Cryptography

Diffie Hellmann Key Exchange

→ Key Exchange between two users.
→ not an encryption algo.
→ uses Asymmetric encryption
→ 1976 by names
→ It's a precursor to Asymmetric cryptography.
→ How do one-way function work.

User1 — Red — Yellow
User2 — Blue — Yellow
Mix colors
Red + Blue = Orange ⟷ Yellow + Blue = green

Insecure channel.

Steps in key exchange

I, Choose q and α
↳ a) q → a prime no.
b) Select α as primitive root of q.

▶ How to find primitive root of any value, let say q.

→ $\alpha \in (1, q)$

$\alpha \bmod q$
$\alpha^2 \bmod q$
$\alpha^3 \bmod q$
⋮
$\alpha^{q-1} \bmod q$

$< q$
$(1, 2, 3, \ldots, q-1)$
if these mods produce the above

II Deriving the key pair

User 1	User 2
Assume private key X_a where $X_a < q$	Assume private key $= X_b$. public key (Y_b) $X_b < q$
public key (Y_a)	(Y_b)
$Y_a = \alpha^{X_a} \bmod q$	$Y_b = \alpha^{X_b} \bmod q$
key pair (X_a, Y_a)	key pair (X_b, Y_b)

III Key generation

parameter X_a parameter Y_b, q	parameter X_b, parameter Y_a, q
Secret key generation	& Secret key generation
$K_a = (Y_b)^{X_a} \bmod q$	$K_b = (Y_a)^{X_b} \bmod q$

Must be same

$q = 7$

$\Rightarrow 2 \quad 4 \quad 1$

$3 \rightarrow$	3^1	3^2	3^3	3^4	3^5	3^6	$\perp 6$
	3	2	6	4	5	1	

$\therefore$ 3 is a

(1)

$x_A = 3$

$Y_A = 3^3 \bmod 7$

$= 6$

$\therefore$

key pairs $(3, 6)$

$(Y_B)^{x_A}$

$\Rightarrow (6)^3 \bmod 7$

$\Rightarrow 6^6 \bmod 7$

$\Rightarrow 1$

(2)

$x_B = 4$

$Y_B = 3^4 \bmod 7$

$= 4$

$(3, 4)$

$(Y_A)^{Y_B}$

$(3)^4 \bmod 7$

$(7-1)^4 \bmod 7$

$\Rightarrow 1$

VP

Elgamal cs 9/4/24

Primitive root: A number g is a primitive root modulo n. if every number a coprime to n is congruent to the power of g modulo n.

g, a

if a is coprime to n, then int k for which.

$g^k = a \pmod n$

g primitive root

---

g is primitive root $\forall n$ only if g is a generator of the multiplication group of int. mod n.

$3^1 = 3 \bmod 7$

$3^2 = 2 \bmod 7$

$3^3 = 6 \bmod 7$

$3^4 = 4 \bmod 7$

$3^5 = 5 \bmod 7$

$3^6 = 1 \bmod 7$

$3^7 = 3 \bmod 7$

Eg let $n = 14$.

find out $\forall n^o$ & primitive roots mod n.

$\Rightarrow \{1, 3, 5, 9, 11, 13\}$

$\Rightarrow 3^1 \; 3^2 \; 3^3 \; 3^4 \; 3^5 \; 3^6$

$3 \quad 9 \quad 13 \quad 11 \quad 7 \quad 1$ ✓

5 & 11, 13 ✓

Let $n = 15$

$\phi \{1, 7, 11, 13\}$

2, 4, 8

	1	2	3	4	5	6	7
7	7	4					

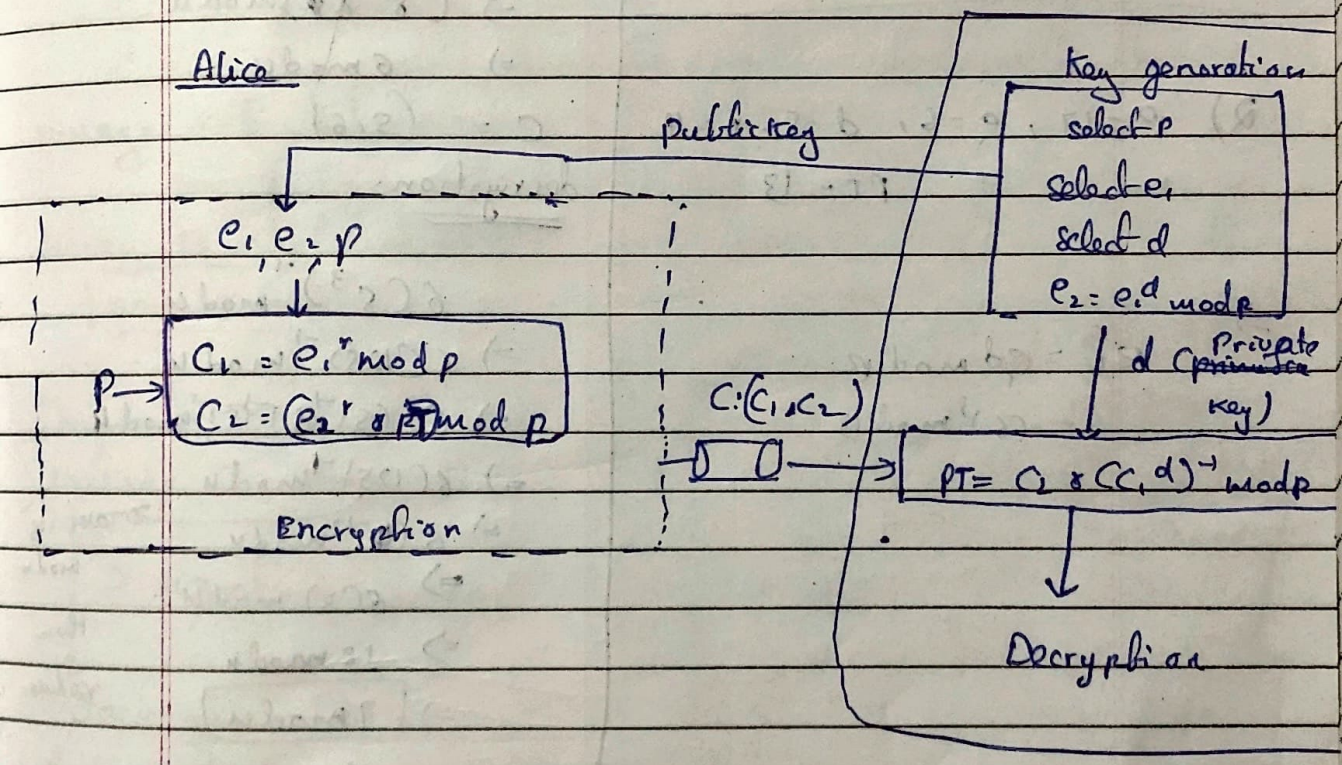
no generator

RC4

Given p, a very large prime
$e_1 \rightarrow$ primitive root in the group $\langle Z_p^*, \otimes \rangle$
& r is an integer.
$e_2 = e_1^r \bmod p$. $\rightarrow$ fast exponentiation (feasible).

But given e_1, e_2 & p, it is infeasible to compute r, which is $r = \log_{e_1} e_2 \bmod p$ which is discrete log problem.

BG.



Encryption (Alice)

$C_1 = e^r \bmod p$

$C_2 = (e_2^r \otimes p) \bmod p$

Decryption. **Proof**

$p \leftarrow C_2 \otimes (C_1^d)^{-1} \bmod p$

$\cdot\ (e_2^r \otimes p) \otimes (e_1^{rd})^{-1} \bmod p$

$(e_1^{dr} \otimes p)(e^{rd})^{-1} \bmod p$

$p \leftarrow p \bmod p$

Q) $p=11, e_1=2, d=3,$
$r=4, \quad PT=7$

(i) what is public key

$e_2 = e_1^d \bmod p$
$= (2)^3 \bmod 11$
$\Rightarrow 8 \bmod 11$

public key = $(2, 8, 11)$

(ii) Encrypt & decrypt the PT = 7

$C_1 = e_1^r \bmod p$
$= (2)^4 \bmod 11$
$= 16 \bmod 11$
$= 5 \bmod 11$

$C_2 = (e_2^r \times PT) \bmod 11$
$\Rightarrow (8^4 \times 7) \bmod 11$
$\Rightarrow 6 \bmod 11$

$C = (5, 6)$

decryption:

$6(5^3)^{-1} \bmod 11$
$\Rightarrow 6(125)^{-1} \bmod 11$
$\Rightarrow 6(125)^{-1} \bmod 11$
$\Rightarrow 6(8^{-1}) \bmod 11$ → Take 125 mod 11 then reduce
$\Rightarrow 6(3) \bmod 11$
$\Rightarrow 18 \bmod 11$
$\Rightarrow 7 \bmod 11$

Q) $p=17, e_1=6, d=5, r=4$
$PT=13$

$\Rightarrow e_2 = e_1^d \bmod p$
$= (6)^5 \bmod 13$
$=$

How to make it strong.

→ randomize r. (p be atleast 1020 bits, 300 decimal digits)

because key gen doesn't make or depend on r.

This avoids chosen PT attacks.

→ also the prime mod must be very large.

10/10/24

Message Integrity:

we studied about secrecy & confidentiality | now integrity |

Documents & fingerprint.
Message & Message Digest

→ signature / fingerprint

→ electronic Equivalent of document & fingerprint pair.

↓ Message, ↓ digest pair

Difference

Current message → Hash function → Current digest → same → message not changed / message changed (← previous digest)

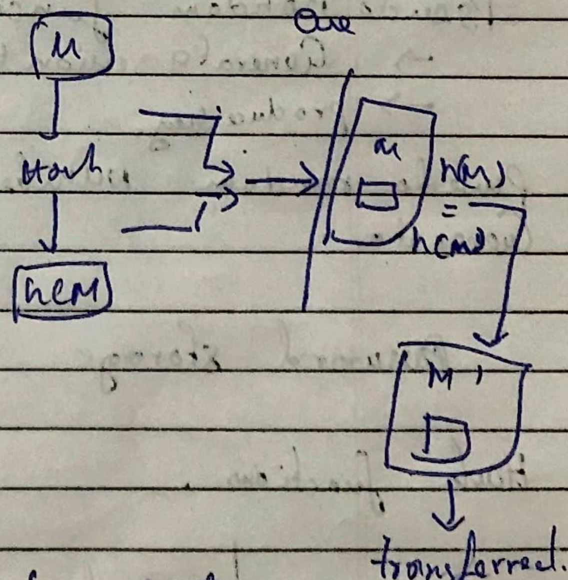
Cryptographic Hash function criteria.

(i) preimage resistance.

Given a $y = h(M)$
it should be impossible to find
M' such that $y = h(M')$

Eve can transfer M' instead.

(ii) second preimage system.



$M' \neq M$
Such that $h(M) = h(M')$

(iii) collision Resistance

M and M' such that
$h(M) = h(M')$
then $M = M'$
(or) not → $M \neq M'$

Hash function use.
→ Message integrity check.

MAC (Message Authentication code)
→ if Integrity + Data origin Authentication
usually Hash only for Integrity.

Hash function & Message Authentication.

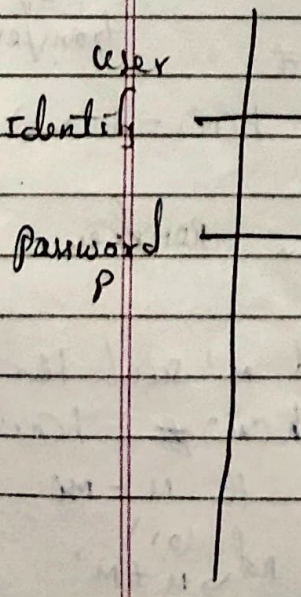
Symmetric key keyed Hash

Pseudo Random function.
→ Generate session key.
→ protect keys.

pseudo random number Generator.

Password storage.

Hash functions.



Data Integrity check.

limitation
→ does not provide assurance about originat
→ Sending different msg & Hash requirement

Variable input size
Fixed output size
Collision resistance
other two.
Efficiency
pseudo randomness.

Most popular.
MD
SHA
RIPEMD
WhirlPool.

Message digest
MD2, MD4, MD5 & MD6

128 bit hash function

SHA
→ SHA 0, 1, 2, 3.
structurally popular
SMDS.
used in SSL

Ruby
PAGE :
DATE :

SHA2 Merkle Damgard construction four variants.

Sufficient.
Till now nothing to worry.

SHA3 → SHA2 ++

SHA 512

RIPEMD 160 replaced by SHA 4 SHA1.

Attack on Hash function
→ collision.

MDC → Modification Detection code.

Sender must create a MDC,
Reciever creates a new one and compares.

MDC → transferred through channel immune to change.

If doesn't give authentication → MD.

HASH vs MAC

I/P	single input	two input.
o/p	Hash/digest	MAC or Tag.
Alg	sensitive	
Security goals	C No	No
	I yes	yes
	A No	yes.
Examples	SHA1, SHA2, SHA3, MD5	HMAC, CBC-MAC
Kind of key	None	Symmetric key.
Application	Password storage	Financial encryption

Need for Message Authentication:
- Disclosure C
- Traffic Analysis C
- Masquerade A
- content M A, I
- sequence M A I
- time M A I
- source repudiation D S
- destination " D S

Message Authentication.

Any MA / DS give.
→ Authentication A
→ Public key encryption

MAC vs Symmetric Encryption.

Symmetric Message Encryption

Public key Message Encryption.
↳ private key encryption is Authentication proof.
↳ then encrypt using public.
then both Authentic & Secrecy.
M → private → public

Message Authentication Code:
→ signature.
→ fixed size Hash.
MAC = C(K, M)
↑ ↑
key message.
we append it to message.
recomputed to check.

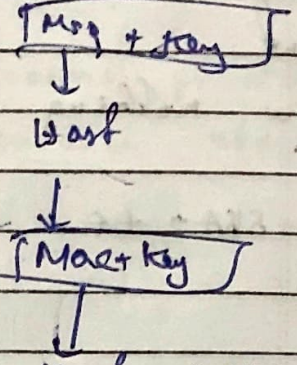
MAC? coz we need only Auth or stronger Auth.

- ▷ Checksum.
- ▷ many to one function?
collision resistant.

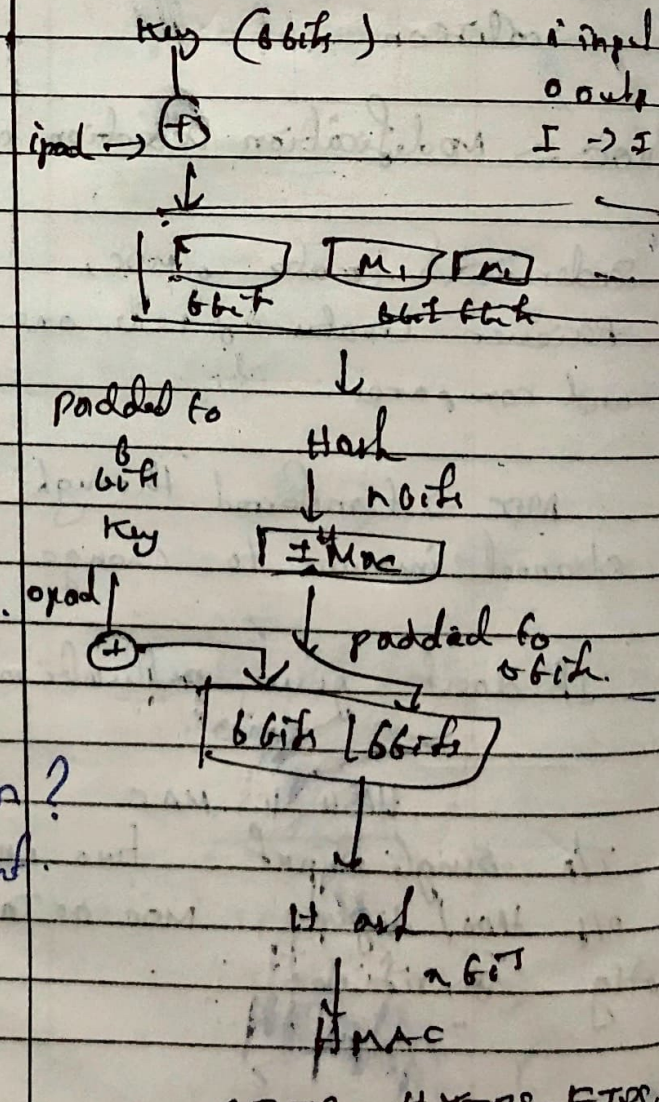
Security on MAC.
Brute force.

Keyed Hash function as MAC.

Keyed MAC

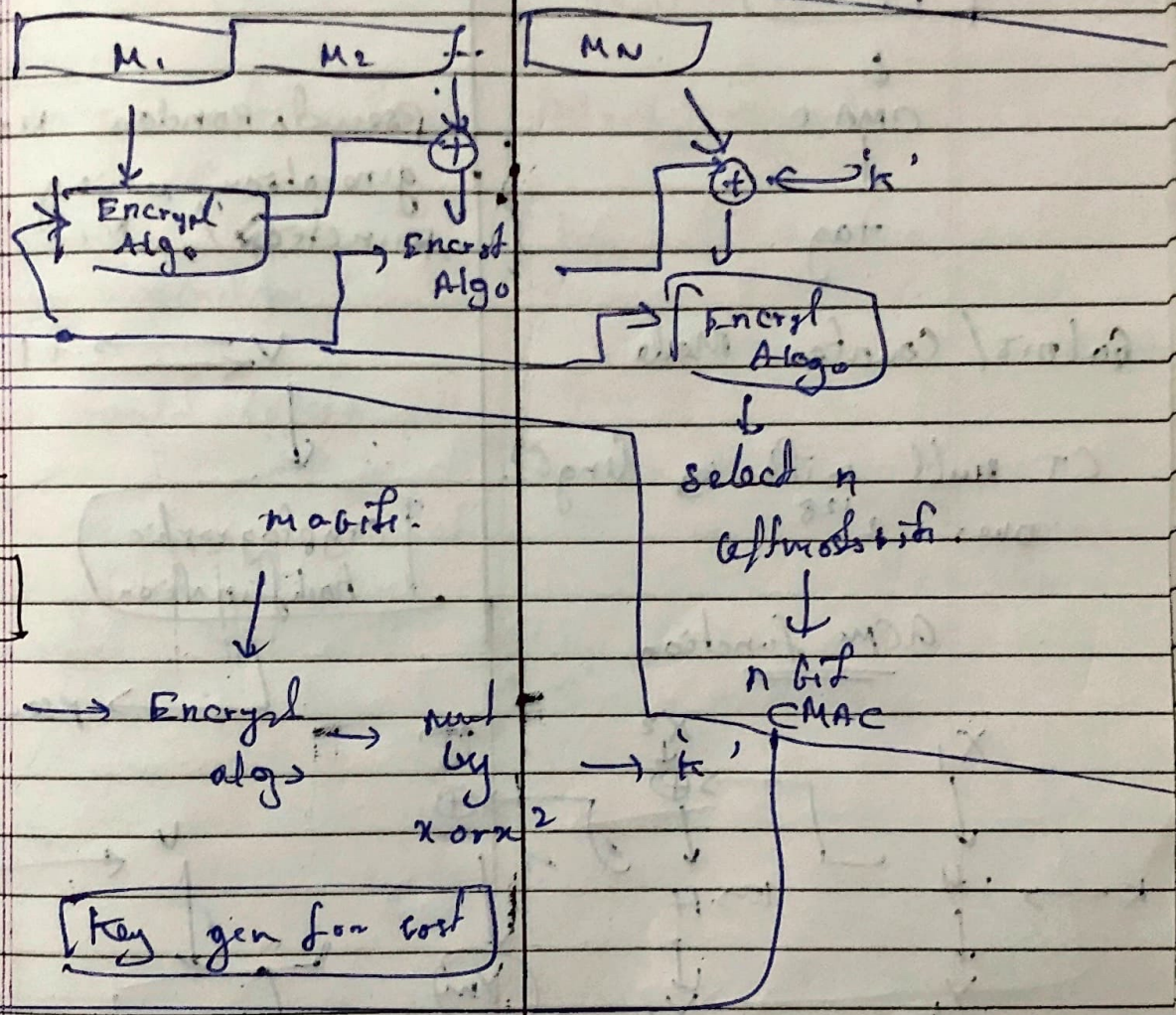


HMAC design:
IP security -
One-way property.



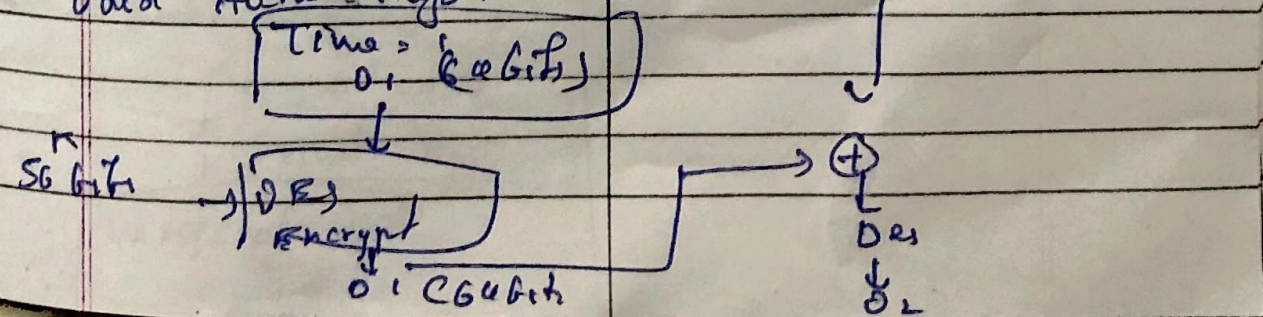
HMAC ⇒ Symmetric keys

CMAC
→ cipher based MAC
→ good / Indust.
One block of MAC from
N blocks of plaintext.



Symmetric cipher for MAC.

Data Auth. Algo.



counter with cipher Block Chaining - Message Auth code.

WIFR

uses AES

CTR

CMAC.

PT

↓

CMAC

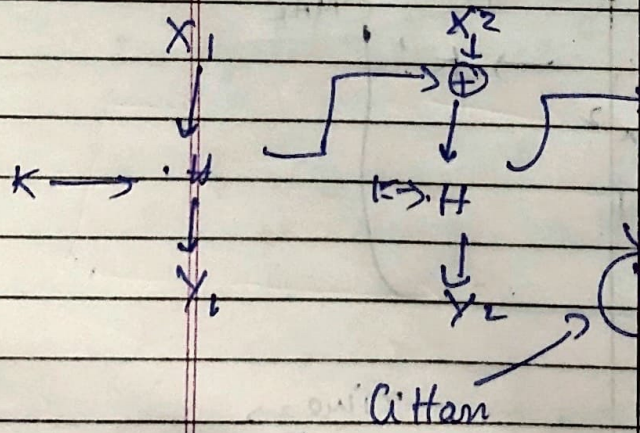
↓

Tag.

Galois/ Counter Mode.

CT Mult with key (ingCL over 2^{128}

ACMI function



ICB → inc → ICB

K → E

M1 → ⊕

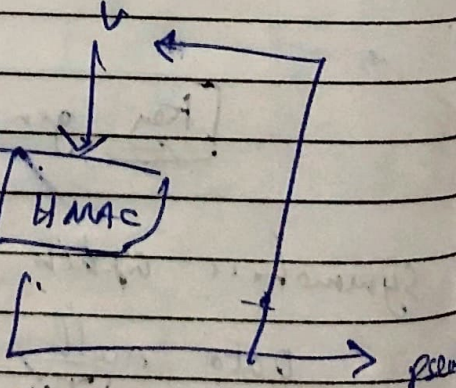
Y1

Pseudo Random Number generation using Hash Function / MAC.

V → +1

Cryptographic hash function

→ pseudorandom o/p.



Digital Signature:
digitally → seperately sent.

verification Method:-
↳ Signature in doc is sign in file.
usual. digitally.
Verification technique

Relationship:
one-to-one relationship.

Duplicity:
no ~~copy~~ distinction between Copies,
time should be stamped.

→ provide non-repudiation
→ MAC like.
→ Encrypt hash with Pvt key.

M
↓
Signing Algo
MS

M
↑
verifying Algo

Ds uses only public-key system.
signed with private key.
verification with public key

Services.
Msg Auth
Msg Integ
Non-repudiation.

for confidentiality → Enc/dec.
not by default.

Trusted order.

Attacks Types

Key only Attack:
Same as CT only attack.

Known Message attack
similar to known PT attack.

Chosen Msg Attack
→ Chosen PT attack.

Forgery Types.

> Existential forgery.

> Selective forgery.

domain or output.

Direct digital signature.
(M+S)+E(M+S)

E(M)+S

Digital Signature scheme.
RSA, ElG, etc.,

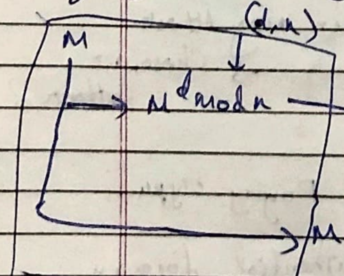
RSA

M: Message (e,n): Alice's public key
S: Signature d: Alice's private key.

(M e, n) → [f(...) mod n] → S
signing

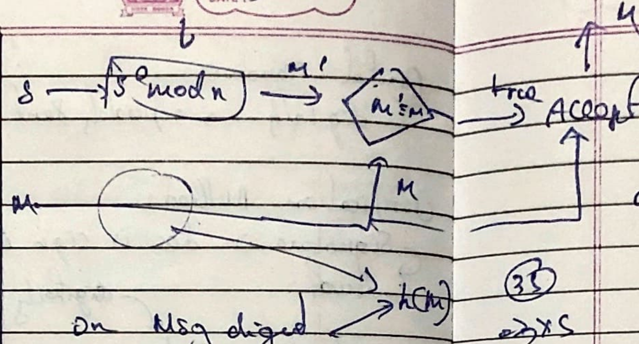
Key generation: done by sender.

Signing and verifying.



signing.

(e,n)



on Msg digest

[M^d mod n]
↓ changed with

→ h(M) → [0^d mod n]

DSS vs RSA. DSS ElGamal
faster smaller.

Variation of D.S.

Time stamped.
Blind signature → sign without knowing the file content

Q)

Public key sys with RSA

C = 10, sent to user
public key is (5, 35)

What is PT.

35
⇒ 7×5
⇒ 6×4
⇒ 24
(24)
8×3
⇒ [3×2^3]
⇒ (2)(8−4)
⇒ 2(4)
⇒ 8
⇒

35×8
24 280
⇒
35×20
35×

$d = e^{-1} \bmod \phi(n)$
$= 5^{-1} \bmod 24$
⇒ $5^7 \bmod 24$

⇒ $5 \times 25 \times 25 \times 25 \bmod 24$
⇒ $(5 \times 1 \times 1 \times 1) \bmod$
⇒ 5

d = 5

$P = (10)^5 \bmod 35$
~~100×100~~
⇒ (100 × 100 × 10) mod 35
⇒ (30 × 30 × 10) mod 35
⇒ (300 × 30) mod 35
⇒ (20 × 30) mod 35
⇒ 600 mod 35
⇒ 5 mod 35.

525