horizontally
2) Identify tap bits: take $f(x)$, discard $+1$, & look at which powers have a non-zero coef.

Eg: $f(x) = x^4 + x + 1 \Rightarrow$ tap bits are 4 & 1

3) Shift left & pop-off left most bit → key bit
4) XOR tap bits of prev row → put it as right-most bit

↑ repeat

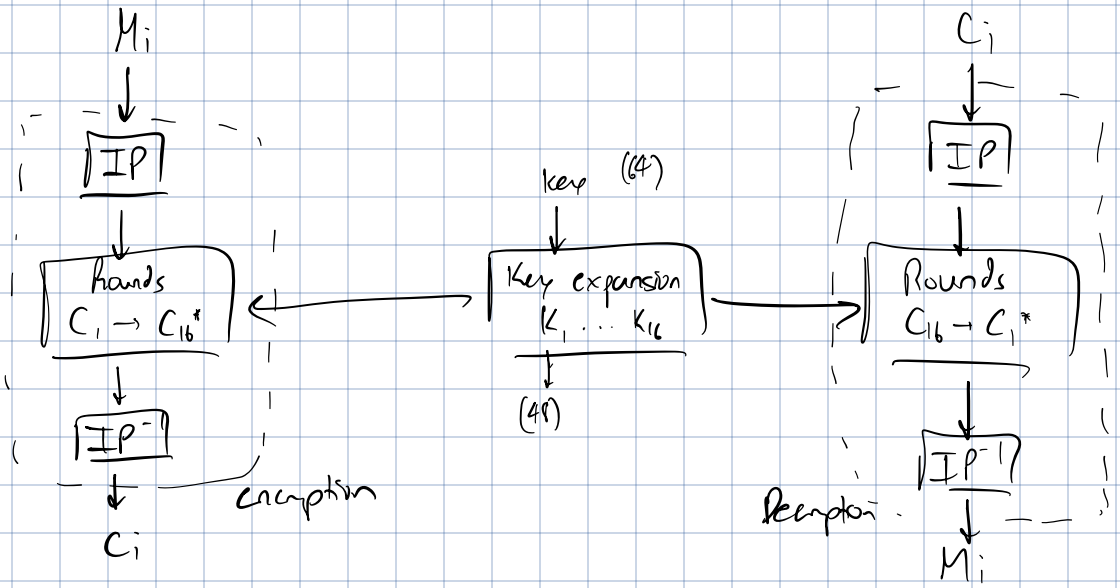
Recall: max T (or period) $= 2^n - 1$ (n = # of bits in seed)

② Encryption: $C_i = K_i \oplus M_i$

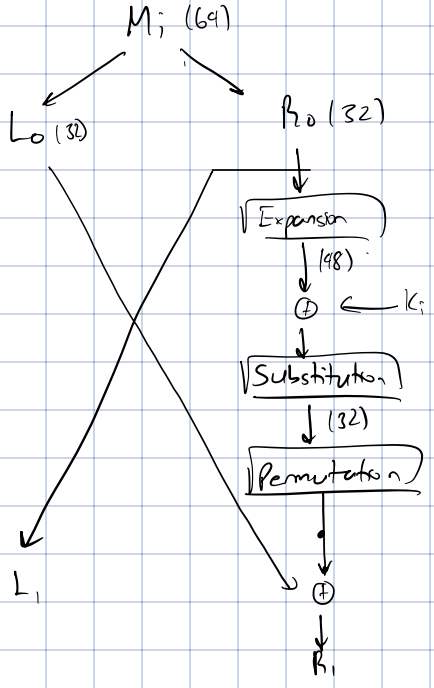
③ Decryption: $M_i = C_i \oplus K_i$

Block ciphers

① DES



Round scheme



1) Expansion / permutation box:

1: Take 32 bits

2: Read E-box L→R, T→D

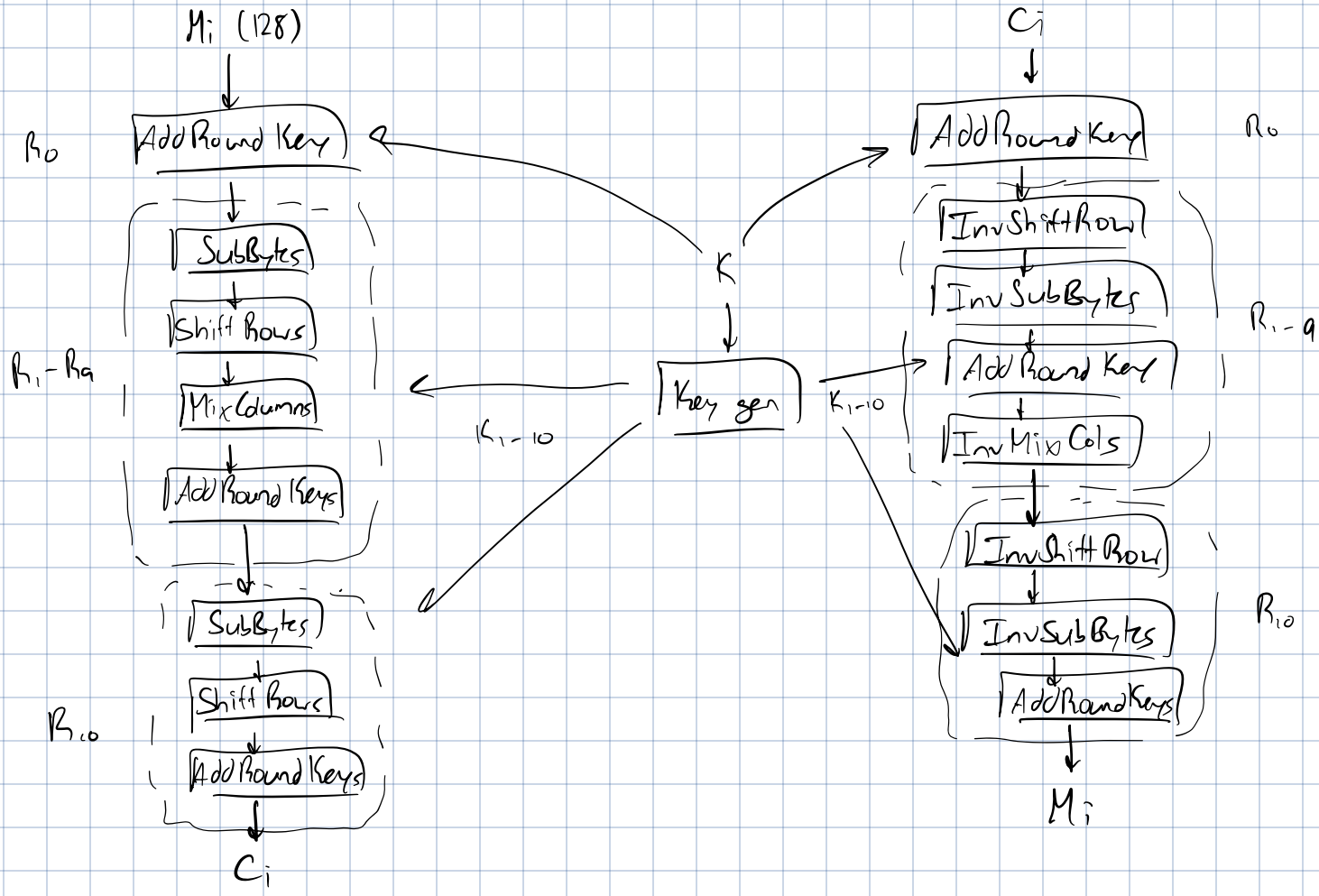
3: For each cell i, take i^{th} bit from 32 bits & mix it down.

2) S-boxes

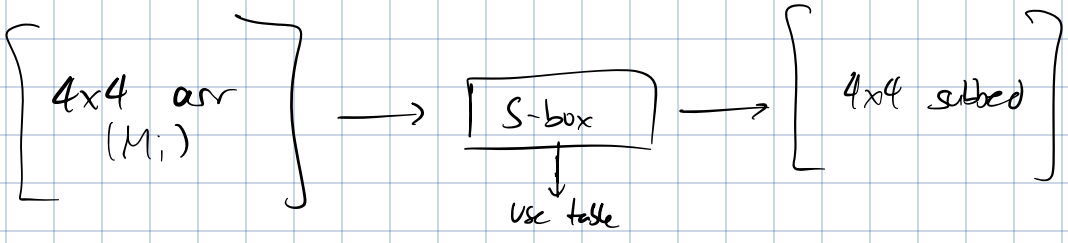
1. Split input into group of 6 bits
2. i = 1
3. For group i, take 1^{st} & last bit → decimal → row
4. Take $2^{nd} - 5^{th}$ bit → decimal → column
5. Look at box i mod 8 & find bits using row & col

i += 1

② AES



1) SubBytes



2) ShiftRows

$$\begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} \rightarrow \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,1} & S_{1,2} & S_{1,3} & S_{1,0} \\ S_{2,2} & S_{2,3} & S_{2,0} & S_{2,1} \\ S_{3,3} & S_{3,0} & S_{3,1} & S_{3,2} \end{bmatrix}$$

3) Mix Columns

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \cdot \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

To get value $s'_{i,j}$:

i) Multiply i^{th} row of matrix w/ j^{th} col of ~~stat~~ matrix & XOR each term

ii) For each term, convert to algebraic → multiply → convert back to binary via reduction of $p(x) = x^8 + x^4 + x^3 + x + 1$

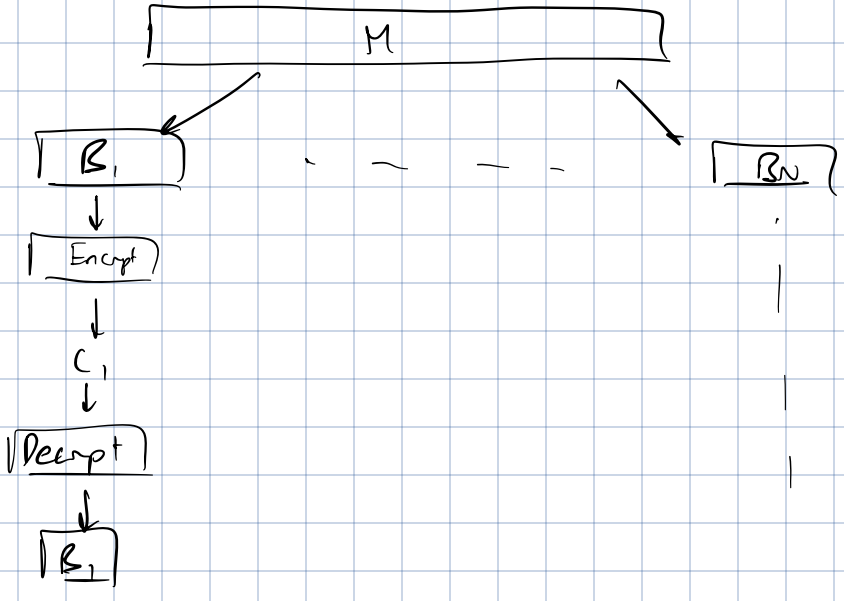
iii) XOR each term together

4) AddRoundKeys

$$\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} \rightarrow \begin{bmatrix} s_{0,0} \oplus k_0 & s_{0,1} \oplus k_4 & s_{0,2} & s_{0,3} \\ s_{1,0} \oplus k_1 & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} \oplus k_2 & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} \oplus k_3 & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$

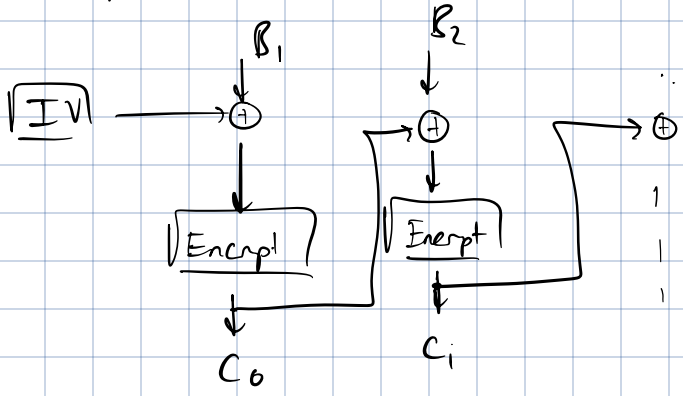
Block cipher operation modes

① Electronic code block (ECB)

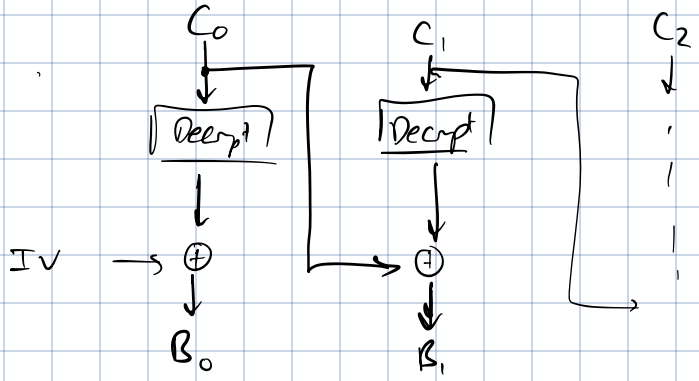


② Cipher block chaining (CBC)

1) Encryption

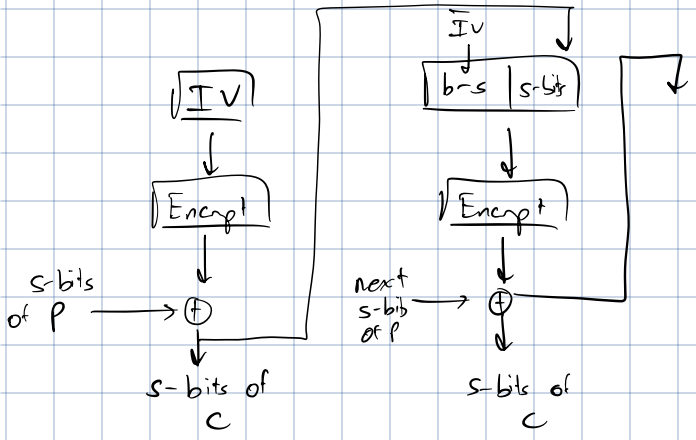


2) Decryption

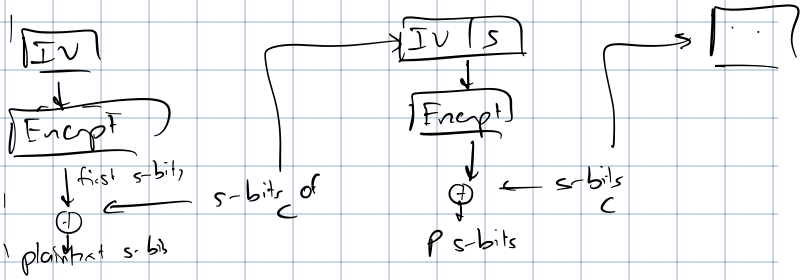


③ Cipher feedback (CFB)

1) Encryption

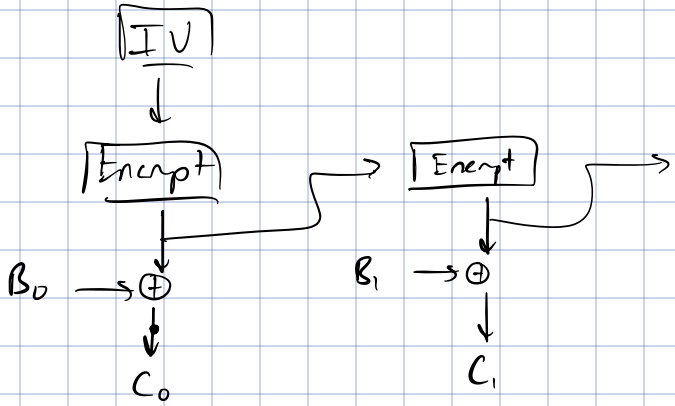


2) Decryption

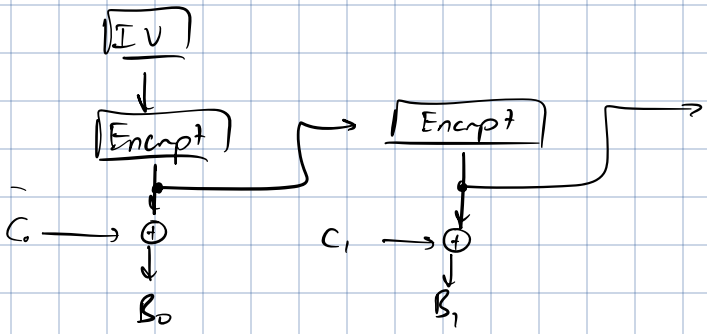


④ Output feedback (OFB)

1) Encryption

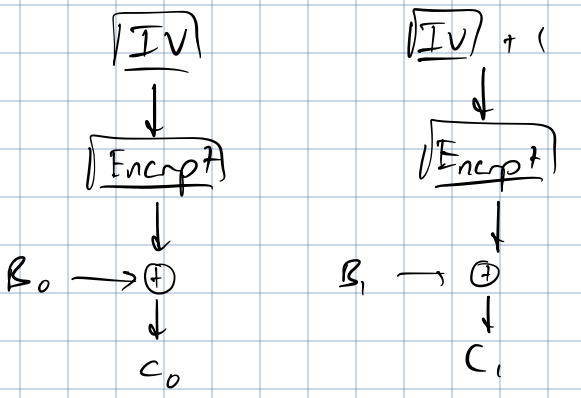


2) Decryption

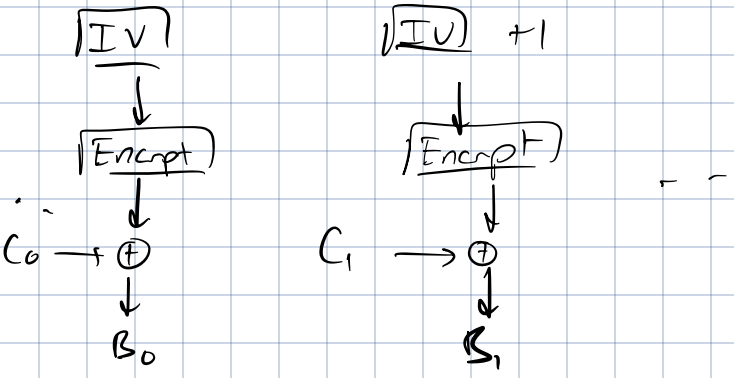


⑤ Counter (CTR)

1) Encryption



2) Decryption



Key Distribution And Asymmetric Encryption

RSA

Process:

① Key gen

1) Choose 2 pimes: p, q

2) Calculate $n = p \cdot q$

3) Calculate $\Phi(n) = (p-1)(q-1)$

4) Choose e: $1 < e < \Phi(n)$ & $\gcd(e, \Phi(n)) = 1$

5) Calculate d: $d \cdot e \bmod \Phi(n) = 1$

② Public & pirate key ctor:

Public: (e, n) Private: (d, n)

③ Message encryption (sender)

$$C = M^e \bmod n$$

④ Message decryption (receiver)

$$M = C^d \bmod n$$

Problems:

1. Deterministic
2. Malleable

RSA-OAEP

Process:

① Padding:

$$X = M \oplus G(r)$$
$$Y = r \oplus H(X)$$

② New msg:

$$M' = X \| Y$$

③ Encrypt:

$$C = (M')^e \mod n$$

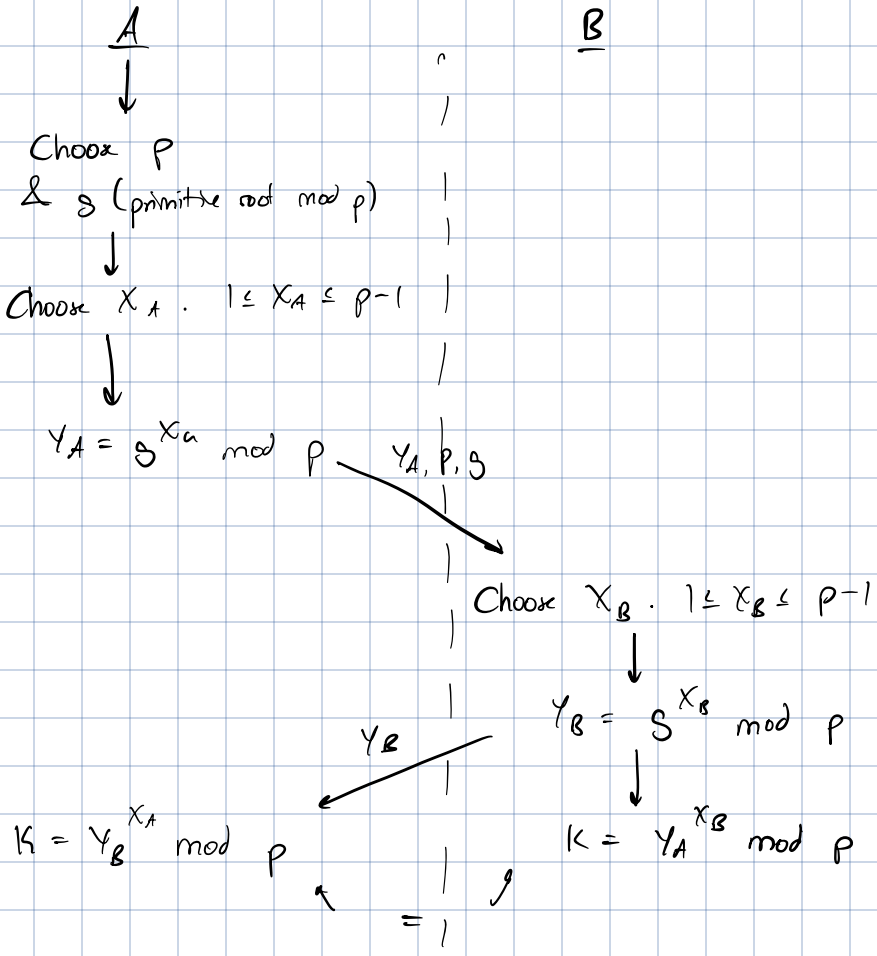
④ Decrypt:

$$M' = C^d \mod n$$

⑤ Extract msg:

$$M' = X \| Y$$
$$r = Y \oplus H(X)$$
$$M = X \oplus G(r)$$

Diffie-Hellman



Hash Functions, MAC, Authenticated Encryption

Attack Probabilities

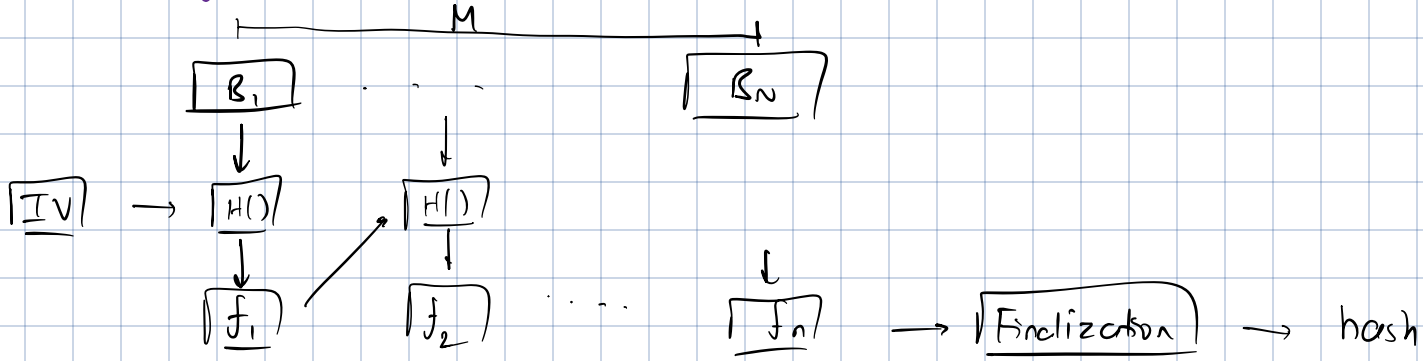
One way / preimage attack: for given h, find M s.t. $H(M) = h \Rightarrow 1/2^n$

Weak collision / second preimage: given M, find $M' \neq M$ s.t. $H(M) = H(M') \Rightarrow 1/2^n$

Strong collision attack: find M & M' s.t. $H(M) = H(M') \Rightarrow 1/2^{n/2}$

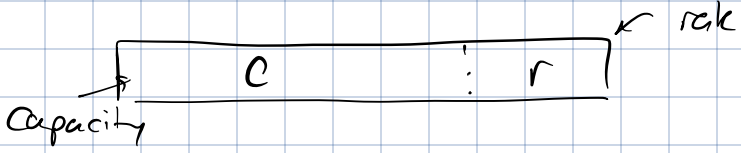
n = length of hash

Merkle-Damgård Hash Construction

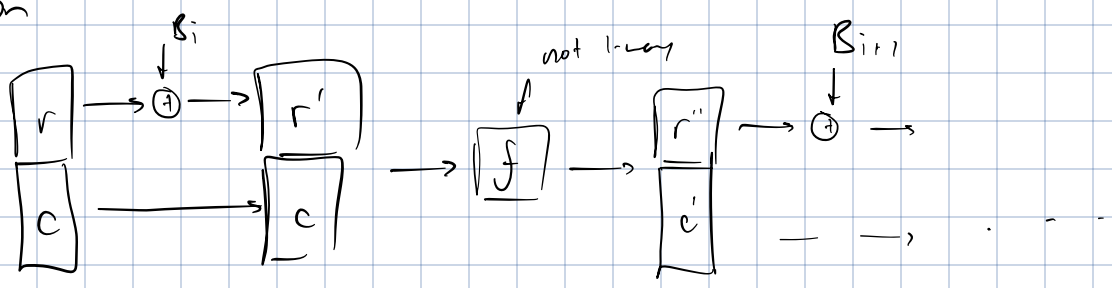


Sponge Hash Construction

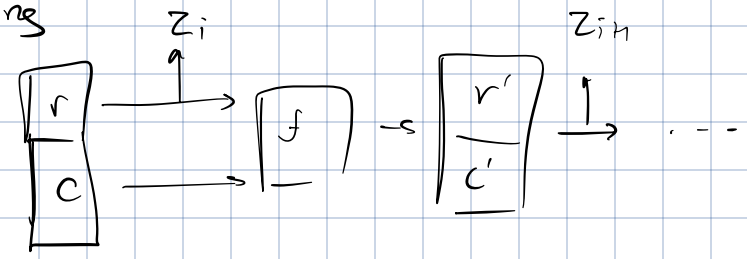
① State init



② Absorption



③ Squeezing



final hash = $\sum z_i$

Hash-based MAC

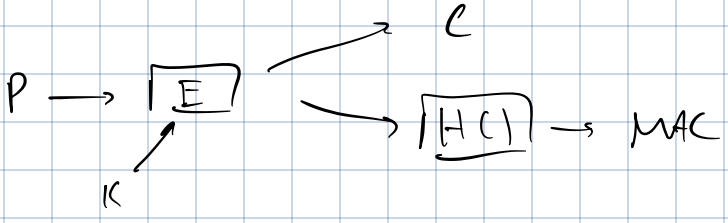
$$HMAC(K, M) = H[(K \oplus opad) \,||\, H[(K \oplus ipad) \,||\, M]]$$

Block-based MAC

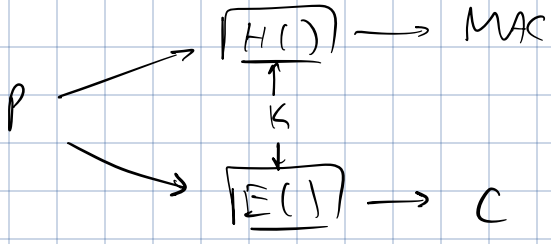
1. Divide M into blocks
2. Encrypt each block w/ block cipher using K, CBC mode of $IV = 0$
3. Last ciphertext block is MAC

Authenticated Encryption Schemes

1. Encrypt-then-MAC

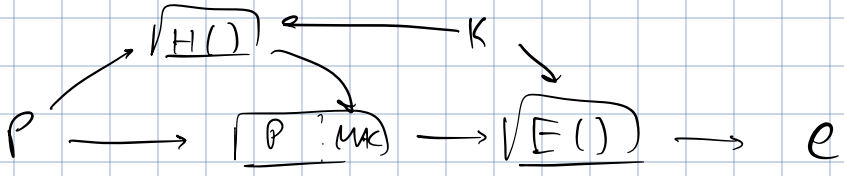


2. Encrypt & MAC



Same text → same MAC

3. MAC-then-encrypt



DIGITAL SIGNATURES

RSA

1. Signer creates public & private keys
2. Signature: $\sigma = M^{d_A} \bmod n_A$
3. Verifier computes $M' = \sigma^{e_A} \bmod n_A$ & verifies if $M == M'$

ElGamal

① Signer computes p_A, g_A, x_A, y_A like Diffie-Hellman

② Private key: x_A
Public key: (p_A, g_A, y_A)

③ Signer choose k_s, $0 < k_s < p_A$, computes $r = g^{k_s} \bmod p_A$

④ Signer computes:

$$\sigma = (M - x_A \cdot r) \cdot k_s^{-1} \bmod (p_A - 1)$$

s.t. $k_s \cdot k_s^{-1} = 1 \bmod (p_A - 1)$

⑤ Verifier computes

$$v_1 = y_A{}^r r^s \bmod p_A$$
$$v_2 = g_A{}^M \bmod p_A$$

$\rightarrow v_1 == v_2$ is verification

PUBLIC KEY INFRASTRUCTURES

Using a certificate

If $A \xrightarrow{M} B$, then:

① B sends C_B to A

② A verifies C_B:
i) Use AC public key to verify C_s (usually public key)

③ A signs M & sends C_A to B

④ B verifies C_A & signature