

Intro to Modern Cryptography

- J. KATZ + Y. LINDELL

POTS

FANTASTIC Q When is a problem Infosec one? [Ans] When the problem is impossible to solve logically / perfectly.

Will be shown with standard examples.

e.g.: Hashing password.

Theoretically impossible to be perfect if length is not infinite.

e.g.: Secure communication

$$\text{① } \text{to } \text{info}(R) = \text{info}(eve) \quad (S) \xrightarrow{\text{Enc}} (R)$$

$$\text{② } \text{to } \text{info-rec}(R) = \text{info-rec}(eve)$$

e.g.: Data integrity

If m was sent modified to m' \rightarrow It is the same to receiver.
and if $m' \neq m$, L cannot thus identify.



OO Eg of non infosec \rightarrow Infosec

• problem in distributed computing \therefore now an infosec problem.

Solution: use signatures \Rightarrow implying signatures are impossible

FASCINATING Q. How to logically solve/circumvent a logical impossibility?

[Ans] Bring in another impossibility and make it destructively interfere with the original one.

We focus on 4-5 sources of impossibilities in the semester.

Course: See impossibilities

Introduce others

Solve them

1 per month approx.

Sources of Impossibility.

① Computational Hardness [Resource Complexity]

Only happy
cats are secure.

② Practical Uncertainties

Speed of light
distance stuff

③ Natural Limits

④ Logical / Philosophical Impossibilities

Random Words

- Hamming Distance

- Information security is God
- all non-trivial works of
science must include
Info sec

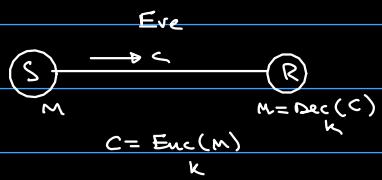
- logical norgo

Topics to cover

- Kerckhoff's Principle
- Designing/Breaking classical ciphers.

Starting off with secure communication networks.

- traditional ciphers, and how to break them.
- Shannon next class.
 - defined information
 - path breaking.

Caesar Cipher

$$c = (m + 3) \% n_c$$

m = message
 c = coded message

n_c = no. of characters in alphabet.

Big talk about his perspective of infosec as an 'art', and a bigger rant on what is art and what is science.

Exact words in book.

Kerckhoff's Principle

Security of a system must NOT depend on the OBSCURITY of the algorithm, rather must solely depend on the SECRECY of the KEY.

Kerckhoff's Reasons

1. Algorithms are reverse engineerable.

e.g.: $h(x) = bx + c$

Attacker can feed x_1, x_2, \dots, x_n and see that all outputs $h(x_1), \dots, h(x_n)$ for $g | h(x_i) - h(x_j)$. And then solve for c .

2. Updation/ Recovery Complexity.

if passwd rarely in secure system: change pass.
if algo " in obscurity " : // fixed .

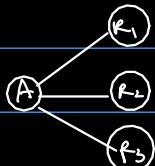
3. Secure Memory is costly.

ASK ACHIEV'A

Prob in modern times: UTM is algo and full input is key.

— bad information storage efficiency.

4. Scalable



Without: Diff algorithm for everyone.

With: only 'key' changes among people.

Additional Reasoning

1. Ethical Hacking

hypothesis: no system is secure.

bug (algo) will be found because \exists L

→ bug exists
→ nonethical people exist
→ only " " search for bug.
→ " " will find "
→ bug F, take that L

2. Standards

needed for efficiency.
if an algo is used by every system, it should obviously follow

Thus we can see why Caesar cipher fails.

Next iteration:

Shift Cipher:

$$c = (x+k) \bmod n_k$$

m = message

c = coded message

k = key

n_c = no. of characters in alphabet.

ATTACK

1. Can be broken by humans

brute force
→ If keyspace is smal, attackez.

first principle learnt!

Principle of large key space

how do we know

the key is

correct

2. Autobreaking:

- frequency analysis

$$p_i = P(i^{\text{th}} \text{ char in } m)$$

$$\text{Precompute } \sum_{i=0}^{25} p_i^2 \approx 0.065$$

$$q_i = P(" " = c)$$

Now compute

$$\sum_{i=0}^{25} (p_i q_{i+k})$$

wrong guess
correct guess

$\approx 1/26$

≈ 0.065

Next iteration:

Monoalphabetic Substitution Cipher

- Diff alphabets shift by different amounts.

- no repetitions allowed

- basically, permutation

applying first principle

for brute force: $26!$ keys to search

Attack

$$\forall i \exists j : q_j \approx p_i$$

⇒ 1. Sort q_i 's

2. Sort p_i 's

{ since distribution is same,
e.g.: $P_a = q_{t_n}$, $P_c = q_{t_b}$, $P_t = q_{t_b}$

Issue: susceptible to frequency attacks.

Next iteration

Vigenère Cipher

i. does not maintain frequency

e.g.: $\begin{array}{l} \text{h e l l o} \\ \text{s e a s e} \\ \text{z i l d} \end{array}$

FAILED ATTACKS

broke for: too many keys

• freq anal: freq. not maintained

Can be broken if:

1. key length known

2. k " is findable.

ATTACK

PART 1:

if we know length of key,

e.g.: 3

$c_0 c_3 c_6 \dots$

partition ciphertext into (length) parts

$c_1 c_4 c_7 \dots$

then shift cipher attack on all (length) parts.

$c_2 c_5 c_8 \dots$

PART 2:

Guess an k .

take a string $c_0 c_1 c_2 \dots$

$$\text{check if } \sum_{i=0}^{25} q_i^2 = \sum_{i=0}^{25} p_i^2$$

easy

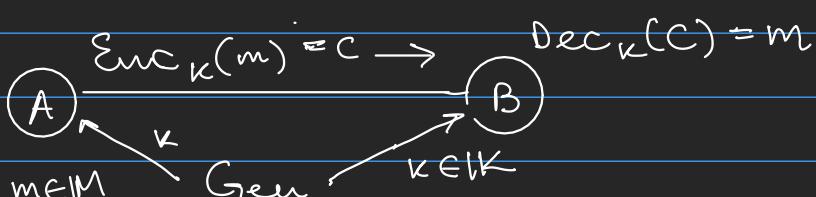
Topics for 10/01/2020

- Shannon's Perfect Secrecy
- Vernam Cipher is perfect (one-time pad)
- Limitations of Shannon's Approach

* Shannon's Perfect Secrecy

Definition of perfectly secure cipher.

An encryption scheme is a 4-tuple
 $\langle \text{Gen}, \text{Enc}, \text{Dec}, \mathcal{M} \rangle$



$$\text{Dec}_K(\text{Enc}_K(m)) = m$$

Perfectly secret
encryption
scheme.

Schemes that meet this
are largely impractical

An encryption scheme is said to be perfectly secret if for all probability distributions over \mathcal{M} , and for all $m \in \mathcal{M}$, for all $c \in \mathcal{C}$ [where $P(C = c) > 0$],

$$P[\text{Message} = m | \text{Ciphertext} = c] = P[\text{Message} = m]$$

We will prove one time pad is perfectly secret.

$$P[M = m | C = c] = P[M = m]$$

Random Variables

Vernam Cipher

$$\mathcal{M} = \mathcal{K} = \mathcal{C} = \{0, 1\}^n \quad (\text{n-bit space})$$

$$\text{Gen : } K \leftarrow_R \{0, 1\}^n$$

$$\text{Enc : } C = m \oplus k$$

$$\text{Dec : } m = c \oplus k$$

- correctness is fairly obvious and not shown here.

$$\begin{aligned} a \oplus a &= 0 \\ m \oplus a \oplus a &= m \oplus 0 = m \\ \text{Dec}_k(\text{Enc}_k(m)) &= m \oplus k \oplus k = m \\ &\quad (\text{Proved}) \end{aligned}$$

PROOF

First Showing that definition of perfect security is equivalent to

$\forall p \in \mathcal{P}$ over \mathcal{M}

$\forall m \in \mathcal{M}, \forall c \in \mathcal{C}$

$$P[C = c | M = m] = P[C = c]$$

have to do an iff.

Suppose this holds.

$$P[C = c | M = m] = P[C = c]$$

multiply both sides,

$$\frac{P[M = m]}{P[C = c]} \cdot P[C = c | M = m] = P[M = m]$$

$$\Rightarrow P[M = m | C = c] = P[M = m]$$

Second

Showing $\forall c \in \mathcal{C}, \forall m_0, m_1 \in \mathcal{M}$ is eq to

$$P[C = c | M = m_1] = P[C = c | M = m_0]$$

2 \rightarrow 1 is trivial ($P(C = c | M = m) = P(C = c), \forall m = m_0, m_1$)

$$P[C = c] = \sum_{m \in \mathcal{M}} P[C = c | M = m] P[M = m]$$

$$\downarrow P$$

$$= P \sum_{m \in \mathcal{M}} P[M = m] = P$$

Workability $\propto \frac{1}{\text{Intuitiveness}}$

The above probability tells us that the encryption of m_0 and m_1 are indistinguishable.

For Vernam Cipher:

$$\text{LHS} = P[C = c \mid M = m_0]$$

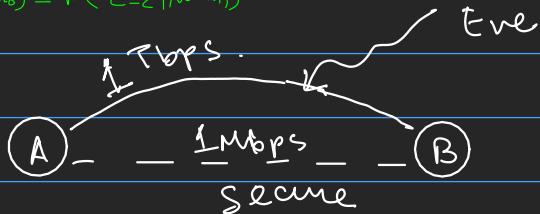
((intuition))

$$= P[C = c \oplus k] = P[k = c \oplus m_0]$$

equivalence

$$= \frac{1}{2^n} = \text{RHS} \quad (P \text{ is not dependent on } m_0)$$

Now also for m_1 ,
then by $P(C=c \mid M=m_0) = P(C=c \mid M=m_1)$



Uses of Vernam Cipher:

- i) To save your shady business from the feds, encrypt the data and delete after raid.
- ii) Use secure channel on low load days to transfer keys and on high load days, use insecure channel by encrypting.

Now, bifurcation in the field.

Symmetric key cryptography.

Two impossibilities

Slow secure channel
+ fast insecure channel

Impossible if perfect security.

Public key cryptography

Only in secure channel \Rightarrow slow secure channel

Both are impossible

Now showing that limitations of Vernam cipher apply to any perfect cipher system as per Shannon's definition.

Universality of Shannon theorem.

Thm: For any perfectly secret encryption scheme $|K| \geq |M|$

\Rightarrow no. of bits required to store message is lower bound to no. of " " " " " key.

Shannon did:

$$H(|K|) \geq H(|M|)$$

We use a handy bypass to this.

ASK ATHREYA FOR INTUITION.

Missing here:

M is compressible.

Proof:

Suppose not.

(the contrary)

$$|K| < |M|$$

some ciphertext C

we will show this directly

implies this cannot be perfectly secret

$$D = \{m \mid \exists k \in K \text{ } Dec_k(c) = m\}$$

all m 's s.t. there is a key mapping it to c

$$\text{now, } |D| \leq |K| \therefore < |M|$$

$$\Rightarrow \exists m^* \in M \text{ s.t. } m^* \notin D$$

$\log |K|$ is the max entropy of $|K|$.

and $H(K) = \log |K|$ for uniform dist $|K|$

$H(M)$ can never be larger than

$$\log(|M|)$$

if $|K| < |M|$,

$$\log |K| < \log |M|$$

not possible.

∴ we have to consider a dist where $P(M=m^*) \neq 0$

$$\therefore P[M=m^* \mid C=c] = 0$$

but we said $P(M=m^*) \neq 0$

\Rightarrow Scheme is not perfectly secret

\Rightarrow For perfectly secret scheme, $|K|$ must be at least $|M|$.

∴ one-time pad not a one-off.

17.1.20

Oh no it's Chiranjeevi

Class on either 1. Finite Fields

2. Elliptic Curve.

Groups:	(set, binary operation) satisfying axioms	eg: $(\mathbb{Z}, +)$
- closure	- Identity	
- associative	- Inverse	

for group G ,

if $H \subset G$ and satisfies property, H is a subgroup of G .

Cyclic group if $a \in G$, and $G = \{a^0, a^1, a^2, \dots\}$

eg. of $(\mathbb{Z}_n, +)$ groups

Next:

Ring, Integral Domain, Field.

Ring

$(R, +, \cdot)$ two binary operations on a set R

a) $(R, +)$ is a commutative group

b) Closure: $a, b \in R \Rightarrow a+b \in R$

c) Associative: $(a+b)+c = a+(b+c) \forall a, b, c \in R$

d) Distributive laws: $(a+b).c = a.c + b.c ; a.(b+c) = a.b + a.c$

if $a.b = b.a \forall a, b \in R$

Review

1940s

→ Shannon's Perfect Secrecy
and Pessimistic Result

$$[\text{P.S.} \Rightarrow |\mathcal{K}| \geq |\mathcal{M}|]$$

1970s - early 80s

- Two famous Relaxations

One Universal
Assumption

Class focuses on Mid 70s - early 80s.

They noted that due to pessimistic result, they had to relax some fundamental assumptions, and hopefully they would be sufficient.

Current situation: The two necessary relaxations are almost sufficient, assuming "some yet undiscovered object exists"

In Perfect Secrecy

Adversary was unbound

However, we know [adversary is computationally bounded] → First Relaxation

Zero error

with finite passwords, one must necessarily work.

[Instead of zero error, "small negligible value of err"] → Second Relaxation

turns out, these 2 are almost sufficient: Security against

- computationally bounded adversary
- negligible error

Definition of efficiency / practicality: bounded by polynomial time

↳ helps model adversary as Probabilistic Polynomial Turing Machines (PPTM)

What is a
"computationally bounded
adversary"?

Security Parameter k

Practical Adversaries: PPTM in Security Parameter

Negligible Function: A function $\mu(n)$ is said to be negligible $\xrightarrow{n \rightarrow \infty}$ if it is smallerthan the inverse of any polynomial for a sufficiently large n .

$$\left\{ \forall p() \quad \exists n_0 \text{ s.t. } \forall n \geq n_0, \mu(n) \leq \frac{1}{p(n)} \right\}$$

⇒ Secrecy (Relaxed): \nexists PPTM Adversary; $P[A \text{ can break it}] \leq \text{negl}()$

Next

define
"negligible error"

What exactly do we need to reasonable negligibility - and very few functions do not work.

Adversary trying to crack system will try multiple times. } probability grows polynomially.
 ↳ will try polynomial no. of times. }
 now, $\frac{1}{\text{fixed no.}}$ will be surpassed in time.
 We need $\frac{1}{\text{polynomial limit + 1}}$ negligibility.

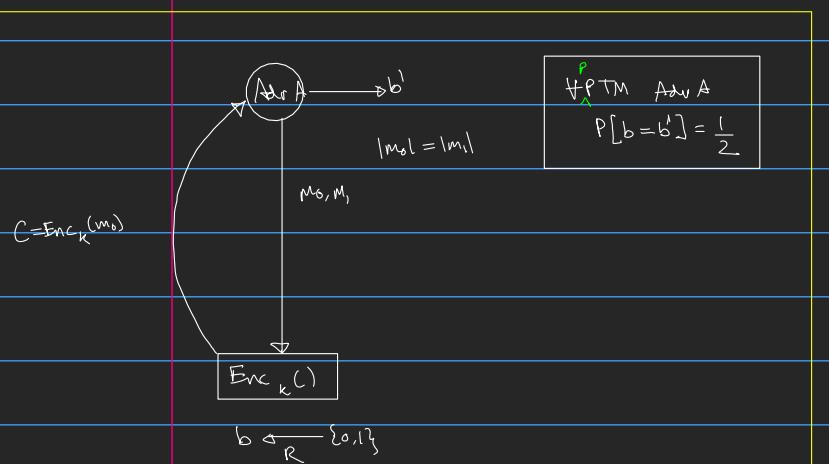
negligible means "smaller than the inverse of any polynomial".
 for example exponential growth

$\nexists p()$, $\exists n \text{ s.t. } \forall n \geq n_0$

$$\frac{1}{2^n} \leq \frac{1}{p(n)}$$

} future proofs, in the sense:

- with increased computation (polynomial bound), we keep the algorithm, increase security parameter n only.



Equivalent Definition of Perfect Secrecy.

An encryption scheme is ciphertext-only secure if $\forall \text{PTM } A, P[b = b'] \leq \frac{1}{2} + \text{negl}(n)$
 (P of selecting correct message from the message space is negligibly more than random)

other than perfect secrecy.

Bottleneck: no known encryption scheme is known to meet this definition.

Λ

Apart from these 2 relaxations, if _____ exists, it is possible.

- Pseudorandom generator
- Collision resistant hashing
- one way hash functions exist

All are the same mathematical objects. One way functions.

Does this exist? No fun due there is a faster.

Field, at this point, trifurcated.

① Heuristic Security.

(Secure because we have assumed it is)

e.g.: 1960s - mid 1990s was DES

late 1990s - current is AES

} focusing least attention to this.

because

- heuristics are prone to obsolescence
- no answer for security

③ Proven Security $\rightarrow \emptyset$ | taught when it exists

② Provable Security

We will give a conditional proof.

Definition of One-Way Function

A function $f: \{0,1\}^* \rightarrow \{0,1\}^*$ is said to be one way if : (easy to compute, hard to invert)

a) Easy to compute, $x \rightarrow f(x)$, polynomial time.

b) Hard to invert : $\nexists \text{PPTM } A$

$$P[A(f(x)) = y \mid f(x) = f(y)] \leq \text{negl}(|x|)$$

Weaker : $P[A(f(x)) = x] \leq \text{negl}(|x|)$; assumes $f^{-1}(x)$ exists

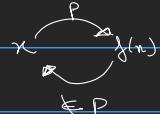
$P(\text{successfully inverting function})$ is negligible in previously defined terms.

Trio of impossibility based: Two famous relations + one way functions exist.

Not too sure
on topic heading.
Ask someone.

Now, demonstrating that proving that one way functions exist is at least NP-hard.

One Way functions:



NP, complexity theory

proving that reverse process has to be NP,

$L \in NP$ if L has an efficient verifier

decider: given $x, x \in L$?

Verifier: given proof, we can verify L .

if $x \in L$, \exists at least 1 v which works

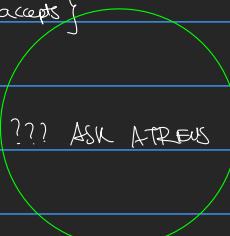
A verifier V is said to verify L if

$$L = \{x \mid \exists v, V(x, v) = \text{accepts}\}$$

Now,

if \exists p(t) NDTM, \exists efficient verifier

$\nexists \text{PPTM, not in BPP}$
 $\text{and } P \subseteq BPP \subseteq NP$
 $\Rightarrow \text{not in P}$



Reverse process has efficient verifier

as it must be NP

TODAY

- Pseudo-randomness
- An encryption scheme
- Discrete Log Problem



24.01.2020

Discrete Logarithm Problem (DLP)

Given the group \mathbb{Z}_p^* and its generator g , and $y = g^x$. Find x .

p is exponential in terms of $\log p$
bits of input.

there are no known algorithms that can solve this in polynomial time.

conjecture: this is a one-way function

Concept of one-time pads using computational entropy.



Pseudo-Random number generator

An efficient deterministic code G , that inputs n -bit string and outputs

$\lambda(n)$ -bit string is a PRG if

a) Expansion: $\lambda(n) > n$

b) Pseudo-randomness: If PPTM distinguishes D (that try to distinguish pseudo-randomness from randomness)

$$P[D(u_{\text{new}}) = 1] - P[D(G(u_n)) = 1] \leq \text{negl}(n)$$

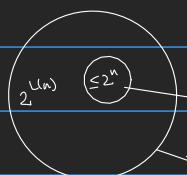
exist if one-way functions exist

G is a PRG if no efficient distinguisher can distinguish
between a uniformly distributed random of length $\lambda(n)$
and a expanded number from a smaller key of length n



Use case: send key k over slow secure channel.

Have G on both sides to get key of sufficient length to allow practical usage of one-time pad.



Even if every seed is expanded to be unique, hard cap.

world of pseudorandom seed
world of full length key

ratio is fine

to make the claim of indistinguishability is a tall ask.

e.g.: a distinguisher can ask for $2^n + 1$ samples. $P(\text{collision}) = 1$

Distinguishers definitely exist, but polynomial time unknown.

this is not polynomial time

(Idea: Any test created by any polynomial time distinguisher will be passed by both.)

Proof: $\text{PRGs} \Rightarrow \text{one way function}$.

$$G(s) = y \quad (\text{intuitive implication})$$

$$\text{for some inputs, } G(s) = y \quad s \in P$$

now given y , find s st $G(s) = y$

|| hard as at least 2^n possibilities need to be tried

Proof: One way function \Rightarrow PRG.

\Rightarrow (G is one way)
 1. forward easy
 2. reverse hard

$$p = \text{prime}$$

$$\{1, 2, \dots, p-1\} \quad g \text{ is generator of } \mathbb{Z}_p^* \text{ if } \\ \mathbb{Z}^* = \{g^0, g^1, g^2, \dots, g^{p-1}\}$$

Discrete log problem

Given the group \mathbb{Z}_p^* and generator g , $y = g^n \bmod p$, find n .

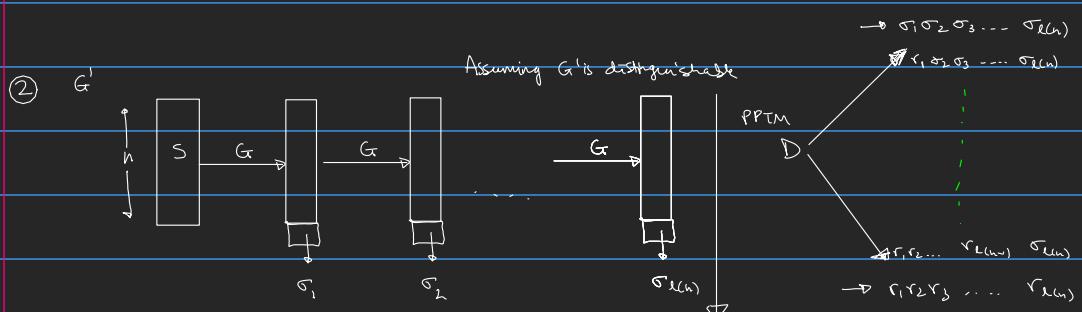
No efficient algorithm.

Can be used
as a PRG

$$\begin{array}{c|c} f: \{0,1\}^n \longrightarrow \{0,1\}^n & G: \{0,1\}^n \longrightarrow \{0,1\}^{(n+1)} \\ \downarrow \text{PRG } G & \downarrow \text{DLP } \Rightarrow G \\ \end{array}$$

① $\Rightarrow G': \{0,1\}^n \rightarrow \{0,1\}^{(n+1)}$

- Proof that
1. If I have length-preserving one-way function, I can make a PRG G that can expand its seed by 1 bit.
 2. If I have PRG G that can expand its seed by 1 bit, I can have PRG G' that can expand its seed by an arbitrary length.



if D can distinguish 1st from last,

PROOF THAT G' IS PRG

if G is PRG.

- it can distinguish that Δ is changed
- (G') was not pseudorandom
- $\Rightarrow (G')$ is not pseudorandom

$\exists i$ st i^{th} row and $(i+1)^{\text{th}}$ row is distinguished.

$\Rightarrow D$ has power to distinguish σ_i, r_i

$\Rightarrow D$ can distinguish σ_i, r_i from one of the G 's.

but None can be , contradiction.

\therefore if G is PRG, G' is also PRG.

PROOF that if one-way functions exist, single bit expanding PRGs exist.

$$\textcircled{1} \quad G(s) = f(s) \parallel h(s)$$

length-preserving
PRG
trivial enough with
one-way function

looking at first n -bits, no polynomial function should be able to predict next n -bit with $\leq \frac{1}{2} + \text{negl}(n)$ probability.

given s , easy to compute $h(s)$

$$h(s) \quad \text{given } f(s), P[A(f(s)) = h(s)] \leq \frac{1}{2} + \text{negl}(n) \quad \text{guessing this is hard}$$

$$h: \{0,1\}^n \rightarrow \{0,1\}^n$$

hard-core predicates of $f(\cdot)$

now, looking directly at DLP:

DLP_M: hardcore predicate.

$$f(x) = g^x \pmod{p}$$

$$\text{then } h(x) = \text{MSB}(x)$$

DLP: given $g^x \pmod{p}$

what is x

DLP(MSB): given $g^x \pmod{p}$
what is MSB(x)
 $[x \geq p^{-1}]$

polynomial
reducing DLP_M to DLP

Now, we will show that if there is polynomial time algorithm for DLP(MSB) will also lead to a " " DLP, which is already discounted.

\textcircled{1} DLP_L: Given $g^x \pmod{p}$, what is the LSB(x)?

DPL_L $\in P$

\textcircled{2} DLP_L + DLP_M \Rightarrow DLP.

$$\text{Let } y = g^x \pmod{p}$$

Fermat's Little Theorem implies that

$$y^{p-1} \equiv 1 \pmod{p}$$

F.L.T.
 $a^{p-1} \equiv 1 \pmod{p}$
 if $\text{GCD}(a,p) = 1$

$$\therefore (g^x)^{p-1} \pmod{p} \geq 1$$

$$\text{Find } y^{\frac{p-1}{2}} \pmod{p} = g^{x \frac{(p-1)}{2}} \pmod{p}$$

x is even

x is odd

↓

$y^{\frac{p-1}{2}} \pmod{p} \equiv -1$

we know that $g^{\frac{p-1}{2}} \pmod{p} \equiv 1$

But $g^{\frac{p-1}{2}} \pmod{p} \equiv +1 \text{ or } -1$

now we know it cannot be 1,
it has to be -1.

$$g^x \bmod p$$

↓
 x is even x is odd
 $\pm g^{\frac{x}{2}} \bmod p$ $g^{\frac{x+1}{2}} \bmod p$.

this won't tell me
 which was the original
 x unless I know
 p

(SQR) gives 2 outputs → $g^{\frac{x}{2}} \bmod p, g^{\frac{x+1}{2}} \bmod p.$

$p \equiv 3 \pmod{4}$

if $x > p^{-1}/2$
 which was our
 problem for DLP

can sqrt b.
done in poly
true?

if yes, but 2
answers

Given x find $\sum \text{sf } z^2 \pmod{p} = x$.

$$z^2 = x^{\frac{p+1}{4}} \pmod{p}$$

$$z^2 = x^{\frac{p-1}{2}} \pmod{p}$$

$$\Rightarrow x^{\frac{p-1+2}{2}} \pmod{p}$$

$$= \cancel{x^{\frac{p-1}{2}}} \cdot x \pmod{p}$$

$$\text{because } x \in \mathbb{Q}_p \quad x^2 = x \Rightarrow x^{\frac{p-1}{2}} \pmod{p} = x \pmod{p}$$

$$= 1 \pmod{p}$$

So if $p = 7$

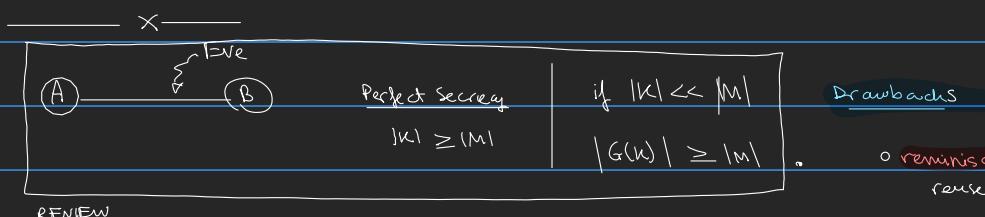
$$\sqrt{2} = 2^{\frac{7+1}{4}} = 4$$

28/1/20

Today

- o CPA-Security (prove that no deterministic encryption scheme can be CPA-secure)

o Quiz Paper



o reminiscent of one-time-pad: Cannot reuse $G(K)$

- needs state-awareness of what was passed before.

o Even if state is maintained, it is very fragile

- messages sent in parallel will break the system
- if a packet is dropped, F

o will not work at internet scale
- the internet is stateless

Pseudorandom functions (PRF)

$G_r(k)$

random Access
to $G(k)$

required for
high scalability

Assume, atm, using this, we can have good, stateless encryption schemes.

((— it is not secure to run algorithm on text.))

CPA

$C \rightarrow m$ ciphertext only attack

if we also know

$c_1 \rightarrow m_1$

$c_2 \rightarrow m_2$

:

$c_t \rightarrow m_t$

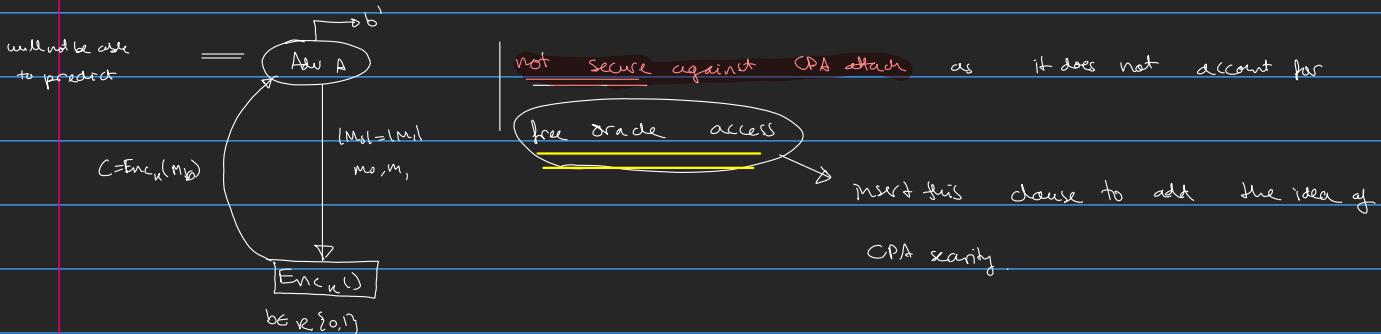
||

adversary's choice

Q Why is such an attack practical?

— due to free availability of encryption servers, adversary can generate any number of ciphertexts for corresponding messages.

RECOLLECTING DEFN



Theorem: No deterministic encryption algorithm is CPA secure.

this will not affect

Stateful algo

Ex:

if $c = c_0, b = 0$

$(m_0, m_1) \rightarrow (c_0, c_1)$

$(m_0, m_2) \rightarrow (c_0, c_2)$

elif $c \neq c_0, b = 1$

Stateless is the only viable solution, but for blocking CPA attacks we need probabilistic encryption.

Q How to probabilistic encryption and deterministic decryption?

— trick: if $\text{Enc}_k = \{0,1\}^n \rightarrow \{0,1\}^n$ is deterministic.

Choose $r \in R^{0,1}^n$

$C = \langle r, \underbrace{m \oplus \text{Enc}_k(r)}_{v} \rangle$

$\text{Dec}_k(r, v) = v \oplus \text{Enc}_k(r)$

((((DO NOT encrypt message directly)))

this is length doubling

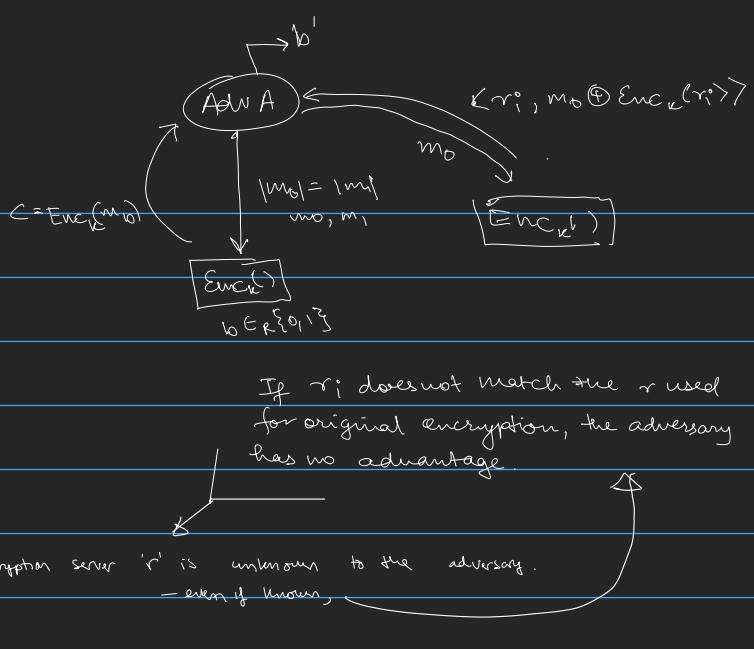
Modes of Operation

(Block Cipher)

① Cipher Block Chaining (CBC)

② output feedback mode (OFB)

③ Randomized Counter mode



OFB:

$m_1, m_2, m_3, \dots, m_t$
 $\langle r_1, c_1 \rangle, \langle r_2, c_2 \rangle, \dots, \langle r_t, c_t \rangle$

Define $r_i = \text{Enc}_k(r_{i-1})$

now, we give $\langle r_0, c_1, \langle r_1, c_2, \dots, c_t \rangle \rangle$ ALMOST LENGTH PRESERVING

converts one-wayness to a PKE.

Still in order of seconds. (t encryptions)

Randomized Counter Mode

$r_i = r_0 + i$ } is secure, will not go into prog now.

— allows for sub-of-order decryption (good for connectionless networks)

CBC

— useful in data integrity

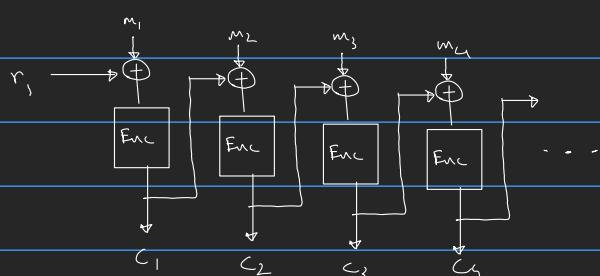
— pretty useless in encryption, compared to other 2.

if we have $m_1, m_2, m_3, \dots, m_t$

$\langle r=c_0, c_1, c_2, \dots, c_t \rangle$

$c_i = \text{Enc}_k(m_i \oplus c_{i-1})$

} has single r instead of several



Everything is enjoyable if it is unpredictable.

enjoyment or unpredictability

Quiz Qs

1. b) Show that shift cipher / vigenere cipher is perfectly secure if one alphabet/message length \leq key length

2. b) if $p-1 = s^{2^r}$, s is odd

then $(r+1)^{th}$ LSB is a hardcore predicate for DLP

Design a PRG using this

1. Shift cipher where one character is encrypted:

$$P(M=m | C=c) = P(k=k_0)$$

$$Enc_k(x) = (x+k) \% 26$$

for perfect secrecy,

$$P(M=m) = P(M=m | C=c)$$

$$P(M=m) = \frac{P(C=c | M=m) \cdot P(M=m)}{P(C=c)}$$

Now,

$$\text{form } P(C=c | M=m_0)$$

$$P(C=c) = P(C=c | M=m)$$

$$= P(k=k_0)$$

$$= \frac{1}{26}$$

(assume for M ,

$$P(C=c | M=m_0) = P(C=c | M=m_1)$$

(perfect secrecy)

$$\begin{aligned} & \text{for any } m=m_0, & & \text{if } m \in M \\ & P(C=c) = P(C=c | M=m_0) & & P(C=c) = \sum P(C=c | M=m_i) \cdot P(M=m_i) \\ & \text{otherwise for any } m=m_1, & & P(C=c) = \dots \end{aligned}$$

$$\Downarrow$$

$$P(C=c | M=m_0) = P(C=c | M=m_1)$$

2. b) if $p-1 = s^{2^r}$, s is odd

then $(r+1)^{th}$ LSB is a hardcore predicate for DLP

Design a PRG using this

- DLP = x given y, g, p , \mathbb{Z}_p^\times , $y = g^x \pmod{p}$
is one way function

$$G(f(u), h(u)) = f(u) \parallel h(u)$$

$$G(u) = u \parallel$$

$$G(u) = u+1 \Rightarrow G(u) \Rightarrow h(u)$$

Today

31.1.20

- * What is $F_k(r)$
 - Pseudo-random Functions
 - * PRFs exist iff PRGs exist
- OWF \Leftrightarrow PRG
 \Leftrightarrow PRF
 \Leftrightarrow CPT - secure
- not our concern now* \Leftrightarrow MAC \Leftrightarrow

Truly Random

$R_K(r)$

} Intuitively: First r returns random, subsequent r s give same answer for same r

$$R_K: \{0,1\}^n \rightarrow \{0,1\}^{2^n}$$
$$(2^n)^n = 2^{n^2}$$

function.

pick one at random
and choose that

Keylength required for indexing R_K ?

$n \cdot 2^n$ bits

If key is n bits



So again if you cannot distinguish pseudo random from truly random, then $F_K(r)$ can be pseudo random

A function $F_K: \{0,1\}^n \rightarrow \{0,1\}^{2^n}$ is said to be a PRF if

(a) Efficiency: \exists a efficient deterministic algorithm that computes $F_K(x)$.

(b) Pseudo-randomness: \forall PPTM D , sufficiently large n :

$$\left| P[D(F)=1] - P[D(R)=1] \right| \leq \text{negl}(n).$$

$$\text{or } \left| P[D^{F_K(\cdot)}(1^n) = 1] - P[D^{R(\cdot)}(1^n) = 1] \right| \leq \text{negl}(n).$$

You can sample only
polynomial outputs.
 $\xrightarrow{Y} F(K, Y)$
 $\xrightarrow{D} F_K(r)$

Theorem: PRFs exist iff PRGs exist

PRF \Rightarrow PRG (OFB mode of operation)

$$G(s) = F_K(s) \quad \text{or} \quad s_i = F_K(s_{i-1}) ; s_0 = s$$

$$\text{PRG} \quad G(s) = F_K(s_0) \parallel F_K(s_1) \parallel F_K(s_2) \parallel \dots$$

$\text{PRG} \Rightarrow \text{PRF}$

Suppose we have $\text{PRG} \rightarrow G$, we use it to build PRF

$$G : \{0,1\}^n \rightarrow \{0,1\}^{2n}$$

$G_0(s) = \text{left half of } G(s)$

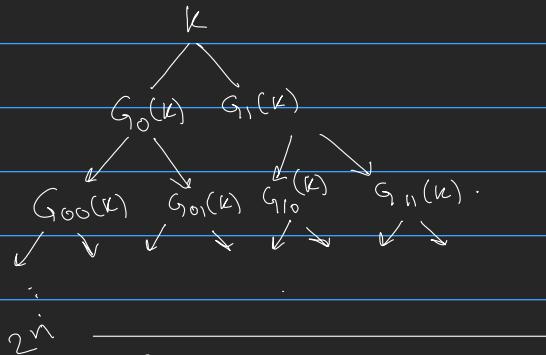
$G_1(s) = \text{right half of } G(s)$

$$G(s) = G_0(s) \parallel G_1(s)$$

$$\gamma = \gamma_0 \gamma_1 \gamma_2 \dots \gamma_{n-1}, \gamma_i \in \{0,1\}$$

$$k \in \{0,1\}^n$$

$$f_k(\gamma) = G_{\gamma_{n-1}}(\dots G_{\gamma_2}(G_{\gamma_1}(G_{\gamma_0}(k))) \dots)$$



If there is a distinguisher which distinguishes $G_{01}(k)$ from truly random, then it can distinguish $G_0(k)$ which is not possible as $G_0(k)$ is a pseudo random generator.

PRG

$$S_0 = s$$

$$S_i = f(S_{i-1}) = g^{S_{i-1}} \bmod p$$

$$G(s) = h(s_1) \parallel h(s_2) \parallel h(s_3) \parallel \dots$$

$$G(s) = \text{MSB}(g^s \bmod p) \parallel \text{MSB}(g^{s \bmod p}) \parallel \dots$$

$$= \text{MSB}(s_1) \parallel \text{MSB}(s_2) \parallel \text{MSB}(s_3) \parallel \dots$$

$$\boxed{\text{Enc}_k(m) = \langle r, f_k(r) \oplus m \rangle} \rightarrow \text{CPA secure encryption}$$

$$\text{Enc}_k(m_1 \dots m_t) = \langle r_0, f_k(r_0 + i) \oplus m \rangle$$

$$\text{AES}_k(\gamma) = c$$

$$|K| = 128 \text{ bits}$$

$$|c| = |\gamma| = 128 \text{ bits}$$

can be 256 bits

\downarrow
PRG

\downarrow
OWF

$P = NP$

Strong AI

Algorithmica

Worst-case hardness
not on average

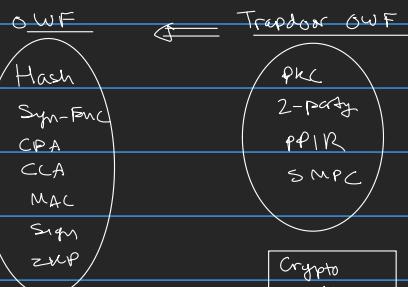
Avg-case hardness
OWF not exist

Pessiland

Minicrypt

PKCs do not exist

Syndecrypt exist



Crypto
Maria

Heuristic

04/02/2020

Review

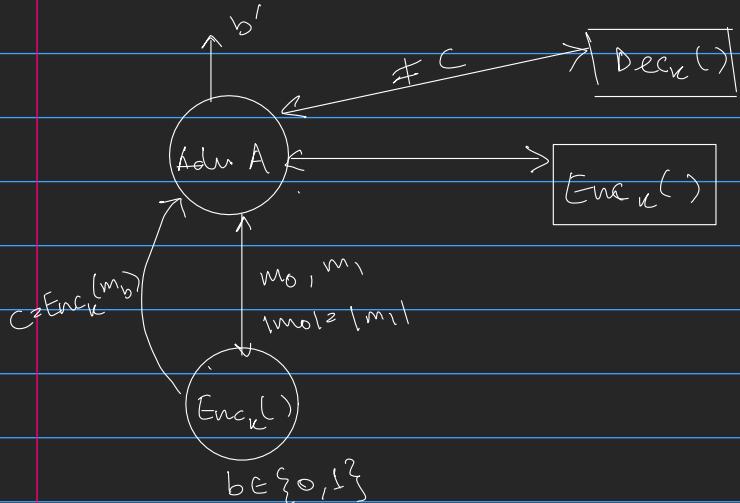
- CPA - secure Enc
- PRF

$$C = \langle r, m \oplus F_K(r) \rangle$$

Today

- # CCA - security
- # MAC

Chosen-Ciphertext Attack



CCA - sec

$\forall \text{ PPTM } A$

$$P[b' = b] \leq \text{negl}(n)$$

Consider

$$m_0 = 0^n$$

$$m_1 = 1^{\otimes n-1} 0$$

$$c = \langle r, m_0 \oplus F_K(r) \rangle$$

$$c' = \langle r, \text{Toggle MSB } (m_0 \oplus F_K(r)) \rangle$$

$$c_0 = \langle r, m_0 \oplus F_K(r) \rangle$$

$$c_1 = \langle r, m_1 \oplus F_K(r) \rangle$$

$\text{Dec}_k(c') \rightarrow$ if $b=0 \rightarrow m_1$
if $b=1 \rightarrow m_0$.

$$\text{So, } b' = \begin{cases} 0 & \text{if } \text{Dec}_k(c') = m_1 \\ 1 & \text{if } \text{Dec}_k(c') = m_0 \end{cases}$$

$$P[b = b'] = 1$$

This system of encryption is malleable (with a known change to c , the change in m can be predicted).

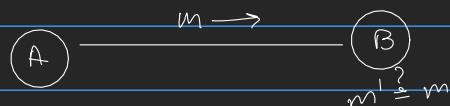
CCA - Security (To be done later)

MAC - Message authentication codes.

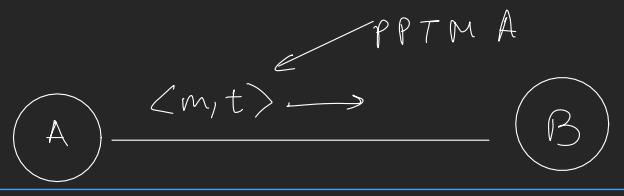
$$\boxed{\text{CPA} + \text{MAC} \Rightarrow \text{CCA}}$$

So what is MAC?

MAC (Message Authentication Codes)



Avalanche effect.
explore this



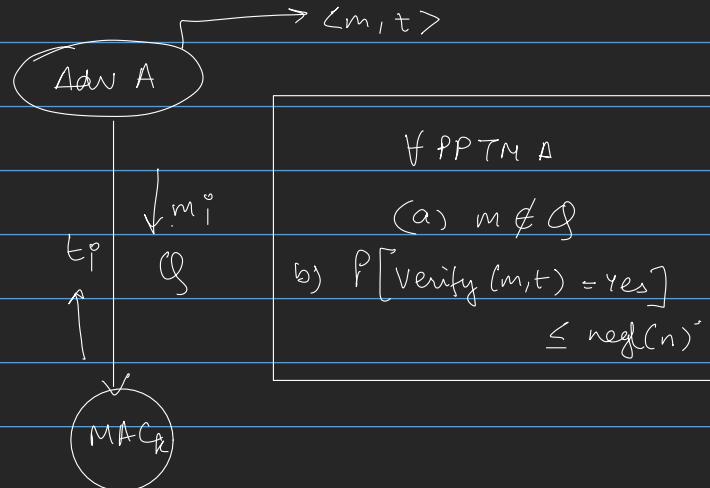
$$\text{MAC}_K(m) = t$$

$$\langle m', t' \rangle$$

$\text{Verify}_K(m', t') = \text{Yes/No}$

Security of MAC

- Solves data integrity



Encrypt - Then - Authenticate

$$\text{Ciphertext} = \langle c, t \rangle$$

$\xrightarrow{\text{CPA-Enc}_{K_1}(\cdot)}$ $\xrightarrow{\text{MAC}_{K_2}(\cdot)}$

Dec. = check for 'no-change'

$\text{Verify}_{K_2}(c, t) = \text{Yes}$

if No - 1

Yes $\text{Dec}_{K_1}(c) = m$.

Designing MACs from PRFs

If the only messages to be authenticated

$$F_K : \{0,1\}^n \rightarrow \{0,1\}^n$$

are $\{0,1\}^n$

↳ then straight forward

1 way: $t = F_K(m)$ people don't do this // usually $|m| \gg |key|$

Some approaches that failed miserably:

1. $t = F_K(\oplus m_i)$, where $m = m_1, m_2, \dots, m_n$ X FAIL

- easy to find other m' to match

2. $t = \langle t_1, t_2, \dots, t_n \rangle$ where $t_i = F_K(m_i)$ X FAIL

- can drop packets (drop $\langle m_n, t_n \rangle$) or permute

3. $t_i = F_K(i||m_i)$ X FAIL

- can drop packets at the end

4. $t_i = F_K(i||m_i)$ X FAIL

- $i = p_1, p_2, \dots, p_n \rightarrow t_1, t_2, \dots, t_n \}^{m_i} \quad p_1, p_2, p_3, p_4, \dots$
 $q_1, q_2, \dots, q_n \rightarrow s_1, s_2, \dots, s_n \}^{t_1, t_2, \dots, t_n, \dots}$

|| Exponentially many messages possible.

Send seq. no.



Send length

→ PPTM A,

$$P[A^n = \langle x,y \rangle \mid x \neq y, h^s(x) = h^s(y)] \leq \text{negl}(s)$$

Uses

- Password hash
- Message digest
- Hash chain

if DLP hard, construction for hash function is correct.

first thing noticed: no matter how hard you try, you cannot get past generic Birthday Attack.

choose an n that is at least double the security needed, as GBA can halve due to square root.

generic Birthday Attack.

• Set of N elements

• Pick q at random (with replacement)

$\Rightarrow Q \quad P(\text{picking an element twice or more})$

$$\text{Then } \frac{q(q-1)}{4N} \leq P[\text{Collision}] \leq \frac{q^2}{2N}$$

{ if $q > \sqrt{N}$... are in trouble }

$$q = \sqrt{N} \approx 2^{n/2} \quad N = 2^n$$

Prob {Union bound?} scrapped
 $P \leq \frac{\binom{q}{2}}{N}$

$$P[\text{Collision}] = P[N_{C_1}] \cdot P[N_{C_2} \mid N_{C_1}] \cdot P[N_{C_3} \mid N_{C_1, C_2}] \cdots P[N_{C_{q-1}} \mid N_{C_{q-2}}]$$

$$= \prod_{i=1}^{q-1} \left(1 - \frac{i}{N}\right)$$

$$= \prod_{i=0}^{q-1} \left(1 - \frac{i}{N}\right)$$

$$P(\text{Collision}) = 1 - \prod_{i=0}^{q-1} \left(1 - \frac{i}{N}\right)$$

$$\text{if } 0 < n < 1 \quad 1-n \leq e^{-n} \leq 1-\frac{n}{2}$$

Now,

$$\prod_{i=1}^{q-1} \left(1 - \frac{i}{N}\right) \leq \prod_{i=1}^{q-1} e^{-\frac{i}{N}}$$

$$\leq e^{-\sum_{i=1}^{q-1} \frac{i}{N}}$$

$$\leq e^{-\frac{q(q-1)}{2N}}$$

$$\leq 1 - \frac{q(q-1)}{4N}$$

$$\therefore P(\text{coll}) \geq \frac{q(q-1)}{4N}$$

Merkle-Damgard Transform

given a hash function with some compression ratio, build hash function with any compression ratio.

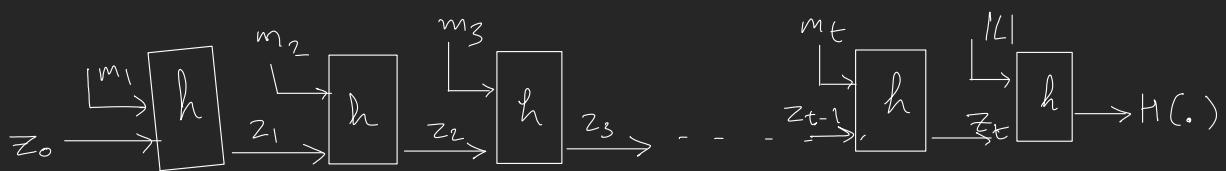
$$h : \{0,1\}^{2^n} \rightarrow \{0,1\}^n$$

If h is collision resistant, then

$$H : \{0,1\}^* \rightarrow \{0,1\}^n$$

Merkle-Damgard will give us collision resistant H .

$$H(m_1 | m_2 | m_3 | \dots | m_t)$$



$$H(m_1 \dots m_t) = H(m'_1 m'_2 \dots m'_{t'})$$

If $t \neq t'$: then $h(z_t | t) = h(z'_t | t')$

Since $t \neq t'$ then $z_t | t \neq z'_t | t'$

If $t = t'$: $h(z_t | t) = h(z'_t | t)$

If $z_t \neq z'_t$ then, $h(z_t | t) = h(z'_t | t)$
and $z_t | t \neq z'_t | t$

If $z_t = z'_t$:

If $m_t \neq m'_{t-1}$: $h(z_{t-1} | m_t) = h(z'_{t-1} | m'_t)$
and $m_t \neq m'_t$

If $m_t = m'_{t-1}$:

needs doing this till the end.

Finally we find $m_1 | m_2 | \dots | m_{i^*} | \dots | m_t = m'_1 | m'_2 | \dots | m'_{i^*} | \dots | m'_t$
which are different at i^*

Now

$$h(z_{i^*} | m_{i^*}) = h(z'_{i^*} | m'_{i^*})$$

but $z'_{i^*-1} | m'_{i^*} \neq z_{i^*-1} | m'_{i^*}$

we have
found
collision
then.

Designing $h : \{0,1\}^{2n} \rightarrow \{0,1\}^n$.

$$z = g^k \bmod p$$

Define $h(x, y) = g^x \cdot z^y \bmod p$.

g is generator of \mathbb{Z}_p^*

$$h(x_1, y_1) = h(x_2, y_2) \quad \text{but } x_1, y_1 \neq x_2, y_2$$

$$y_1 \neq y_2$$

$$g^{x_1} \cdot z^{y_1} \equiv g^{x_2} \cdot z^{y_2}$$

$$g^{x_1 - x_2} \equiv z^{y_2 - y_1}$$

$$g^{x_1 - x_2} \equiv g^{k(y_2 - y_1)}$$

$$(g^{x_1 - x_2})^{(y_2 - y_1)} \equiv (g^{k(y_2 - y_1)})^{(y_2 - y_1)}$$

$$g^{\frac{x_1 - x_2}{y_2 - y_1}} \equiv g^k \Rightarrow$$

$$k = \frac{x_1 - x_2}{y_2 - y_1}$$

but this means that
you have found
discrete log of z
So h is collision
resistant function