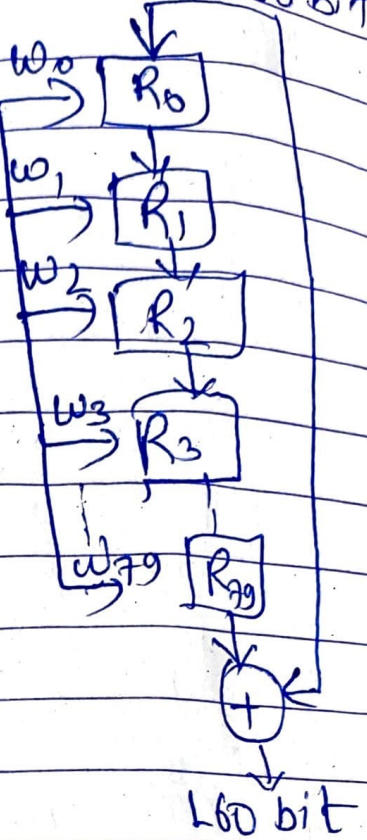
Date:
Page:

ⓒ SHA-1 Algorithm:

Variable length I/P (M) → [SHA-1] → Hash value (160 bits)

$M(m_1, m_2, \ldots, m_n)$ → [MCA] → (16 words) (32 bit each)

448 + 64 = 512 bits



$$K_t = W_{t-16} \oplus W_{t-14} \oplus W_{t-8} \oplus W_{t-3}$$

A = H_0 = 67452301
B = H_1 = EFCDAB89
C = H_2 = 98BADCFE
D = H_3 = 10325476
E = H_4 = C3D2E1F0

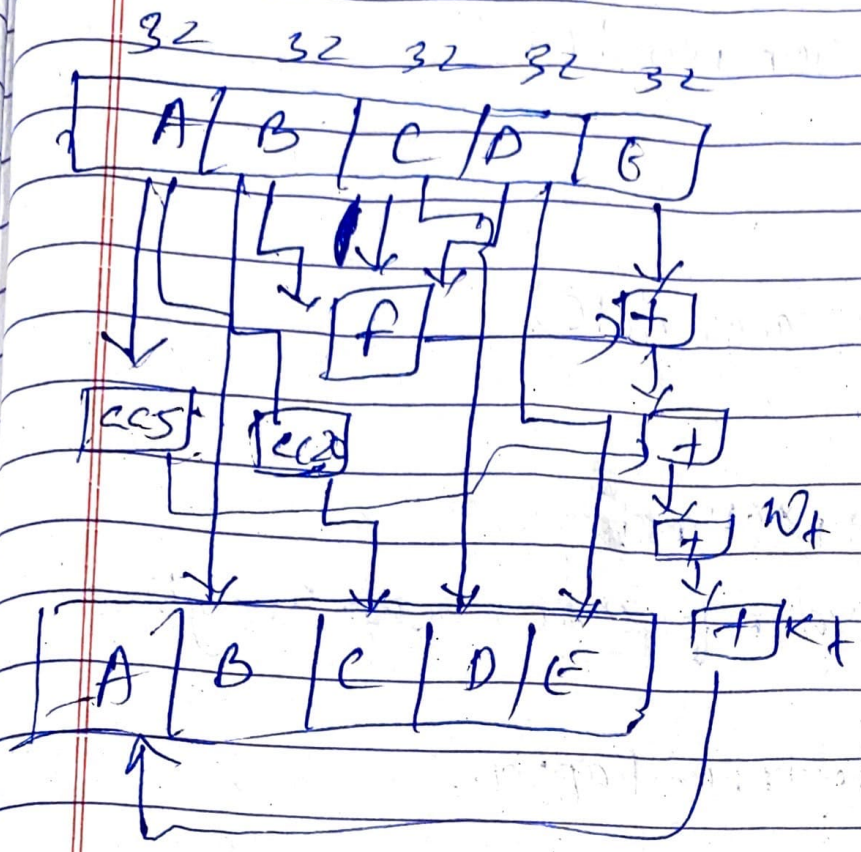
K_{1t} = 5A827999 · $0 \le t \le 19$ → $B \cdot C \wedge \bar{B} \cdot D$
K_{2t} = 6ED9EBA1 $20 \le t \le 39$ → $B \oplus C \oplus D$
K_{3t} = 8F1BBCDC $40 \le t \le 59$ → $B \cdot C \wedge B \cdot D \wedge C \cdot D$
K_{4t} = CA62C1D6 $60 \le t \le 79$ → $B \oplus C \oplus D$

Date:
Page:

→ properties:-

(i) Generating original message from digest

(ii) finding two messages generating same digest



SHA-1	MD5
(i) 160 bits	(i) 128 bits
(ii) 2^{160} operations	(ii) 2^{128} operations to find original mess.
(iii) 2^{80} operations (two msgs with same MD)	(iii) 2^{64} operations
(iv) No such claim	(iv) Some related incident of MD5 break
(v) Slower	(v) faster

Date:
Page:

Message Digest Algorithm:

→ Hash function & Compression function

Features:

(i) fixed length O/P
(ii) Compression fn
(iii) Digest (smaller repⁿ of larger data)

Properties:

(i) M → H (Easy) H → M (Hard)
(ii) M → H } same hash value for same message everytime
n → H
(iii) $M_1 \to H_1$, $M_2 \to H_2$, $H_1 = H_2$ should not happen.

→ MD5:

↳ developed by Ron Rivest
↳ Fast and produces 128-bit message digests

→ Working:

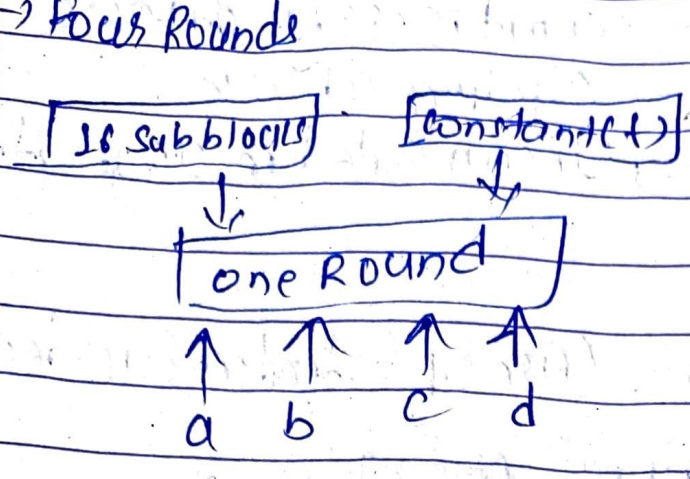
exact multiple of 512 } Original Message | Padding → Original Message | Padding | length → 512 bit block → 512-bit block

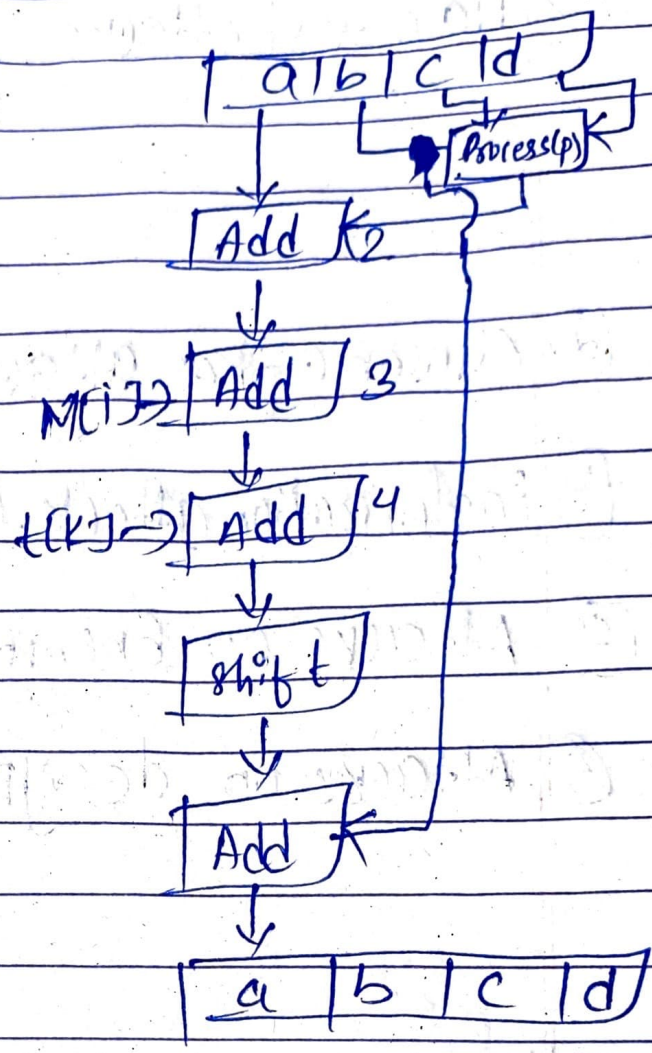
→ Padding done s.t. total length is 64 bit less than exact multiple of 512.

(iv) Initialize four chaining variables (A, B, C, D) (32-bit)

Date:
Page:

(v) Process Blocks: → Copy A=a, B=b, C=c, D=d
→ divide 512-bit block into 16 (32 bits blocks)
→ Four Rounds





Date:
Page:

⑥ Security of RSA:

① Plain text attacks:
- → Short message attack
- → Cycling attack
- → Unconcealed message attack

② Chosen cipher attack: (use of Extended Euclidean Algorithm)

③ Factorization attack: (using finding n ~~using~~ p and q)

④ Attacks on Encryption Key: $(2^{\wedge}16 + 1)$

⑤ Attacks on decryption Key:
- → Revealed description exponent Attack
- → low description exponent attack
- $(D = 2^{\wedge}16 + 1$ (at least))

⑥ Key Management:

It is the process of putting certain standards in place to ensure the security of cryptographic keys in an organization. It deals with:

(i) Generation
(ii) distribution
(iii) use
(iv) storage
(v) ~~[illegible]~~ rotation
(vi) backup/recovery
(vii) Revocation
(viii) Destruction

→ How to
(i) Avoid
(ii) least pr
(iii) HSMs
(iv) Autom
(v) Create a
(vi) Separate
(vii) Split

⑥

→

(i)
(ii)
(iii)
(iv)

Date:
Page:

→ How to do it?

(i) Avoid hard-coding keys
(ii) least privilege
(iii) HSMs
(iv) Automation
(v) Create and Enforce policies
(vi) Separate duties
(vii) Split Keys

thm)

---

① Elgamal Encryption: (Asymmetric Key)

→ Key Generation:

(i) Select large Prime number (P)
(ii) Select decryption Key / private Key (D)
(iii) Select second part of Encryption Key or public Key (E_1)
(iv) Third part of the encryption key or public Key (E_2). $E_2 = E_1^D \bmod P$.
(v) Public Key = (E_1, E_2, P), Private Key = D

xponent Attack
ent attack
1 (at least))

→ Encryption:

(i) Select Random Integer (R)
(ii) $C_1 = E_1^R \bmod P$.
(iii) $C_2 = (PT \times E_2^R) \bmod P$
(iv) $CT = C_1, C_2$

→ Decryption:

$PT = [C_2 \times (C_1^D)^{-1}] \bmod P$

Date:
Page:

⑥ Authentication Requirements:

(I) Revelation:
(II) Analysis of traffic:
(III) Deception:
(IV) Modification in the content
(V) Modification in the sequence
(VI) Modification in the timings
(VII) Source refusal
(VIII) Destination refusal

→ Two functionality levels:
(I) lower level
(II) Higher level

→ Message Authentications Functions:
(I) Message Encryption
(II) Symmetric Encryption
(III) Asymmetric Encryption
↳ Public Key Encryption
↳ MAC
↳ Hash function

→ Measures to deal with these attacks:
(I) Message Confidentiality
(II) Message Authentication
(III) Digital Signatures
(IV) Combination of protocols with DS

Date:.....

Page:.....

① HMAC

→ Used in SSL

→ M —MD5/SHA→ message digest is generated

↓ ← Key(K)

encryption

↓

MAC (CT)

→ from message digest to CT

→ Objectives:

(I) One-way function

(II) less affected by collisions

(III) more secure hash functions

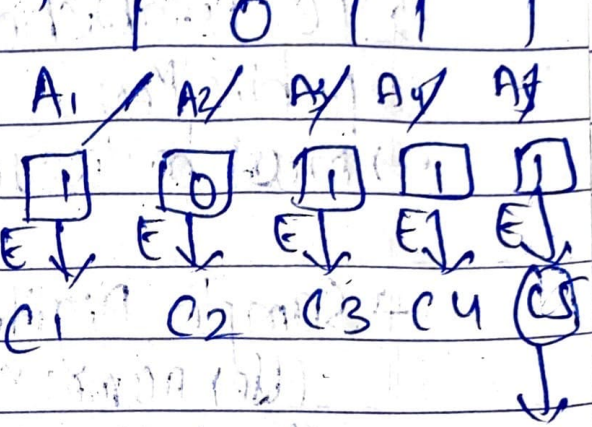
(IV) handle Keys in simple manner.

② CMAC

→ based on block cipher (message size limit)

→ Given message is divided into equal number of blocks and each block is encrypted separately

example:



→ $C_1 = E(K, A_1)$

→ $C_2 = E(K, (A_2 \oplus C_1))$

→ ——

→ ——

→ $C_n = E(K, (A_n \oplus C_{n-1}))$

↓↓

Acts as MAC

Date:
Page:

Public Key Infrastructure: (Standard for Digital Certificates)

- Related to the idea of Asymmetric-Key Cryptography
- includes MD, Digital Signatures and Encryption Services
- To enable all the services Digital Certificates are required

- Digital Certificates:
  - → small file on Computer / Electronic device
  - → File extension is generally .cer
  - → 'DC' establishes the relation b/w a user and the public key.
  - → must be issued by trusted party

→ Sample Digital Certificate

User name: XYZ
Public Key: <12$a#7
Serial No: 12345
Other Inf: Email-Id
Valid from: — — —
Valid TO: — — —
Issued By: — — —

Fields of Digital Certificate

(I) Version
(II) Sig. Algo identifier
(III) Issuer User Id
(IV) CA Digital Signature