Intro to Modern Cryptography
– J. KATZ + Y. LINDELL

POIS

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

Will be shown with standard examples.

eg1: Hashing password.
theoretically impossible to be perfect if length is not infinite.

eg2: secure communication
@ t_0 info(R) = info(eve)
@ $t > t_0$ info_rec(R) = info_rec(eve)

eg3: Data integrity
If m was sent, modified to m' → It is the same to receiver.
and if m' " " ,
└ cannot thus identify.

Eg of non infosec → infosec

- problem in distributed computing ∴ now an Infosec problem.

Solution: use signatures. ⇒ 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.

FUNDAMENTAL
- Information security is God
- all nontrivial works of science must include info sec

Random Words
– Hamming Distance

–logical nogo

Sources of Impossibility.
① Computational Hardness [Resource Complexity]
② Practical Uncertainties (Only buggy code are secure.)
③ Natural Limits (Speed of light, Quantum Stuff)
④ Logical / Philosophical Impossibilities

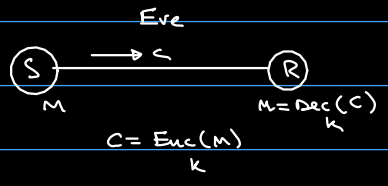
7.1.20

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

$C = (x + 3) \bmod n_c$

M = message
C = coded message

n_c = no. of characters in alphabet.

Brig 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 Reasoning

1. Algorithms are reverse engineerable.

   eg: $h(x) = 5x + c$

   Attacker can feed $x_1, x_2, \ldots x_n$ and see that for all outputs $h(x_i)\ldots$

   $8 \mid h(x_i) - h(x_j)$. And then solve for c.

2. Updation / Recovery Complexity.

   if password revealed in secure system: change pass.
   if algo " in obscurity " : !! fixed.

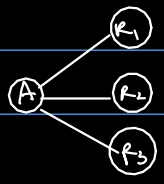
3. Secure Memory is costly.

   Proof in modern times: UTM is algo and just input is key.

   — bad information storage efficiency.

ASK ATHREYA

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

- → bug exists
- → nonethical people exist
- → only " " search for bug
- → " " " will find "
- → big 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.

→ If keyspace is smol, brute force attackez.

first principle learnt:
Principle of large key space

ATTACK

1. Can be broken by humans
2. Autocorrelating:
   - frequency analysis

how do we know the key is correct

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

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

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

Now compute $\sum_{i=0}^{25} (p_i q_{i+k})$ — wrong guess: $1/26$; correct guess: $\simeq 0.065$

Next iteration:

Monoalphabetic Substitution Cipher

- Diff alphabets shift by different amts.
- no repetitions allowed
- basically, permutation

for brute force: 26! keys to search ← applying first principle

ATTACK

$\forall i \ \exists j : q_j \simeq p_i \Rightarrow$ 1. Sort q_is, 2. Sort p_is — since distribution is same, eg: $p_a = q_n$, $p_e = q_t$, $p_t = q_b$

Issue: susceptible to frequency attacks.

Next iteration

Vigenere Cipher

1. does not maintain frequency

eg: hello
sease
zilh_

FAILED ATTACKS
- brute force: too many keys
- freq anal: freq. not maintained.

Can be broken if:
1. key length known
2. " " is findable.

ATTACK

PART 1:

if we know length of key, eg: 3 $c_0\ c_3\ c_6\ \ldots$

partition ciphertext into (length) parts $c_1\ c_4\ c_7\ \ldots$

then shift cipher attack on all (length) parts. $c_2\ c_5\ c_8\ \ldots$

PART 2:

Guess an l.

take a string $c_0 c_1 c_2 c_3 \ldots$

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

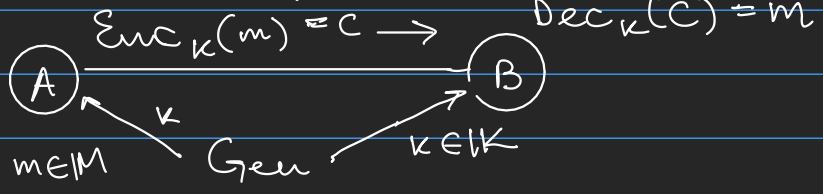
easy.

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
$\langle Gen, Enc, Dec, M \rangle$



$Dec_k(Enc_k(m)) = m$.

Perfectly secret encryption scheme.

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

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

Schemes that meet this are largely impractical

Giving definitions is easier than proving, but understanding definitions is also much.

We will prove one time pad is perfectly secret.

$$P[M=m \mid C=c] = P[M=m]$$

Random Variables.

Vernam Cipher

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

Gen : $k \xleftarrow{R} \{0,1\}^n$

Enc : $c = m \oplus k$

Dec : $m = c \oplus k$.

- correctness is fairly obvious and not shown here.

$a \oplus a = 0$

$m \oplus a \oplus a = m \oplus 0 = m$

$\text{Dec}_k(\text{Enc}_k(m))$

$= m \oplus k \oplus k = m$

(Proved)

PROOF

First showing that definition of perfect security is equivalent to

$\forall$ p dist over $\mathbb{M}$

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

$$P[C=c \mid M=m] = P[C=c]$$

have to do an iff.

1. Reduce Shannon to

2.

Suppose this holds.

$$P[C=c \mid M=m] = P[C=c]$$

multiply both sides.

$$\left\{\frac{P[M=m]}{P[C=c]}\right\} P[C=c \mid M=m] = P[M=m]$$

Bayes'

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

Second

showing $\forall c = \mathbb{C}, \forall m_0, m_1 \in M^2$ is eq to (1)

$$P[C=c \mid M=m_1] = P[C=c \mid M=m_0]$$

2 → 1 is trivial

(if $\forall m_1, m_0$, then $\forall$ p dist over M)

$(P(C=c \mid M=m) = P(C=c)$, LHS=RHS)

$$P[C=c] = \sum_{m \in M} P[C=c \mid M=m]\, P[M=m]$$

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

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:

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

equivalence

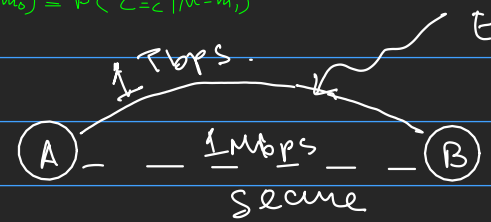
((intuition))

$$= P[C = m_0 \oplus K] = P[K = C \oplus m_0]$$

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

now also for m_1

then by $P(C=c \mid M=m_0) = P(C=c \mid M=m_1)$

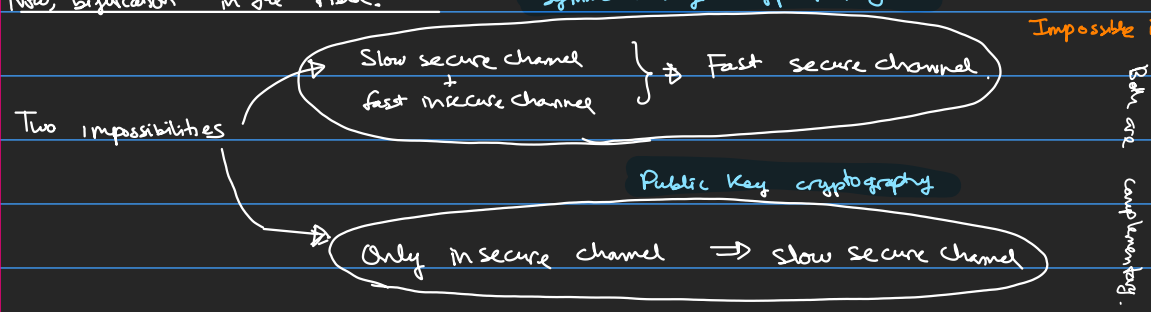


Uses of Vernam Cipher:

i) To save your shady business from the feds, encrypt the data and decrypt 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.



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

Mini version of Shannon theorem.

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

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

Missing here: M is compressible.

Shannon did: $H(|K|) \geq H(|M|)$

We use a hacky bypass to this.

ASK ATHREYA FOR INTUITION.

Proof:

Suppose not. (the contrary)

$$|K| < |M|$$

we will show this directly ↗ implies this cannot be perfectly secret.

some ciphertext c

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

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

now, $|D| \leq |K| \quad \therefore < |M|$

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

∴ 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 $|M|$.

- $\log |K|$ 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.

∴ 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
- associative
- identity
- inverse

for group G,

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

Cyclic group if $a \in G$, and $G = \{a^0, a^1, a^2, \ldots\}$ eg. of $(\mathbb{Z}_n, \cdot_n)$ groups

Next: Ring, Integral Domain, Field.

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

a) $(R, +)$ is a commutative group (if $a.b = b.a \ \forall a, b \in R$)

b) Closure: $a.b \in R \ \forall \ a, b \in R$

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

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

Ring need not be commutative with $\cdot$

Commutative Ring is also " " $\cdot$

eg. of Ring: $(\mathbb{Z}, +, \cdot)$ is a ring.

$(\mathbb{Q}, +, \cdot)$ " " "

Zero Divisors: For a ring R, a, b s.t $a(\neq 0) \in R$, $\exists\ b \in R$, $b \neq 0$, $ab = 0$ or $ba = 0$

eg: $(\mathbb{Z}_n, +_n, \cdot_n)$

Integral Domain: A commutative ring with no zero divisors.

Division Ring: An integral domain s.t $(R - \{0\}, \cdot)$ is a group.

Field: $(F, +, \cdot)$

Finite field: Field where set F is finite.

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

b) $(F - \{0\}, \cdot)$ " " "

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

Not a field: $(\mathbb{Z}, +, \cdot)$, $(\mathbb{Z}_n, +_n, \cdot_n)$ generally

field: $(\mathbb{R}, +, \cdot)$, $(\mathbb{Q}, +, \cdot)$, $(\mathbb{Z}_n, +_n, \cdot_n)$ if $n = p$, $(\mathbb{Z}_n^*, +_n, \cdot_n)$

Characteristic of Integral Domain (I)

least positive integer m s.t $\underbrace{m \cdot a}_{\text{m times}} = 0 \quad \forall a \in I$

If such m does not exist, characteristic of I is 0.

Lemma: If $(F, +, \cdot)$ is finite, $\text{char } F = p$ (for some prime)

$\Rightarrow \exists$ atleast $0, 1$. now $1+1,\ 1+1+1,\ 1+1+1+1 \ldots$

$\exists\ n$ $\cdot$ sums have to be repeated, as finite.

$i.1 = j.1$

$(i-j).1 = 0$

$n = i - j$

n is prime $\cdot$ if n is not prime, $n = ab$.

how do we know $a, b \in F$

$(a.b).1 = 0$

$\Rightarrow (a.1)(b.1) = 0$

$\Rightarrow a.b = 0$ $(a, b$, non-zero$)$

$\Rightarrow$ one of a, b zero divisor

contradictory to definition

$\Rightarrow n$ is prime.

Now, $(\mathbb{Z}_p, +_p, \cdot_p)$ is a field char $= p$

if for any given F, $\text{char } F = p$, $\mathbb{Z}_p (= \mathbb{Z}_{/pZ}) \subset F$

- if $(F, +, \cdot)$ is finite, $\text{char } F = p$ (for some prime p)

Let $(F, +, \cdot)$ be a finite field $|F| = q$, let $F \subset K$ where K is also a finite field

then K has q^n elements where n is dim of K over F

$\{v_1, v_2, \ldots v_n\}$ is a basis of K over F

$a \in K$, $a = a_1 v_1 + a_2 v_2 + \ldots a_n v_n$, $a_i \in F$

$|K| = q^n$

$(K, +, \cdot)$, $\text{char } K = p$

eg: $(\mathbb{C}, +, \cdot)$ is a vector space over $(\mathbb{R}, +, \cdot)$ $\quad |K| = p^n$

given a prime, field with exactly those many elements.

Read Herstein

Elliptic Curves

Can be defined over finite fields
- gives an abelian group.
- multiple curves can be taken.

Should not have multiple roots.

— $y^2 = x^3 + ax + b$ defined over F

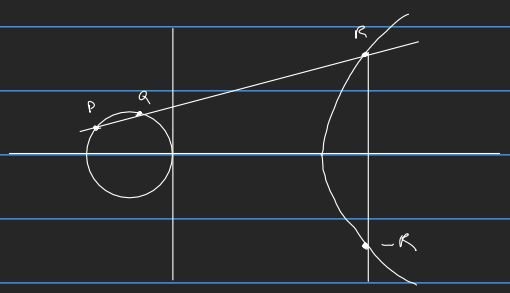
$P, a \qquad l = \overline{PQ}$

l intersects this curve exactly at one more point

$P + Q = -R$

$P + O = O + P$ — O is point at infinity

$R - R = O$



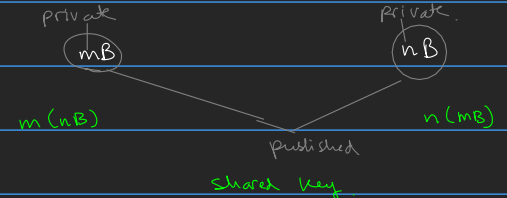
How this helps in crypto:

2 parties agree on:

1. Field F
2. Eq. of elliptic curve
3. Random B

Alice — Bob

F, a, b, B



Shared key

21. 1. 20.

Review

1940s
⇒ Shannon's Perfect Secrecy and Pessimistic Result
[P.S ⇒ |K| ≥ |M|]

1970s - early 80s
• Two famous Relaxations | One Universal Assumption.

Class lectures 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 error"] → 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

Used to define "negligible error"

Practical Adversaries: PPTM in Security Parameter

Negligible Function: A function $\mu(n)$ is said to be negligible in n, if it is smaller than the inverse of any polynomial for a sufficiently large n.

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

∴ Secrecy (Relaxed): ∀PPTM Adversary; P [A can break π] ≤ negl()

What exactly do we need for reasonable negligibility — and why fixed fractions 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 $\frac{1}{\text{polynomial (inverse +1)}}$ negligibility.

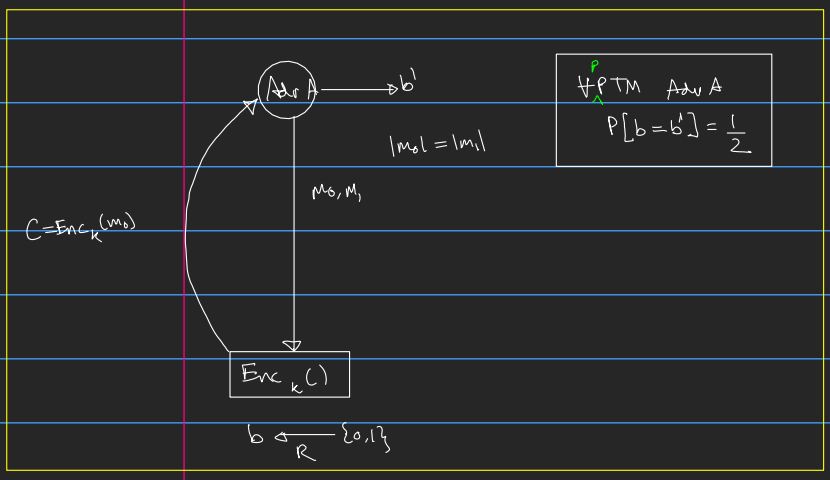
negligible means "smaller than the inverse of any polynomial".

for examples exponential growth.

$\forall p(), \exists n_0$ 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.



$\forall$ PPTM Adv A
$P[b = b'] = \frac{1}{2}$

Equivalent Definition of Perfect Secrecy.

An encryption scheme is ciphertext-only secure if
$\forall$ PTM A, $P(b = b') \leq \frac{1}{2} + negl(n)$
(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 need this definition.

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 object: One way functions.

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

Fields, at this point, trifurcated.

① Heuristic Security.

(Secure because we have assumed it is)

eg: 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 → ϕ | taught when it exists

② Provable Security

we will give a conditional proof.

Definition of One-Way Function

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

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

b) Hard to invert : $\forall$ PPTM A

$$P\left[A\left(f(x)\right) = y \mid f(x) = f(y)\right] \leq negl(|x|)$$

P(successfully inverting function) is negligible in previously defined terms.

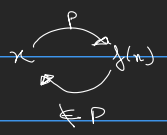
weaker : $P[A(f(x)) = x] \leq negl(|x|)$; assumes $f^{-1}(x)$ exists

Trio of impossibility breakers: Two famous relaxations + 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,

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

Verifier: given proof, we can verify it.

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

if $x \in L$, $\exists$ at least 1 u which works. A verifier V is said to verify L if

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

$\forall$ PPTM, not in BPP

and $P \subseteq BPP \subseteq NP$

$\Rightarrow$ not in P

Now,

if $\exists$ p(t) NDTM, $\exists$ efficient verifier ??? ASK ATREUS

Reverse process has efficient verifier

∴ it must be NP

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

24.01.2020.

Discrete Logarithm Problem (DLP)

Given the group Z_p^* and its generator g, and $y = g^x$ Find x.

p is expressed i.t.o. $\log p$ 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 $l(n)$-bit string is a PRG if

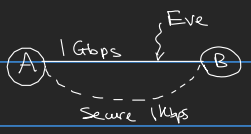
a) Expansion: $l(n) > n$

b) Pseudo-randomness: $\forall$ PPTM distinguishers D (that try to distinguish pseudo-randomness from randomness)

$$P\left[D(U_{l(n)}) = 1\right] - P\left[D(G(U_n)) = 1\right] \leq negl(n)$$

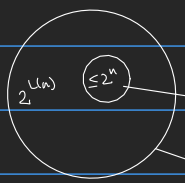
exist if one-way functions exist.

G is a PRG if no efficient distinguishers can distinguish between a uniformly distributed random no of length $l(n)$ and an 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 tiny

o/o to make the claim of 'indistinguishability' is a tall ask.

eg: a distinguisher can ask for $2^n + 1$ samples. P(collision) = 1

this is not polynomial time.

distinguishers

Distinguishers definitely exist, but polynomial time unknown.

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

Proof: PRGs ⇒ one way function

$G(s) = y$ (intuitive implication)

for some inputs, $G(s) = y$ } → P

now given y, find x s.t $G(x) = G(y)$

|| hard as at least $2^n + 1$ possibilities need to be tried

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

Proof: One way function ⇒ PRG.

p = prime

$\{1, 2, \ldots, p-1\}$ g is generator of Z_p^* if

== $\{g^0, g^1, g^2, \ldots g^{p-1}\}$

Discrete log problem

Given the group Z_p^* and generator g, $y = g^x \bmod p$, find x.

No efficient algorithm.

Can be used as a PRG

$f: \{0,1\}^n \longrightarrow \{0,1\}^n$

PRG G

① n-bit seed as input, (n+1) bit seed as output

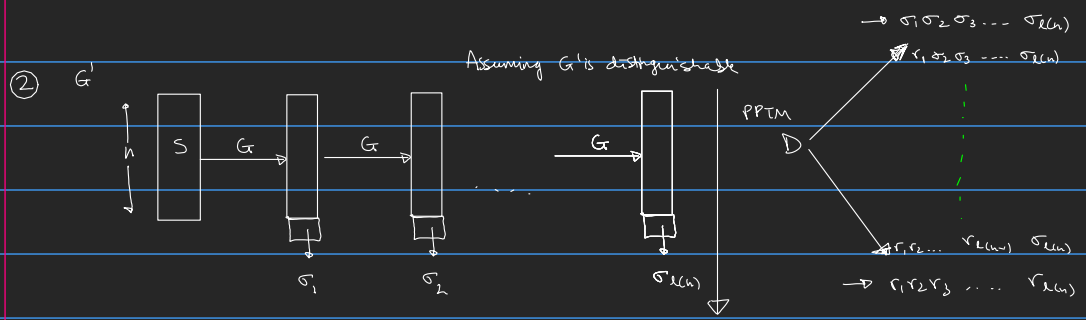
$G: \{0,1\}^n \longrightarrow \{0,1\}^{n+1}$

⇒ $G': \{0,1\}^n \longrightarrow \{0,1\}^{l(n)}$

DLP ⇒ G

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.

② G'



if D can distinguish 1st from last,

∃ i s.t ith row and $(i+1)^{th}$ row is distinguished.

⇒ D has power to distinguish some σ_i, r_i

⇒ D " " " " one of the Gs.

but None can be, contradiction.

∴ if G is PRG, G' is also PRGs.

PROOF THAT G' is PRG

If G is PRG.

- it can distinguish that i'th bit changed
- ⇒ (G_i) was not pseudorandom (contradiction)

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

① $G(s) = f(s) \,||\, h(s)$

- $f(s)$: length-preserving PRG; trivial enough with one-way function.
- $h(s)$: looking at first n-bits, no polynomial function should be able to predict $n+1^{th}$ bit with $\leq \frac{1}{2} + negl()$ probability.

$h(s)$:
- given s, easy to compute $h(s)$
- given $f(s)$, $P[A(f(s)) = h(s)] \leq \frac{1}{2} + negl()$ } guessing this is hard

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

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

now, looking already at DLP:

$f(x) = g^x \bmod p$

then $h(x) = MSB(x)$

DLP: given $g^x \bmod p$, what is x

DLP(MSB): given $g^x \bmod p$, what is $MSB(x)$ $\left[x \geq \frac{p-1}{2}\right]$

DLPM: hardcore predicate.

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

① DLPL: Given $g^x \bmod p$, what is the LSB(x)?

DLPL $\in$ P

② DLPL + DLPM $\Rightarrow$ DLP.

Let $y = g^x \bmod p$

Fermat's Little Theorem implies that (F.L.T)

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

$a^{p-1} \equiv 1 \pmod p$ if $GCD(a,p) = 1$

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

Find $y^{\frac{p-1}{2}} \pmod p = g^{x\left(\frac{p-1}{2}\right)} \bmod p$.

$-1^n = 1 / -1$

- x is even: $\to 1$
- x is odd: $\to g^{\frac{p-1}{2}} \bmod p \equiv -1$

we know that $g^{p-1} \bmod p \equiv 1$

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

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

$g^x \bmod p$

x is even → $\pm g^{x/2} \bmod p$ (sqrt) ↳ gives 2 outputs → $g^{x/2} \bmod p$, $g^{\frac{x}{2}+\frac{p-1}{2}} \bmod p$.

x is odd → $g^{x-1} \bmod p$.

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

If $x > p^{-1}/2$

which was our problem for DLPM

$p \equiv 3 \pmod 4$

Can sqrt be done in poly time?

Ie: gets back 2 answers

Given x find z s.t. $z^2 \pmod p = x$.

$z = x^{\frac{p+1}{4}} \bmod p$.

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

$= x^{\frac{p-1+2}{2}} \bmod p$

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

because $x \in QR$ $\alpha^2 = x$ $\Rightarrow x^{\frac{p-1}{2}} \pmod p = \alpha^{p-1} \pmod p = 1 \pmod p$

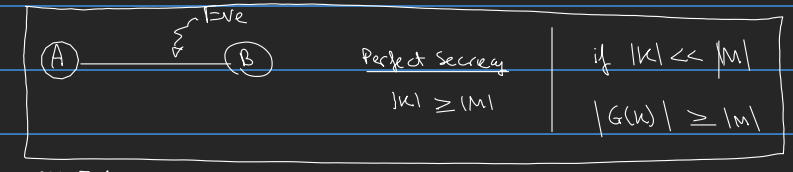
So if $p = 7$

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

28.1.26

Today

- CPA-Security (prove that no deterministic encryption scheme can be CPA-secure) [Chosen Plaintext]
- Quiz Paper



REVIEW

Drawbacks

- reminiscent of one-time-pad: Cannot reuse $G(k)$
  - needs state-awareness of what was passed before.
- Even if state is maintained, it is very fragile
  - messages sent in parallel will break the system.
  - if a packet is dropped, F
- 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

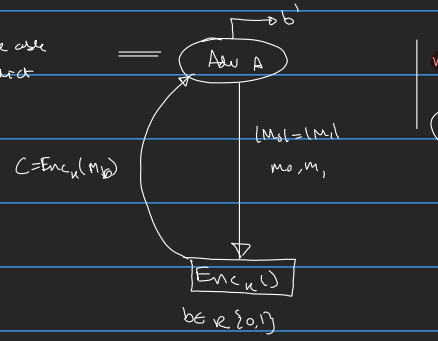
$C \longrightarrow m$ ciphertext only attack

if we also know $c_1 \longrightarrow m_1$, $c_2 \longrightarrow m_2$, ⋮, $c_t \longrightarrow m_t$ ‖ 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



not secure against CPA attack as it does not account for free oracle access

insert this clause to add the idea of CPA security.

Theorem: No deterministic encryption algorithm is CPA secure.

this will not affect ← Stateful algo

Expt:

if $c = c_0$, $b = 0$

elif $c \neq c_0$, $b = 1$

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

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

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 $Enc_K = \{0,1\}^n \longrightarrow \{0,1\}^n$ is deterministic.

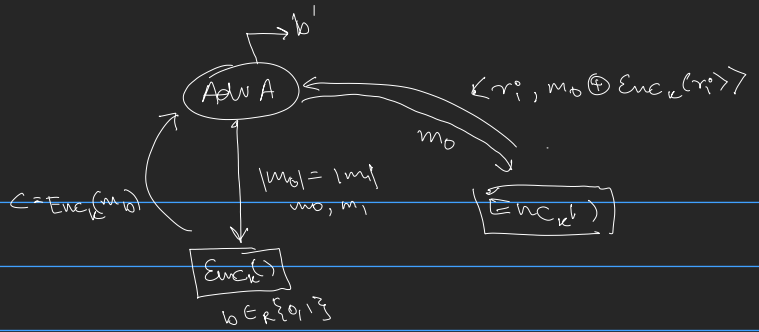
Choose $r \in_R \{0,1\}^n$

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

$Dec_K(r, v) = v \oplus Enc_K(r)$

(((DO NOT encrypt message directly)))

this is length doubling



If r_i does not match the r used for original encryption, the adversary has no advantage.

Encryption server 'r' is unknown to the adversary. — even if known,

Modes of Operation

(Block Cipher)

1. Cipher Block Chaining (CBC)
2. output feedback mode (OFB)
3. Randomized Counter mode

OFB:

$m_1, m_2, m_3, \ldots m_t$

$\langle r_1, c_1 \rangle, \langle r_2, c_2 \rangle, \ldots \langle r_t, c_t \rangle$

Define $r_i = Enc_k(r_{i-1})$

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

converts one-wayness to a PRG.

Still in order of seconds. (t encryptions)

Randomized Counter Mode

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

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

CBC

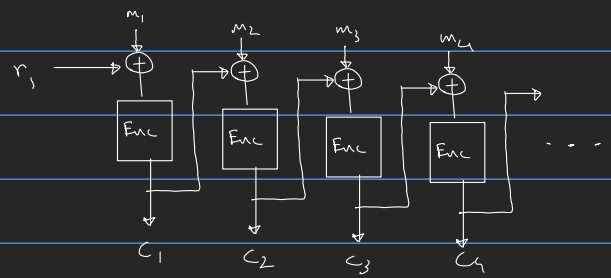
— useful in data integrity

— pretty useless in encryption, compared to other 2.

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

$\langle r = c_0, c_1, c_2, \ldots c_t \rangle$ } has single r instead of several

$c_i = Enc_K(m_i \oplus c_{i-1})$



Everything is enjoyable if it is unpredictable. enjoyment $\propto$ unpredictability

Quiz Qs

1. b) Show that shift cipher / vigenere cipher is perfectly secure if one alphabet/message length ≤ 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: $P(M=m \mid C=c) = P(K=k_0)$

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

for perfect secrecy, $P(M=m) = P(M=m \mid C=c)$

$P(M=m) = \dfrac{P(C=c \mid M=m) \cdot P(M=m)}{P(C=c)}$

Now,

for m $P(C=c \mid M=m_0)$

$= P(K=k_0)$

$= \frac{1}{26}$

Likewise for m_1

$\therefore P(C=c \mid M=m_0) = P(C=c \mid M=m_1)$

(Perfect secrecy)

$P(C=c) = P(C=c \mid M=m)$

for any $m=m_0$

$P(C=c) = P(C=c \mid M=m_1)$

Likewise for any $m=m_1$

$\forall\ m \in M$

$P(C=c) = \sum P(C=c \mid M=m) \cdot P(M=m)$

$P(C=c) \cdot 1$

$P(C=c \mid M=m_0) = P(C=c \mid M=m_1)$

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, Z_p^*, $y = g^x \bmod p$

  is one way function

- $G(f(x), h(x)) = f(x) \mid h(x)$

  $G(x) = x+1$

- $G(x)=x+1$, $\Rightarrow G(x) \Rightarrow l(x)$

Today

31.1.20

* What is $F_k(r)$
  — Pseudo-random Functions
* PRFs exist iff PRGs exist

OWF $\Leftrightarrow$ PRG
$\Leftrightarrow$ PRF
$\Leftrightarrow$ CPA-secure
$\Leftrightarrow$ MAC $\Leftrightarrow$ ...

(more concerning now)

$\Leftrightarrow$

Truly Random
$R_k(r)$

} Intuitively: first r returns random, subsequent rs give same answer for same r

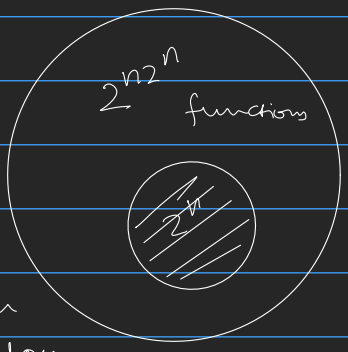
$R_k : \{0,1\}^n \rightarrow \{0,1\}^n$

$(2^n)^{2^n} = 2^{n2^n}$ 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 $F_k : \{0,1\}^n \rightarrow \{0,1\}^n$ is said to be a PRF if

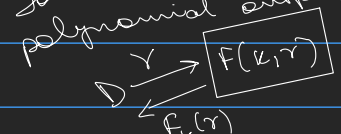
(a) Efficiency: $\exists$ a efficient deterministic algorithm that computes $F_k(x)$.

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

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

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.



Theorem: PRFs exist iff PRGs exist

PRF $\Rightarrow$ PRG (OFB mode of operation)

$G_1(s) = F_k(s)$ or $s_i = F_k(s_{i-1})$; $s_0 = s$

PRG $G(s) = F_k(s_0) \,||\, F_k(s_1) \,||\, F_k(s_2) \,||\, \ldots$

PRG ⇒ PRF

Suppose we have PRG $\to G$, we use it to build PRF

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

$G_0(s)$ = Left half of $G(s)$

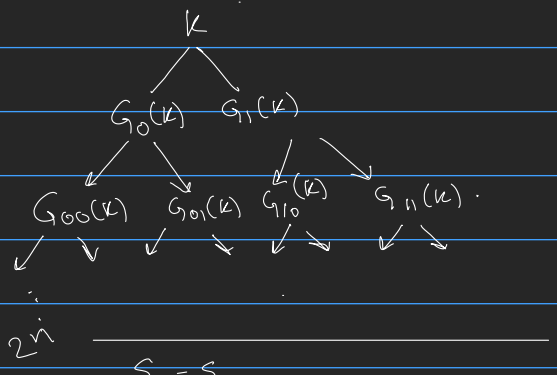
$G_1(s)$ = Right half of $G(s)$

$G(s) = G_0(s) \,||\, G_1(s)$

$r = r_0 r_1 r_2 \ldots r_{n-1}$, $r_i \in \{0,1\}$

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

$$F_k(r) = G_{r_{n-1}}(\ldots G_{r_2}(G_{r_1}(G_{r_0}(k)))\ldots)$$



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.

$s_0 = s$

PRG $\quad s_i = f(s_{i-1}) = g^{s_{i-1}} \bmod p$

$$G_1(s) = h(s_1) \,||\, h(s_2) \,||\, h(s_3) \,||\, \ldots$$

$$G(s) = MSB(g^s \bmod p) \,||\, MSB(g^{g^s} \bmod p) \,||\, \ldots$$

$$= MSB(s_1) \,||\, MSB(s_2) \,||\, MSB(s_3) \,||\, \ldots$$

$\boxed{Enc_k(m) = \langle r, F_k(r) \oplus m \rangle}$ → CPA secure encryption

$Enc_k(m_1 \cdots m_t) = \langle r_0, F_k(r_0 + i) \oplus m \rangle$

$AES_k(r) = c$ $\quad |K| = 128$ bits $\quad |C| = |r| = 128$ bits

can be 256 bits

PRG

OWF

OWF ⟸ Trapdoor OWF

OWF: Hash, Sym-Enc (CPA, CCA), MAC, Sign, ZKP

Trapdoor OWF: PKC, 2-party, PPIR, SMPC

P=NP

Strong AI

Algorithmica

Worst-case hardness

not on average

Heuristica

Avg-case Hardness

OWF not exist

Pessiland

Mini Crypt

PKCs do not exist

Symkeys exist

Crypto Mania

04/02/2020

Review

→ CPA - secure Enc

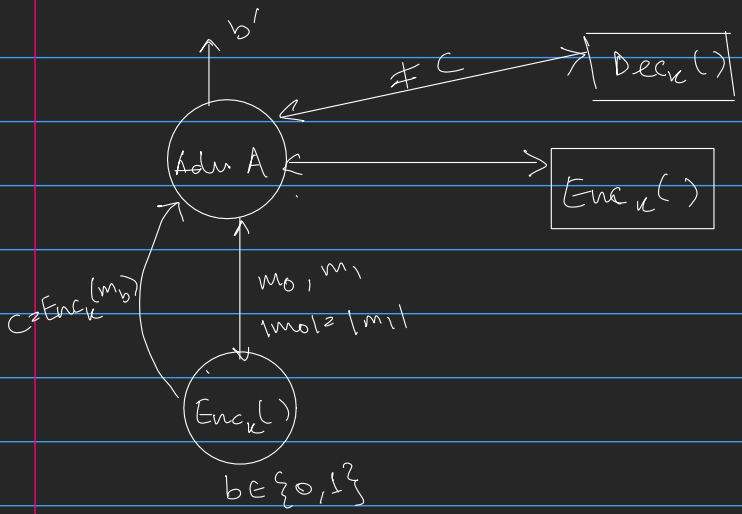
→ PRF

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

Today

* CCA - security
* MAC

Chosen - Ciphertext Attack — CCA - sec



$\forall$ PPTM A

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

Consider $m_0 = 0^n$, $m_1 = 10^{n-1}$

$c = \langle r, m_b \oplus f_K(r) \rangle$

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

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

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

$Dec_K(c') \rightarrow$ if $b = 0 \rightarrow m_1$

if $b = 1 \rightarrow m_0$.

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

$P[b = b'] = 1$

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

Avalanche effect: explore this

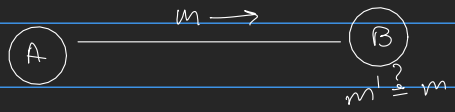
CCA - Security (To be done later)

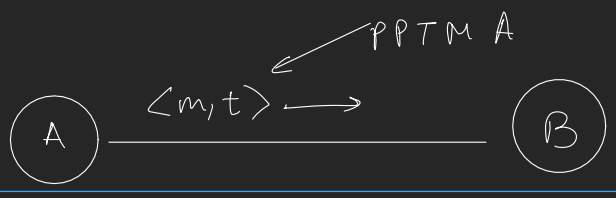
MAC — Message authentication codes.

CPA + MAC ⇒ CCA

So what is MAC?

MAC (Message Authentication Codes)

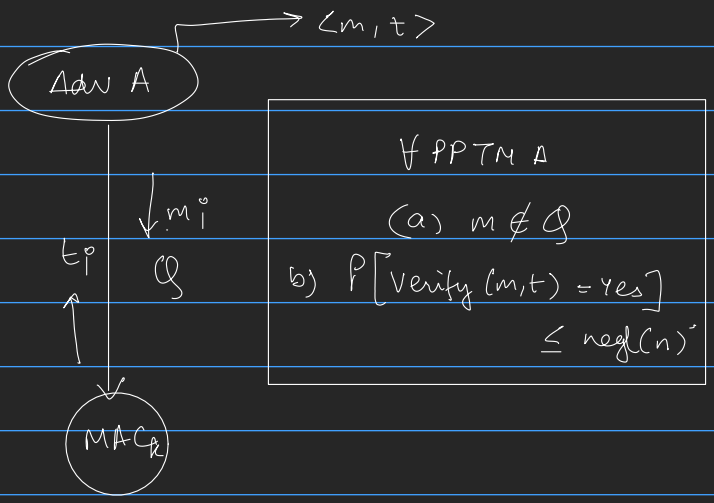




$MAC_K(m) = t$ $\quad$ $\langle m', t' \rangle$

$Verify_K(m', t') = Yes/No$

Security of MAC



- Solves data integrity

Encrypt – Then – Authenticate

CPA-$Enc_{K_1}()$ $\quad$ $MAC_{K_2}(C)$

Ciphertext = $\langle C, t \rangle$

Dec. = Check for 'No-change'

$Verify_{K_2}(C, t) = Yes$

if No – $\perp$

Yes $Dec_{K_1}(C) = m$.

Designing MACs from PRFs

If the only messages to be authenticated are $\{0,1\}^n$ $\quad$ $F_K : \{0,1\}^n \to \{0,1\}^n$

↳ then straightforward.

1 way: $\boxed{t = F_K(m)}$ $\quad$ people don't do this // usually $|m| >> |key|$

Some approaches that failed miserably:

1. $t = F_K(\oplus m_i)$, where $m = m_1 m_2 \ldots m_\ell$ $\quad$ ✗ FAIL

   – easy to find other m' to match.

2. $t = \langle t_1, t_2, \ldots, t_\ell \rangle$ where $t_i = F_K(m_i)$ $\quad$ ✗ FAIL

   Sends seq. no.

   – can drop packets (drop $\langle m_k, t_k \rangle$) or permute

3. $t_i = F_K(i \| m_i)$ $\quad$ ✗ FAIL

   Sends length

   – can drop packets at the end

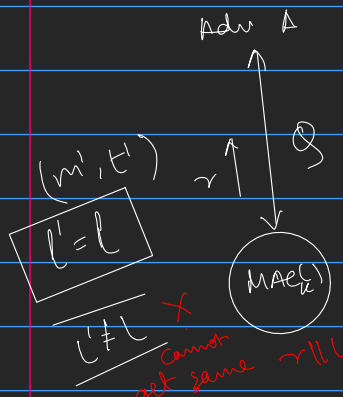
4. $t_i = F_K(\ell \| i \| m_i)$ $\quad$ ✗ FAIL

   – $p = p_1 p_2 \ldots p_\ell \to t_1 t_2 \ldots t_\ell$
   $q = q_1 q_2 \ldots q_\ell \to s_1 s_2 \ldots s_\ell$
   $\Big\}$ m'': $p_1 q_2 p_3 q_4 \ldots$, $t_1 s_2 t_3 s_4 \ldots$

   Exponentially new messages possible.

message id

$t_i = F_k(r \| L \| i \| m_i)$ ✓ WORKS



$m = m_1 m_2 \cdots m_l, \langle t_1, t_2, \ldots, t_l \rangle$

$t_i = F_k(r \| l \| i \| m_i)$

$m = m_1 m_2 \cdots m_l$

$m' = m_1' m_2' \cdots m_l'$

$m \neq m'$

$\exists i \; m_i = m_i'$

But this is NOT practical as the length of tag is as long as the message itself.

CBC MAC_K → CipherBlockChaining.

t_0

$m_1, m_2, \ldots, m_q$

$t_i = F_k(t_{i-1} \oplus m_i)$

$t = t_q$

① $t = F_{k_2}(t_q)$ $t_q = CBCMAC_{k_1}(m)$ ✓ used for stream of data where l is not known beforehand

② $k' = F_k(L)$ $t = t_q$ $CBCMAC_{k'}(m)$

③ Prepend the length ✓ In practice

$m' = l \| m_1 \| \cdots \| m_q$

$t = CBCMAC(m')$.