(F) CCA-sec. and ~~MACS~~ MACs.

(F.1) Defining CCA-sec. Enc. and MACs.

(F.2) MACs from PRFs.

(F.3) CBCMAC

(F.4) CCA-sec. Enc from CPA-sec Enc and secure MAC

CBCMAC scheme. Replay attacks are allowed. Why? (Why is replay attack not considered an attack?)

(G) Hashing

(G.1) Defining collision resistance

(G.2) Birthday Attacks

(G.3) Merkle-Damgard Transform

(G.4) Designing hash func^n from DLP.

→ Q $h: \{0,1\}^{2n} \to \{0,1\}^n$

(MDT $H: \{0,1\}^* \to \{0,1\}^n$

Design MDT-like transform for $h: \{0,1\}^{3n} \to \{0,1\}^{2n}$

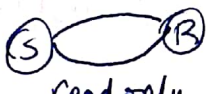
$H: \{0,1\}^* \to \{0,1\}^{\geq 2n}$

(H) Key Exchange

(H.1) Diffie-Hellmann Protocol

(H.2) DDH-assumption

→ Question:



Adv. has read only on one, r/w on other.

S, R does not know which one is which. Design key exchange protocol s.t

LECTURE - XIV

RSA Public Key Cryptosystem

RSA algorithm

→ Gen

choose two large primes p, q and choose (e, d) s.t

$ed \equiv 1 \pmod{(p-1)(q-1)}$ [NOTE: only possible when e, d are co-prime to $(p-1)(q-1)$ and $d = e^{-1}$ in $\mathbb{Z}^*_{(p-1)(q-1)}$]

↳ $ed \equiv 1 \bmod[(p-1)(q-1)]$

Public key $= \langle N, e \rangle$, $N = pq$

Private key $= \langle d, p, q \rangle$

This is an efficient algo, as it only use primality testing, extended Euclid's, etc.

→ Encryption :

~~$m \in \{1, 2, 3, \ldots N-1\}$~~

$m \in \{0, 1, 2, 3, \ldots N-1\}$

$c = m^e \bmod N$ $\quad [O(\log(e))]$

RSA assumption : cannot find m given e. [DLP?]

→ Decryption :

$c \in \{0, 1, 2, 3, \ldots N-1\}$

$m = c^d \bmod N$ $\quad [O(\log d)]$

Note that

$c^d \bmod N = (m^e)^d \bmod N = m^{ed} \bmod N$

Fermat's Little Theorem :-

If $\gcd(a, N) = 1$, then

$a^{\phi(N)} \equiv 1 \quad [\bmod N]$

where $\phi(n)$ is Euler's Totient function i.e no. of numbers $< N$ that are co-prime to N.

~~Here, $ed \equiv$~~ Now,

$m^{\phi(N)} \bmod N = 1$

$\phi(N) = \phi(pq) = pq - p - q + 1$

$= (p-1)(q-1)$

$m^{(p-1)(q-1)} \equiv \bmod N = 1$

$ed \equiv 1 \bmod (p-1)(q-1) \Rightarrow ed = \alpha(p-1)(q-1) + 1$

$\therefore m^{ed} = m^{k(p-1)(q-1)} \cdot m$

$\Rightarrow m^{ed} \bmod N \equiv 1 \cdot m$

$\Rightarrow m^{ed} \bmod N = 1$

Issues: RSA (textbook edition) is deterministic $\Rightarrow$ not CPA-secure.
Also, in public key systems, public key is known to all i.e minimum level of security expected is CPA-security.

↳ Other forms of Attack on RSA systems:-

Let $e = 3$ ($e = 2, 1$ is not possible as $(2,1) \notin \mathbb{Z}_{(p-1)(q-1)}^*$
e small

$\gcd(2, (p-1)(q-1)) = 2$)

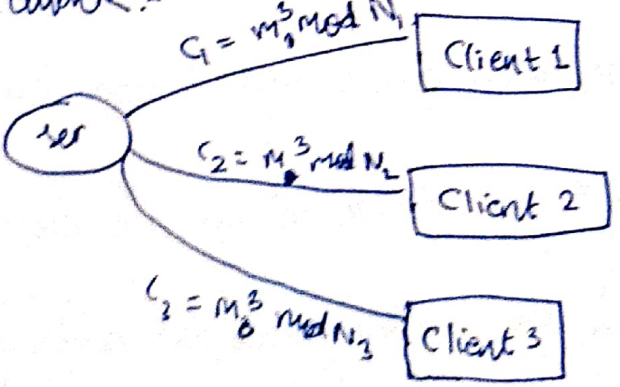
C, N is, $e = 3$

$$c = m^3 \bmod N$$

If $m < n^{1/3}$, $c = m^3 \Rightarrow m = c^{1/3}$

Now, if d is small, similar attacks possible.

~~one~~ Another attack:-



$e = 3$

N_1, N_2, N_3

These clients, same message to all.

$\therefore x \equiv c_1 \pmod{N_1}$

$x \equiv c_2 \pmod{N_2}$

$x \equiv c_3 \pmod{N_3}$

From Chinese Remainder Theorem, we can find unique $x \in [0, N_1 N_2 N_3 - 1]$

Now, $x = m^3 \Rightarrow m = \sqrt[3]{x}$

↳ Public Key Cryptographic System [Standard - PKCS v1.5]

Gen: same as RSA

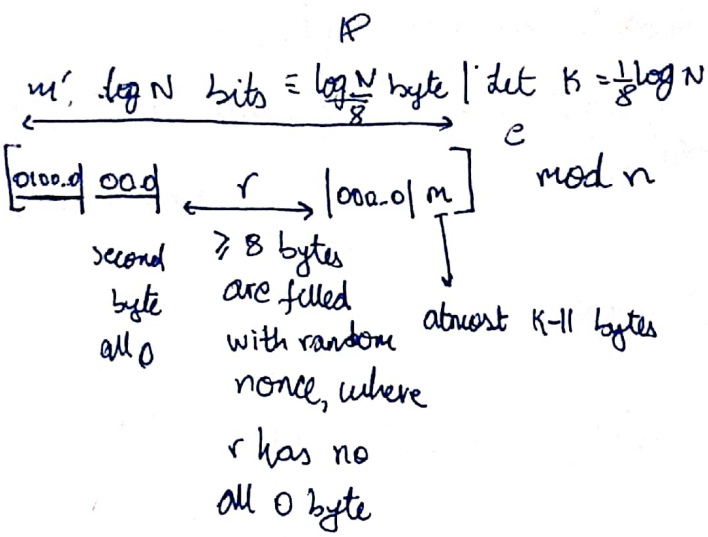
P, q primes

$ed \equiv 1 \pmod{(p-1)(q-1)}$

Private key = $\langle N, e \rangle$, $N = pq$

Public key = $\langle d, p, q \rangle$

Encryption:



Decryption: Do $c^d \bmod N$, then ignore first 2 bytes, then keep removing bytes until we reach a zero byte. After this byte, we have message.

RSA - OAEP: Optimal Asymmetric Encryption padding.

↳ RSA - Digital Signatures: 'Textbook' version:

A ⟶ (m, σ_A) ⟶ B

σ_A : Signature of A.

sk: secret key

pk: private key

$\sigma_A = sign(sk_A, m)$

$Vrfy(m, \sigma_A, PK_A) = Y/N$

RSA signature:

$\text{Sign}(d, m) = m^d \bmod N = \sigma$

Verify:

$\text{Vrfy}(m, \sigma, N, e) = Y$ if $\sigma^e \bmod N = m$, else N

This scheme is breakable:

Attacks: Pick σ, and send

$m_\sigma = \langle \sigma^e \bmod N, \sigma \rangle$ 'from A' $\longrightarrow$ B

$\text{Vrfy}(m, \sigma, P_A, N) = Yes$

Improved attack: Sender signs two messages.

$\langle m_1, \sigma_1 \rangle$

$\langle m_2, \sigma_2 \rangle$

Adv. generates third message:

$\langle m_1 m_2, \sigma_1 \sigma_2 \rangle$

(multi. mod N)

$\text{Verify}(m_1 m_2, \sigma_1 \sigma_2, N, e) = m_1 m_2 \; \sigma_1^e \sigma_2^e \pmod N$

$= m_1 m_2$

↳ Hash-and-Sign Paradigm

$\text{Sign}(d, m) = \sigma = [H(M)]^d \bmod N$, where H is a hash function

$\text{Vrfy}(m, \sigma, N, e) = Y$ if $\sigma^e \bmod N = H(m)$

N otherwise

Above model is proved to be correct in Random oracle model.

LECTURE- XI

↳ Digital signatures & Zero Knowledge Proofs

How to Authenticate the user?

We ask the user to prove that he/she knows the secret key.

Reveal the secret key? [valid proof, but not reusable].

∴ We ask the user to prove something that ~~you have~~ he has, but no one else has [ZERO KNOWLEDGE Proof for e.g. Alien learning to see more colours than humans].

[SEE: PSPACE ≠ P]
↓ polynomial in space
↓ polynomial in time

Digital signatures are Zero Knowledge Proofs.

Theorem:- ~~Zero~~ One way functions ⇒ Zero knowledge proof

↳ 'Cryptographic Firewall'

→ Bit Commitment

Binding Property: → b

Blinding Property: b is secret.

We create a protocol that can commit to a bit b and later reveal it. By blinding property, we want b to be unknown b/w hiding and reveal phase.

→ One-way permutation ⇒ Bit commitment

$f(\cdot)$ e.g. $f(x) = g^x \bmod p$

We create protocol as follows:-

(i) ~~Pick a~~ Commit Phase

Pick a random s.

Publish $\langle f(s), h(s) \oplus b \rangle$, where $h(s)$ is hard core predicate

(e.g. $\langle DLP(s), MSB(s) \oplus b \rangle$)

This has blinding property, as, by the fact that $h(s)$ is hard-core predicate, no one knows (with $> \frac{1}{2} + neg(n)$) prob.) any info. about $h(s) \oplus b$.

(ii) Reveal Phase.

Give b and s.

Check: If $f(s)$ is unchanged and if so, if $h(s) \oplus (h(s) \oplus b) = b$

∴ Digitally, we implemented an opaque box that can be locked [i.e. f is OWF]

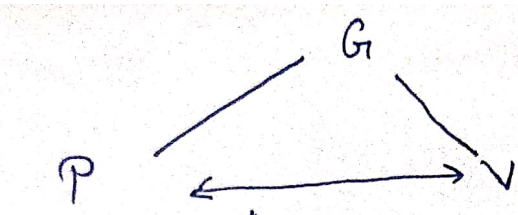
Now, given a bit commitment scheme, we give a ZKP of an NP-complete problem, thus showing any NP-complete problem admits a ZKP.
[3-colouring]

~~Proof~~ Graph 3-Colouring: NP-complete

Given $G = (V, E)$. Is G 3-colourable?

Note: n-colourable ≡ ~~partition~~ is graph tripartite?

Our problem (for which we want ZKP):- We know the sol^n (i.e yes or no), and we which to convince everyone that we know.



(prover, provides 3-colouring if available)

C1: Completeness condition: V accepts if G is 3-colourable

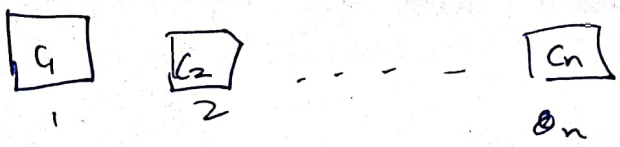
C2: Soundness ''''' : V does not accept with high prob. that if G is not 3colourable

Zero Knowledge condition: V does not get learn anything extra during / after prod.

Code for prover: 3 colors RGB, some colouring

→ Permutes the three colours.

→ Creates n-locked boxes, each with the color of corresponding vertex, and send to V.



Code for V:

→ V picks one edge (i, j) at random and asks to reveal c_i and c_j. [Prover cannot change colours now].

→ If $c_i = c_j$: Reject

if $c_i \neq c_j$: Accept

(V repeats this)

Completeness condition: Note that, V will always accept.

Soundness condition: If G isn't 3-colourable, V rejects with high prob.

~~After k runs~~ In one run

$$Pr[\text{reject}] = \frac{1}{|E|} \Rightarrow Pr[\text{accept}] = 1 - \frac{1}{|E|}$$

∴ After k runs

$$Pr[\text{Accept}] = \left(1 - \frac{1}{\quad}\right)^k = \text{negl}(k)$$

∴ Prob. of accepting is negligible ⇒ V rejects with high prob. if P is false

~~Also, Note that~~

Zero Knowledge condition: After experimenting, with high prob. we know whether prover knows or not

∴ We know OWF ⇒ ZKP ⇒ Digital signature

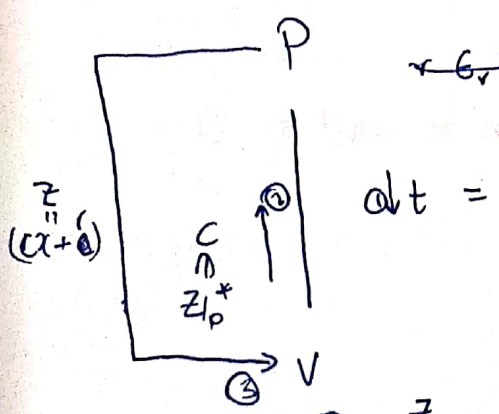
Now, we create a Digital Signature from a ZKP.

e.g. ZKP from DLP:

$$y = g^x \bmod p$$

~~We want V to know~~ to V

P wants to show that it knows x, given g, y, p [PUBLIC DATA] without revealing x, or any other knowledge about x.



④ If $g^z = y^c \cdot t$ accept

else reject

Completeness :- If P knows x, it knows c, r and returns $z = cx+r$ correctly

Also, V always accepts, as $g^z = g^{cx+r} = y^c \cdot t$

Soundness :- If P doesn't know x, then P would have to output

choose some x', r' such that $cx' + r' = cx + r$

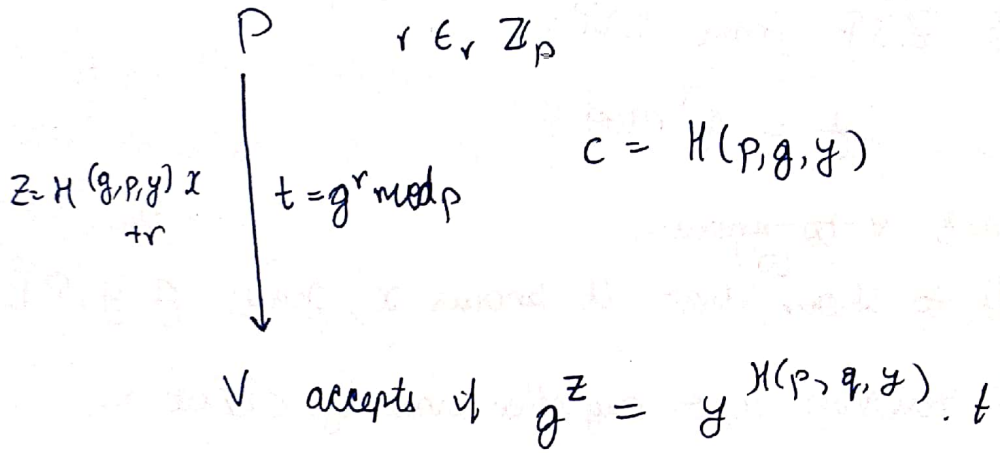
⇒ prob. is atmost $O\left(\frac{1}{p^2}\right)$ ⇒ prob. of acceptance by V is very small.

Zero knowledge : No info passed by the fact that DLP is OWF.

Non-interactive Proof: All info flows from P to V.

Above proof is interactive, as V passes C to P.

Now, if c is chosen by P, then P would have to prove to V that c is chosen randomly. This can be circumvented by using hash functions, assuming that Hash Function outputs randomly [RANDOM ORACLE MODEL].



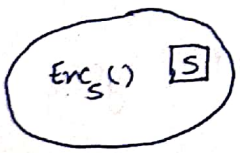
Since this has now become a non-interactive proof, it can be used as a Digital signature.

LECTURE - XVI

SECRET SHARING

→ The Problem

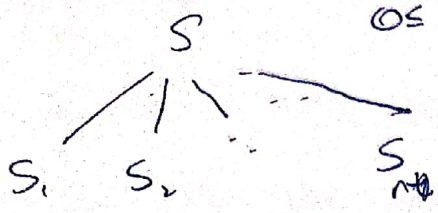
Secret key S



If S is corrupted, all access to our information is lost, if the key is in one place.

But, if you distribute the key, it no longer remains secret.

∴ Tradeoff b/w secrecy and availability.



$0 \le t \le n$

* Any set of $\geq (t+1)$ S_i's can reconstruct S.
* Any set of $\leq t$ S_i's have NO info about S.

t : parameter.

→ Shamir's Secret Sharing :-

Define $s \in \mathbb{F}$ (for ex. $\mathbb{Z}_p$, prime p)

$$Q(x) S_{i+1} = \sum_{j=0}^{t} r_j x^j \quad ; \quad Q : \mathbb{F} \to \mathbb{F}$$

$\alpha_1, \alpha_2, \ldots - \alpha_n$ are distinct public elements of $\mathbb{F}$.

$$S_i = Q(\alpha_i), \; Q(0) = r_0 = S, \; r_j > 0 \xleftarrow{rand} \mathbb{F}$$

We can view it as matrix multiplication with Vandermonde Matrix.
If we have $\leq t$ variables, then we can prove that it has uniform distribution and no info is obtained.

→ General Access Structure

Let $\mathbb{A} \subseteq 2^{\{1,2,\ldots n\}}$

some subsets $B \in \mathbb{A}$ can access S.
Others should have no ~~subset~~ information about S.

Eg. for secret sharing

$B \in \mathbb{A}$ if $|B| \geq t+1$

else $B \notin \mathbb{A}$

i.e

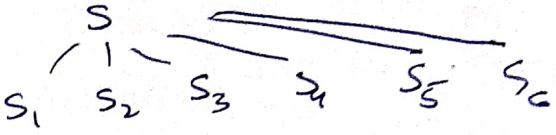
$$\mathbb{A} = \{ B \mid |B| \geq t+1 \}$$

~~e.g. n = 5~~ Subsets $\in \mathbb{A}$ should be able to access.
Otherwise, no access.

But, this is more general.

e.g. we can define following access structure:

$n = 6$



$$A = \{ \{1,2\}, \{3,4,5,6\}, \{2,3,5\} \}$$

these are the bases of A

~~e.g. degree t=3 and~~ t

~~topology with access~~
~~topology~~

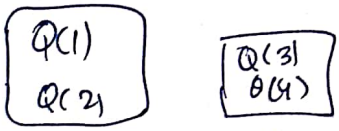
↓

$\{1,2,4\}$?

To construct a solution to this use the following generalization of Shamir:-

degree $t = 3$ and 8 points

we assign ~~8~~ weights - 1,2 should have 2 keys, each, and others should have only one.



$S_3 = Q(5)$
$S_4 = Q(6)$
$S_5 = Q(7)$
$S_6 = Q(8)$

Now, 1 and 2 together can unlock S.

∴ We assign WEIGHTS to different people.

Problem:- Can we solve all ACCESS structure problems using WEIGHTS?

Answer:- NO.

In fact, many of the times, exponential weights are needed, so this is not a good way.

→ Relation with computing

$\mathbb{A}$:- access structure

$f : \{0,1\}^n \to \{0,1\}$ can describe $\mathbb{A}$, as there is a 1-1 relation between $\{0,1\}^n$ and $\mathbb{P}(\{0,1,2,\ldots n\})$

$$\therefore \quad f \longleftrightarrow \mathbb{A}_f = \{B \mid f(B)=1\}$$

Any $\mathbb{A} \longleftrightarrow f_{\mathbb{A}}(B) = \begin{cases} 1, & \text{if } B \in \mathbb{A} \\ 0, & \text{otherwise} \end{cases}$

e.g. for simple secret sharing,

$$f(B) = \begin{cases} 1 & \text{if } |B| \geq t+1 \\ 0 & \text{otherwise} \end{cases}$$

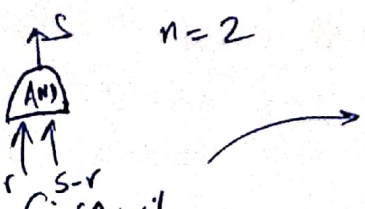
But access structures are monotone i.e if some subset is in $\mathbb{A}$, then its supersets always lies in $\mathbb{A}$.

$\therefore f_{\mathbb{A}}$ must also be MONOTONE.

Equivalence Theorem :- f is MONOTONE iff it can be implemented using only AND and OR's.

∴ We can write ACCESS Structure in terms of AND and OR.

→ FROM AND, OR circuits for $\mathbb{A}$ to Secret sharing scheme for $\mathbb{A}$:-



Circuit

$n=2$

$\mathbb{A} = \{\{1,2\}\}$

$\overline{\mathbb{A}} = \{\emptyset, \{1\}, \{2\}\}$

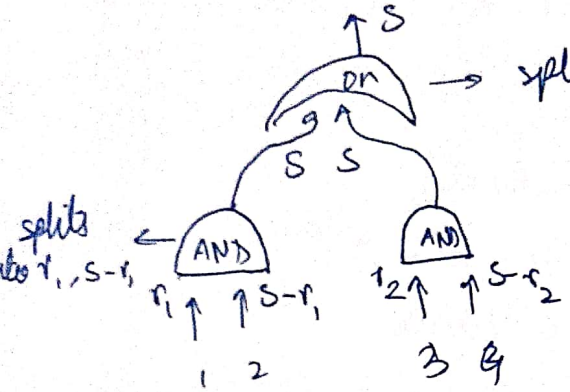
∴ Secret sharing scheme :- Take f and s-r.

$\mathbb{A} = \{\{1\}, \{2\}, \{1,2\}\}$

$\overline{\mathbb{A}} = \{\phi\}$

Here secret sharing scheme is S, S.

Now, consider the following : We apply it recursively.



$\mathbb{A} = \{\{1,2\}, \{3,4\}, \{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}, \{1,2,3,4\}\}$

∴ Solving General Access Structure :-

Given $\mathbb{A}$:

① Model $\mathbb{A}$ as a Boolean function $f_{\mathbb{A}}$

② Design an AND-OR Boolean circuit for $f_{\mathbb{A}}$ (always possible as $f_{\mathbb{A}}$ is monotonic).

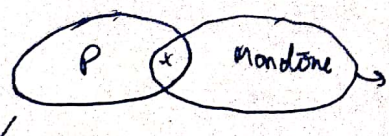
③ Build the secret sharing scheme level by level.

(eg: if you encounter AND, split as r, s-r ; for OR, split SS)

∴ To build efficient secret sharing schemes, we need to build efficient AND-OR circuits for monotonic functions.

QUESTION :- f: MONOTONE ⇒ can be built using AND-OR.

What is complexity gap if we allow the use of AND NOT ?



Monotone functions

f implementable using

If MP is set of ~~those circuits~~ f such that their implementations are poly-sized AND-OR circuits, then is

$$MP \stackrel{?}{=} P \cap \text{Monotone} \; ?$$

However, the answer is

$$MP \neq P \cap \text{Monotone}$$

Example: Perfect matching of Graphs:- Monotone problem with small AND-OR-NOT circuit, but large AND-OR circuit.

Next problem:- Can we find an access structure A s.t

$A \in P \cap$ Monotone but $A \notin MP$.

This is OPEN PROBLEM.

→ COMPUTATIONAL SECRET SHARING

$\forall B \in A \; \exists$ polytime reconstruction algorithm

$\forall B \notin A$, $\forall$ PPTM D,

$$P[D(\text{share}_B) = s] \leq \text{negl}(n)$$

e.g. Consider Shamir's ~~problem~~ solution.

$|s| = \log |F|$ bits

$|s_i|$ → we use $n \cdot \log |F|$ bits [if we stored in a single place, $\log |F|$ bits].

This is also the best solution to ensure PERFECT secrecy.

Now, consider the computationally bounded version.

secret $s \longrightarrow Enc_k(s) = c$

k: small, c: small [using PRG]

Now, we dispose C using Information Dispersal Algo., ~~s=6~~ using Reed-Solomon codes, so that any $\geq t+1$ people can reconstruct ~~&~~ C.

Also, on ~~k,~~ ~~db~~ (subsets = small), do Shamirs secret sharing.

? ~~n|k|+ (t+1) log|F|~~

$\therefore n|k| + \binom{n}{t+1} \log |F|$ bits needed, and computational secret sharing ensured.

We wish to generalize this method to General Access structures.
No known poly-sized solution exists.