Public Key Cryptography

- cont'd. RSA, Elliptical can be used for the usual, and key distribution, signing, Encryption.
- Requirements: easy to enc, dec, keygen → infeasable to break
  → either key can be used to enc/dec...

Diffie Hellman Key Exchange

- Discrete Logarithm Problem: → 1024 bit
  $n = g^k \bmod p$ | n, g are integers, P is prime
- Everyone knows p, g. Random k creates n.

Person A: $PK = g^{K_A} \bmod p$ (sharing Pubkeys) Person B: $PK = g^{K_B} \bmod p$

$K_{AB} = (PK_B)^{K_A} \bmod p$ = shared key (A,B) = $(PK_A)^{K_B} \bmod p$

- Vulnerable to MITM - attacker intercepts all.
  → No host ID. fixed with PK certs.

Totient Function ($\varnothing(n)$) $\varnothing(10) = 4 \leftarrow \{1,3,7,9\}$

- number of numbers < n that are relatively Prime to n (share no common factors)
- Euler: $X^{\varnothing(n)} = 1 \bmod n$; $n = pq$ (comp. of primes)

RSA

|| D=discretionary, M=Mandatory, RB=role Based, AB=Attribute Based

- Choose 2 primes, p, q. Let $n = pq$ therefore $\varnothing(n) = (p-1)(q-1)$. || ACL: By ~~user~~ object
- choose $e < n$ s.t. e is relatively prime to $\varnothing(n)$, and compute d s.t. $d \cdot e \bmod \varnothing n = 1$
  ↳ PK: (e, n) SK: (p, q, d). encipher: $m^e \bmod n = c$. decipher: $c^d \bmod n = m$

Hybrid Encryption: send most using symm, then send symm key using pubkey crypto

1] If S_x with α^*, S_x can transfer α [*]. || * Denotes trans. ability
2] If S_x owns F, S_x may grant any α to any user for F
3] If S_x has control of S_y, S_x may remove α from S_y
4] S_x can copy ACM for files they control / ACM they own
5] Any S can create F and grant any α to F
6] If S_x Owns F, It may delete F
7] S_x may create S_y - then S_x owns S_y
8] If S_x owns S_y, S_x may remove S_y from system

Authentication

- binding of a subject to an ~~sy~~ internal (system) principal
- first, ~~evid~~ claim is made, then evidence is presented (i.e. pw...) || ACM capabilit. → by user

Complementation Information

- all the info you submit when you ~~to~~ sign up or register, [→ comp info] by computer

Password Selection: random phonemes, longer passwords and passphrases are better.

Anderson's PW strength formula: $P = \left(\frac{T \cdot G}{N}\right)$
P = succ prob T = time g = guess/unit time N = num of poss passwords

Password Storage and Bloom Filters

- hash + salt in database [prev. dict. attk]
- Bloom filter: triage bad passwords
  → create array, hash (with K alg's) each bad pw, and set bits in bloom filter. Vulnerable to collisions and will never produce a result of saying bad pw = ok, but might produce the alternatives. || Auth table = subject | Access right | object

Access Control

DB ↔ sysadmin

user ↔ Auth f(x) "PEP" ↔ Resources

authentication, authorization

(whole thing) auditing takes or can take place

- Authentication: bind external entity to a system entity
- Authorization: grant access to resources
- Auditing: independent review of sys actions

DAC{ based on uid ~ access rules; user can chg }
MAC{ Labels (TOP SECRET) assoc w/ proc vs rules; user no chg }
RBAC{ Based on role or group }
ABAC{ Based on attributes and environment } + context of access

Access Control Requirements

- trust inputs - ~~input~~ granularity of access
- Default Closed - Policy conflict resolution
- Admin Policy || setuid | setgid | sticky | o, g, other → no rename or del

Access Control Principles

- least control - separation of duty
- Dual Control: changes to AC* req 2X Admins

Subject || Object || Access Right

CS370 Notecard Lyell Read

Security Notions
- confidentiality: preventing unauthorized access to data (reads)
- Privacy: preventing unauthorized access to personal data (reads)
- Integrity (msg or origin): preventing unauthorized modification of data (writes)
- Availability: timely access to data
- Accountability: trace actions to source.

Sec. mechanisms
- Prevention - Detection
- Response - Recovery

Terminology
- asset: something of value
- threat: circumstances that put sys at danger, i.e. snooping, spoofing repudiation, falsification
- adversary: threat agent that materializes the threat
- Vulnerability: weakness in a system
- attack: when an adversary acts on a vulnerability
- Attack Surface: all (reachable and exploitable) vulnerabilities in a system.

Security Strategy
- Policy/specification - Implement
- Correctness/assurance of - incentives (1-3)

Security Principles
- Economy of ~~Design~~ Mechanism (KISS)
- Fail Closed - Complete Mediation (chk)
- Open Design? - Sep. of Priv: 2 keys/pws @ critical ops
- Least Privilege - Least Common Mechan.
- Psychological Adoptability - Work Factor
- Compromise Recording

Cryptography
- Encryption: confidentiality, privacy
- Hashes: Integrity - MAC: Integrity
- Digital Signatures: Int. N-R. Authenticity

Kerckhoff/Shannon
- Assume Algorithm is public, keys aren't

Attack Types:
- Cyphertext only
  - ↳ find pt / K
- P.t. only
  - ↳ have pt.ct
  - ↳ find K
- Chosen plaintext
  - ↳ pt → ct can be done by adv
  - ↳ find Key
- Chosen ciphertext
  - ↳ inv. of c.pt

Cipher types
- transpositional: scramble
- substitution: shift
- product: both (1), (2)

Stream cipher

$m = b_1 b_2 \cdots \quad E_k(m) = E_k(b_1) E_k(b_2)$

Block Cipher

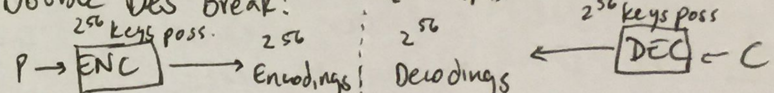
$m = b_1 b_2 \cdots \quad K = k_1 k_2 \cdots$

$E_k(m) = E_{k_1}(b_1) E_{k_2}(b_2)$

Enc Standards
- DES: $\overset{P}{64} + \overset{K}{56} \rightarrow \overset{ct}{64}$
- AES: 128 + n → 128
- 3DES: Do DES 3 times: ENC(D(E))
  - (1) ↳ 112 bit sec, Despite 168 b-K
  - (2) ↳ 2 diff keys, not 3. <80 bit sec

AVALANCHE: change 1 bit causes >50 c.t change

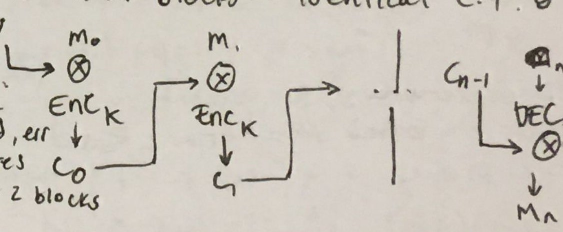
Double DES Break: $C = E_{k_2}(E_{k_1}(p))$



$= 2^{57}$ not 2^{112} find a match and then you know K_1 and K_2

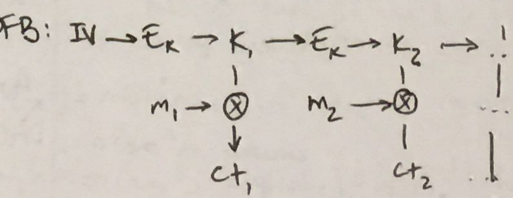
ECB: Do Basic stream cipher on blocks.
↳ issue: identical blocks → identical c.t. blocks

CBC: IV



- self heal: when ct is altered, err propagates only for 2 blocks

OFB:



PK Crypto Basics
- confid: enc PK
- Integrity + Authentic.: enc Priv Key
- Slower than symm
- RSA, Ellip. useful.

Counter: enc(counter$_i$), xor m_n with to make C_i

Hashing Functions
- generates k bits from n, $k \leq n$
- Weak CR: find one match cur
- Strong CR: find two match

Pigeonhole
- If output size < input size, then collisions are unavoid.
- all we need is sizeof(output) +1 to guarantee collision.

Birthday Paradox
- If hash has n bits, $2^{n/2}$ hashes → 50% collis chance

MAC's [H] [ok even if hash is vulnerable to collision]
- Textbook: $h(K \| m \| K)$
- HMAC: ipad 00110011 (·n), opad 01011100 (·n), n = blocksize

$h(K \otimes opad \| h(K \otimes ipad \| m))$