


Textbook: Intro to Modern Cryptography
- J. Katz, Y. Lindell.

Fantastic -

Q: When is a problem an InfoSec (IS) one?

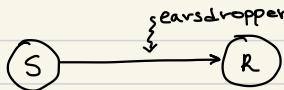
A: When the problem is impossible to solve perfectly.

If this is indeed true, then naturally the field is fantastic, since we are dealing with problems that are themselves impossible.

Consider the simple problem of a password scheme.

If you want to make a "perfectly secure" password scheme, you require infinite length passwords. \therefore perfectly secure password schemes cannot exist. \Rightarrow It is an infosec problem.

Consider the secure communication problem.

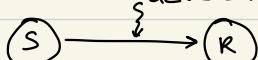


There must be some time t_0 , such that the eavesdropper knows everything that the receiver knows.

Now for each time $t > t_0$, the eavesdropper receives everything that the receiver receives.

Notice this is again an IS problem. Due to the inherent impossibility.

Consider the Data Integrity Problem -

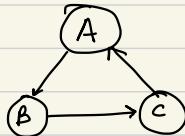


Is the data sent by the sender same as the data received by the receiver?

Assume to the contrary that R knows when a modification is made. If the R can detect this, it can magically distinguish b/w an un-

-modified M' and some m modified into M' .

consider a problem where there is one liar.



How do I find out whose the liar? (who changed the message).

B sends a 0 and claims it sent a 1 OR C claims it received a 1 when it actually received a 0, etc.

Clearly A cannot find out this by simply asking B and C.

Fascinating

Q How do we logically solve a logical impossibility?

A Bring in another impossibility and make it destructively interfere with the other impossibility.

Jerry beating Tom is a logical impossibility.

Similarly Jerry beating Spike is a logical impossibility.

Jerry ensures Tom and Spike destructively interfere.

consider the secure communication problem again.

Now it is impossible for information to travel faster than light.

Add a 2nd impossibility - it is impossible to stop the universe from expanding.

Now its impossible for an adversary sitting at a place where the universe expands faster than light to be able to eavesdrop on this information.

For practical purposes we use the fact that a DTM can't solve

NP-Hard problems. (another impossibility).

Fundamental

Any impossible problem in any field can be solved using IS, since IS itself doesn't care which field the original impossibility actually came from. For ex in coding theory, hashing and fingerprinting can beat the singleton-bound.

⇒ Clearly IS has applications in every field. This is as fundamental as it can get.

Consider randomized algorithms. Pseudorandomness is indistinguishable from a random variable. ⇒ Pseudorandom algorithms should serve as effectively.

In Mathematics, consider a solution to a problem like winning strategy for chess. The shortest proof would contain exp^o entries. Soln - Use interactive / zero knowledge proofs.

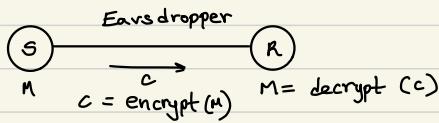
In This Course

1. Computational Hardness. (Resource Complexity)
2. Practical Uncertainties. (Noise, Routing uncertainty etc).
3. Natural Limits (speed of life / quantum)
4. Logical / Philosophical Impossibilities.

Today

1. Kerckhoff's Principle
2. Design / Breaking Ciphers.

Secure Communication Problem



S: Sender R: Receiver
 E: eavesdropper.
 M: message

Caesar's Cipher

Science = every man's art : no natural gifts required.

Goal of science - convert art to science.

Caesar Cipher is back from when the field was still an "art".

$$\text{encrypt}(c) = c + 3 \pmod{26}$$

For ex -

$$\begin{aligned} e(a) &= d & e(\text{hello}) &= \text{khoor} \\ e(b) &= e \end{aligned}$$

Kerckhoff's Principle

In Kannan's words (look at the book for the actual definition).

Security of a system must not depend on the **obscenity** of the algorithm, rather must only depend on the **secrecy of the key**.

Intuition -

Reason 1 - Algorithms can be reverse engineered.

2 - Updation/Recovery complexity - If your security is broken, you will have to change the entire algorithm. However if

you had the latter, changing the key is enough.

3 - Secure memory is costly.

Universal Turing Machine - takes a TM as input and simulates it.

$$U_{\text{TM}}(T, x) = T(x).$$

4 - For one node to communicate with multiple other nodes I would need a different algo for each pair of nodes. In the latter, I would only need a different key for each pair.
⇒ Much more scalable.

Kannan's Reasons -

Ethical Hacking: Arose as a consequence of Kerckhoff's principle. Expose the algo to ethical hackers and make them test security.

Standards: If every person on the internet uses the same algorithm (otherwise communication is impossible), naturally we must follow Kerckhoff. ⇒ A communication standard is .. impossible without following Kerckhoff.

Shift Cipher

$$e(x) = x + k \pmod{26}$$

→ Brute Force Attack - Try each of the possible keys.
Lesson: Use a large key space.

Consider -

$$a = 0, b = 1, \dots, z = 25$$

Let q_i = prob of i th char in C.

p_i = expected prob of i th char in message M.
(using some sort of prior knowledge).

Now $q_{(i+k) \mod 26} \approx p_i$ (because its a shift cipher).

Now based on prior knowledge we know $\sum_{i=0}^{25} p_i^2 = 0.065$

We can compute $\sum_{i=0}^{25} p_i q_{i+k} \approx \frac{1}{26}$ (if our guess of k is wrong)
 ≈ 0.065 (if our guess of k is right)

Mono-alphabetic Substitution cipher.

$e(a \downarrow b \downarrow c \downarrow \dots \downarrow z) = x \ D \ T \ \dots \ K$ Key is a permutation σ_{26} .

\Rightarrow There are $26!$ potential keys.

- \rightarrow Direct Brute force is hard since $26!$ is large. (Lesson Learnt).
- \rightarrow Frequency attack -

Notice that for $\forall i, \exists j$ s.t. $q_j \approx p_i$ (the one that it replaced)
consider this simple attack -

1. Sort the alphabet based on p_i
2. Sort the alphabet based on q_j
3. Now its likely the key is this permutation.

Vigenere Cipher

1. Choose a keyword. For ex - sea.
2. Add it to the message word.

h	e	l	l	o
s	e	a	s	e
z	i	l	d	s

Lesson Learnt: The key messes with the frequency of each character.

3. Decode with additive inverse.

This wasn't broken for ≈ 100 years.
Until -

Assume we know the length of the key. for ex - |sea| = 3.

\rightarrow Partition the ciphertext into $|k|$ parts. Each bucket contains all characters i.s.t. $i \in [1, |k|]$ are equal.

For ex if $|k| = 3$, bucket 0 : $C_0 C_3 C_6 \dots$

1 : $C_1 C_4 C_7 \dots$

2 : $C_2 C_5 C_8 \dots$

\rightarrow Notice each of these are by themselves outputs of some shift cipher. Break these individually using a modified frequency attack. i.e. instead of using q_i = frequency in the ciphertext, consider q_i = frequency in the bucket.

\rightarrow Brute force over the key length $|k|$ and repeat this.

\rightarrow Natural Extension: Make a collection of mono-alphabetic buckets.

A general pattern we notice in these is that people are creating-breaking-repeating. There seems to be no goal in sight. This is the indication that literacy about the field is lacking.

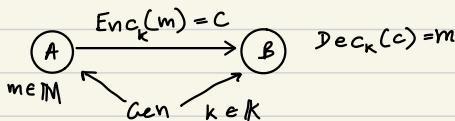
Today -

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

Shannon - Need a mathematical definition for "unbreakable" in order to create an unbreakable scheme.

Shannon's Perfectly Secure Scheme

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



An encryption is said to be perfectly secure if for all probability distributions over \mathcal{M} , and for all $m \in \mathcal{M}$, for all $c \in \mathcal{C}$ (where $p(\text{ciphertext} = c) > 0$) -

$$p(\text{message} = m \mid \text{ciphertext} = c) = p(\text{message} = m).$$

Intuitively: Looking at the ciphertext doesn't give me any information about the message extra than what I had beforehand (apriori).

Vernam Cipher.

$$\mathcal{M} = \mathcal{K} = \mathcal{C} = \{0, 1\}^n$$

Gen: $K \in_r \{0, 1\}^n$ (prob of each is 2^{-n})

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

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

Correctness here is fairly obvious since XOR is invertible.

we need to show that this conforms to Shannon's definition.

Digression: Proofs by induction are easy for the prover, but offers no intuition / what to do next etc.

contrarily, a proof by construction gives an element of intuition.

In information security - simply defining the problem is a contribution. However in Shannon's case, he managed to show that anything that met his definition ended up being impractical.

∴ Appreciate solutions that are thoroughly defined but at the same time can be practically achieved. (Workable + Comprehensive).

Vernam's Cipher meets Shannon's Definition -

consider an equivalent definition for Shannon's definition -

$$\text{if prob dist over } M, \forall m \in M, \forall c \in C, \\ p(C=c | M=m) = p(C=c).$$

Proof : We need to show $p(C=c | M=m) = p(C=c)$
 $\Leftrightarrow p(M=m | C=c) = p(M=m)$

Multiply both sides by $\frac{p(M=m)}{p(C=c)}$ we get,

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

$$\Rightarrow \frac{p(C=c, M=m)}{p(C=c)} = p(M=m) \quad (\text{Bayes' Theorem}).$$

$$\Rightarrow p(M=m | C=c) = p(M=m)$$

The reverse direction can be shown the same way.

lets consider a further equivalent definition -

Defn ③ $\forall c \in C, \forall m_0, m_1 \in M,$

$$p[c=c | M=m_0] = p[c=c | M=m_1]$$

Proof : Our second definition holds for all $m.$

$$\therefore p(c=c | M=m_0) = p(c=c)$$

$$\text{similarly } p(c=c | M=m_1) = p(c=c).$$

\Rightarrow Defn ② \Rightarrow Defn ③.

We now need to show ③ \Rightarrow ②,

$$\text{Now } p(c=c) = \sum_{m \in M} p(c=c | M=m) \cdot p(M=m)$$

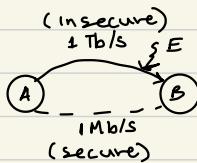
=

Now lets show Vernam meets this Defn ③,

$$\begin{aligned} p(c=c | M=m_0) &= p[c = m \oplus k] = p[k = c \oplus m_0] \\ &= \frac{1}{2^n} = \text{RHS}. \end{aligned}$$

So why is this impractical?

Need a secure channel to share the secure key.



Notice that for each bit of message I need 1 bit of key. This is as bad as sending the entire message on the secure channel.

This makes this 1-time pad really impractical.

So why is it popular? -

→ Consider a secure channel that is available when the message is not available. This sort of "lefers" the messages.

→ Sending messages to the same node. For ex - cases where some thing has to be removed and recovered.

Shannon also proved that any algo reaching his bound doesn't work practically.

This caused a sort of bifurcation in the field -

1. Slow Secure channel.
+
Fast Insecure channel. } Fast Secure (Symmetric Key Crypto) Channel.
2. Fast only insecure channel. ⇒ slow insecure channel. } Public Key Cryptography

Intuition of Shannon's Theorem -

For any perfectly secret encryption scheme $|IK| \geq |M|$.

(For Shannon, it was $H(IK) \geq H(M)$.)

But we can be satisfied by the above statement since $H(IK) = \log_2 |IK|$, and any compression on M only makes $|M|$ smaller.

Consider some ciphertext c . Construct $D = \{m \mid \exists K \in IK, Dec_k(c) = m\}$

Now, $|D| \leq |IK|$. Assume to the contrary that $|IK| < |M|$
 $\Rightarrow |D| < |M|$

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

Now $p(M=m^*) C=c) = 0 \neq p(M=m^*)$

This contradicts the fact that the scheme is perfectly secret.

Finite Fields (chiranjeevi)

Groups

- Non empty set G and an operator \cdot is a group if
- $\cdot \rightarrow$ associative. i.e. $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
 - $a \cdot b \in G$ for $a, b \in G$
 - $\exists 1 \in G$, $a \cdot 1 = a$ (Identity)
 - $\forall a \exists a' \in G$, $a \cdot a' = a' \cdot a = 1$ (Inverse)

Subgroups

$H \subseteq G$ s.t.

- for $a, b \in H$, $a \cdot b \in H$
- $1 \in H$
- $\forall a \in H$, $\exists a^{-1} \in H$, $a \cdot a^{-1} = a^{-1} \cdot a = 1$.

Cyclic Groups

$$\langle a \rangle = \{a^0, a^1, \dots, a^{i-1}\}$$

For ex - $(\mathbb{Z}_p, +)$

Ring

$(R, +, \cdot)$ is a Ring if -

- $(R, +)$ is an Abelian group
- $a, b \in R \Rightarrow a \cdot b \in R$ (closure)
- $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ for $a, b, c \in R$ (Associative)
- $(a+b) \cdot c = a \cdot c + b \cdot c$ } (Distributive)
and $a \cdot (b+c) = a \cdot b + a \cdot c$ }

Zero Divisors

$a (\neq 0) \in R$ if $\exists b \in R (b \neq 0)$, $a \cdot b = 0$ or $b \cdot a = 0$. (0 is identity wrt \cdot)

For ex - $(\mathbb{Z}_4, + \text{ mod } 4, \cdot \text{ mod } 4)$.

since $2 \cdot 2 \equiv 0 \pmod{4}$

Integral Domain

A commutative ring (cont. .) with no zero divisors.

Division Ring

An integral domain such that $(R - \{0\}, \cdot)$ is a group.

Field

$(F, +, \cdot)$ is a Field if -

i) $(F, +)$ is an abelian group.

ii) (F, \cdot) is an abelian group.

iii) $(a+b) \cdot c = a \cdot c + b \cdot c$ } (Distributive)

and $a \cdot (b+c) = a \cdot b + a \cdot c$

Characteristic of an integral domain

The least positive integer such that

$$a + a + \dots + a \text{ (m times)} = 0 \quad \forall a \in I$$

If no such m exists, then characteristic is 0.

Claim: If F is finite, characteristic = p for some prime p.

Proof: consider $1 \in F$.

consider $1, 1+1, 1+1+1, \dots$

Now at some point, since F is finite, nos begin to repeat.
Let the first no. after which it reaches 0 be n. Assume to the contrary that n is not prime.

Then n can be written as a product of 2 nos a, b.
s.t. $a \neq 0, b \neq 0 \Rightarrow (1+1+\dots \text{ a times}) * (1+1+\dots \text{ b times})$
But \Rightarrow either $(1+1+\dots \text{ a times}) = 0$ or $(1+1+\dots \text{ b times}) = 0$.

This violates that n was the smallest no. that $(1+1+\dots \text{ n times}) = 0$.

Let $(F, +, \cdot)$ be a finite field. Let $F \subset K$ where K is also a finite field. Then K has q^n elements where n is dimension K over F .

\Rightarrow every finite field has a prime no. of elements.

Today

- Two famous relaxations
- One universal assumption.

Shannon's Eavesdropper

The adversary in Shannon's case is unbounded.

Let's relax these conditions -

- i) The adversary is computationally bounded.
- ii) There is some finite probability of error.

Negligible Function: A function $M(n)$ is said to be negligible in n if for all +ve polynomials $p(\cdot)$, $\exists n_0$ s.t. $\forall n > n_0$ $M(n) \leq \frac{1}{p(n)}$

We model the adversary as a probabilistic polynomial turing machine (PPTM). \Rightarrow Adv tries to break in a polynomial no. of times.

Now we know (by asymptotics)

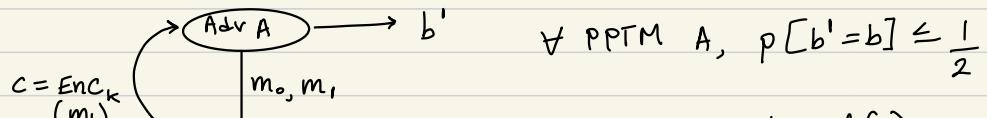
$\forall p(\cdot)$, $\exists n_0$, $\forall n \geq n_0$

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

Notice that PPTM Adv makes the two above assumptions -
polynomial \rightarrow comp bounded
probabilistic \Rightarrow non-zero prob(error)

Now if the adversary uses a larger polynomial, it is sufficient to change only the key length (or other security parameter).

Equivalent Defn for Perfect Security -



- m_0, m_1
1. Adv A chooses 2 messages m_0 and m_1 .
2. $Enc()$ chooses one of them at random and returns $Enc_k(C)$ of it.
3. A must know whether Enc chose m_0 or m_1 .
4. Scheme is perfectly secure if his guess is as bad as guessing at random.

A bunch of papers in this period achieved this definition by making assumptions of the existence of various mathematical objects like - one way functions, collision resistant hash functions etc.

Further papers showed that all these are equivalent. Thus we assume - One Way Functions exist.

This is still an open problem, but if it is assumed, we can still achieve this definition.

This assumption lead to -

i) Heuristic Security

- Algorithms that are unproven / security based on heuristics
- Efficient for terabytes / petabytes of data.

ii) Provable Security

- Assuming one-way funcs exist

iii) Proven Security -

→ Empty set. : 0

In this course we will focus on one-way functions and provable security.

One-Way Function (Kannan's Defn)

A function $f: \{0,1\}^* \rightarrow \{0,1\}^*$ is said to be one-way if -

- a) Easy to compute (polynomial in length of x)
- b) Hard to invert -

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

$\hookrightarrow x=y$. We write

$f(x)=f(y)$ to handle non-invertible f .

Existence of one-way functions $\Rightarrow P \neq NP$

Consider a one-way function f .

$x \longrightarrow f(x)$ is in P .

similarly,

$f(x) \longrightarrow x$ is not in P (not even in BPP)

If $f(x) \rightarrow x$ is in NP , then it $\in NP - P$ thereby separating P and NP .

Clearly this is in NP since \exists a verifier which can verify in poly time - given witness y , we can compute $f(y)$ and check if its equal to $f(x)$.

Verifier - For a language L , V verifies L if -

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

Today

- Pseudorandomness
- An encryption scheme
- Discrete Log Problem

Discrete Log Problem

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

↳ consider a brute force - if g is a generator, we'll have to try all p in the worst case.

Pseudorandom numbers

An efficient deterministic code G_i that inputs n -bit string u and outputs an $l(n)$ bit string is a pseudorandom generator (PRG) if -

a) $l(n) \geq n$ (Expansion)

b) $\forall PPTM D, |p(D(u_{l(n)}) \text{ accepts}) - p(D(g(u_n)) \text{ accepts})| \leq \text{negl}(n)$

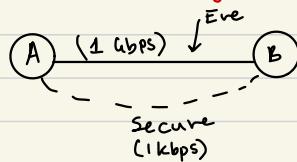
↳ Uniformly output of G_i on a uniformly distr. string of length n .

Intuition: No PPTM can distinguish bw an actually random string and one generated by the G_i .

distr. string of length $l(n)$

uniformly distr. string of length n .

What does this give us?

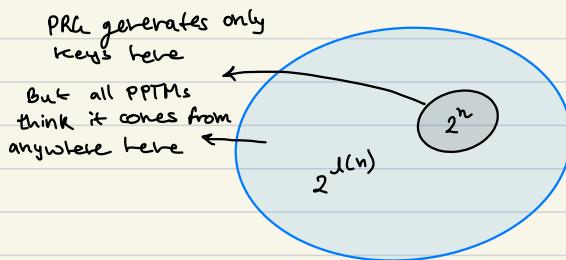


I only need to send my 'seed' on the secure channel. Now using this I can generate a key of a length $l(n)$

and use that on the main channel. Eve cannot figure out if we used an actually random key from $l(n)$ ($2^{l(n)}$ possibilities)

Notice that G_i is deterministic. \Rightarrow There will be only 2^n

keys that are actually used.



How will a distinguisher work?

Ask for samples. Notice that with $2^n + 1$ unique samples, by Pigeonhole principle, if it was pseudorandom, the no. will repeat. However, notice that no PPTM can query $2^n + 1$ times.

PRGs exist \Rightarrow One-way functions exist.

Consider an input s to a PRG G .

$$G(s) = y.$$

since G is a deterministic program. Given G and s , computing y is easy.

However since y is pseudorandom, finding s given G and y is hard. (need to try at least $2^n + 1$ where n is no. of bits in s .)

$\Rightarrow G$ is a one-way function.

If Discrete Log is a one-way func we can construct PRGs.

Consider a length preserving one-way func.

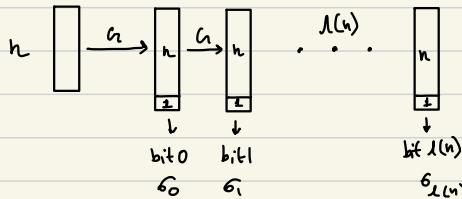
$$f: \{0, 1\}^n \rightarrow \{0, 1\}^n \quad (\text{i}) \text{ This } \Rightarrow \text{ a PRG.}$$

If I have a generator that expands by 1. i.e. $G: \{0, 1\}^n \rightarrow \{0, 1\}^{n+1}$

(ii) This gives us a generator $G': \{0, 1\}^n \rightarrow \{0, 1\}^{1(n)}$, $1(n)$ is poly in n .

(iii) DLP is a specific OWF $\Rightarrow G_1$.

Construction for (ii):



Assume to the contrary that $b_0 \dots b_{l(n)}$ is not pseudorandom.

Consider an expt where row 0 is $b_0 \dots b_{l(n)}$

row 1 is $r_0 \dots r_{l(n)}$ (where r is randomly sampled)

⋮

$d(n)$ is $r_0 \dots r_{l(n)}$

σ not being pseudorandom $\Rightarrow \exists$ a distinguisher can distinguish row 0 and row $d(n)$.

Now $\exists i$ such that this distinguisher can distinguish the rows i and $i+1$.

$$\begin{array}{c} r_0 \ r_1 \ \dots \ r_{i-1} \boxed{r_i} \ \dots \ r_{l(n)} \\ r_0 \ r_1 \ \dots \ r_{i-1} \ r_i \ \boxed{r_{i+1}} \dots r_{l(n)} \end{array}$$

\Rightarrow The distinguisher can diff b/w r_i and r_{i+1} . \Rightarrow it can distinguish the i th use of G_1 . Contradicts that G_1 is a PRG $_1$.

Now consider an OWFs $f(s)$ and $h(s)$ s.t. $f: \{0, 1\}^n \rightarrow \{0, 1\}^n$ and $h: \{0, 1\}^n \rightarrow \{0, 1\}^n$. $h(s)$ satisfies the prop -

Given s , easy to find $h(s)$.

$h(s) \xrightarrow{\text{Given } s, \text{ easy to find } h(s)}$ Given $f(s)$, $p[A(f(s)) = h(s)] \leq \text{negl}(1^n) + \frac{1}{2}$

Consider a $g: \{0,1\}^n \rightarrow \{0,1\}^{n+1}$
 $g(s) = \text{Concatenate}(f(s), h(s))$.

Such $h(s)$ are called **Hardcore predicates** of f .

consider a hardcore predicate of DLP.

Discrete Log Most Significant Bit.

Given $g^x \bmod p$ what is the most significant bit?

Thm: $DLP \leq_p DLPM$

PF Outline i) $DLPL$: Given $g^x \bmod p$ what is the LSB (x)?
 Prove that $DLPL \in P$.

ii) $DLPL + DLPM \Rightarrow DLP$.

Proof i) $y = g^x \bmod p$.

Now by Fermat's Little Thm states -

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

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

Now consider,

$$y^{\frac{p-1}{2}} \bmod p. \quad (p-1 \text{ is even since } p \text{ is prime})$$

$$= g^{x(\frac{p-1}{2})} \bmod p$$

Notice if x is even, $g^{c(p-1)} \bmod p \equiv 1$ where $c = \frac{x}{2}$.
 x is odd, $g^{(\frac{p-1}{2})} \bmod p$

$$= \sqrt{g^{p-1}} = \sqrt{1} = \pm 1.$$

but 1 occurs at $p-1$
 and its a permutation. \Rightarrow must be $p-1$.

Now we can use this to solve DLP.

Given a g^x , take $g^{x(\frac{p-1}{2})}$. If 1 LSB = 0
-1 LSB = 1.

Now an algo to find the rest of the DLP,

If x is even \rightarrow take square root to get $g^{\frac{x}{2}}$ and repeat.
 x is odd \rightarrow find ' g^{x-1} ' (find multiplicative inverse g^{-1}
 $g^{-1} \cdot g^x = g^{x-1}$). Now repeat the even case.

We know we can find inverse in $\log n$ time.

We just need to show sqrt can be done in P. i.e. we
need an algo that runs in $\log n$.

The ones that do run in $\log n$, we get 2 answers, \pm root.

This causes expo blow-up \Rightarrow beats the purpose.

If I had an algo to distinguish, i.e. is $x > \frac{p-1}{2}$

This can be found by checking if MSB of x is $\frac{1}{2}$ or not.
 \Rightarrow DLP M.

\Rightarrow These 2 together \Rightarrow DLP.

Today

1. CPA - Security (chosen plaintext attack)

Pseudo random Functions

Why? ←

PRFs are not sufficient since they do not scale. since PRFs gives a single use key.

Chosen Plaintext Attacks (CPA)

Recall that if the adversary sends two messages m_0 and m_1 . The $\text{enc}(m_b)$ is given to Adv where $b \in_R \{0, 1\}$. Adv should not be able to recognize if m_0 or m_1 was chosen.

For the CPA scenario, the adversary is given oracle access to the encryption function. It is CPA secure if the adv given oracle access to $\text{enc}()$ cannot recognize if m_0 or m_1 was chosen.

Thm: No deterministic $\text{enc}()$ algo is CPA secure.

Pf: Let $C_0 = \text{enc}(m_0)$.

$$C_1 = \text{enc}(m_1)$$

Now simply check if $\text{enc}(m_b) = C_0$, if so return $b=0$
else return $b=1$.

Notice PRF is a deterministic $\text{enc}()$. It is not CPA secure.

but using a non-deterministic $\text{enc}()$ is unintuitive since
decrypting it must be deterministic.

An award winning trick - Probabilistic Encryption

consider an $\text{enc}_k : \{0,1\}^n \rightarrow \{0,1\}^n$

choose $r \in_R \{0,1\}^n$ (called a Nonce).

Set $c = \langle r, m \oplus \text{enc}_k(r) \rangle$

$$\text{Now } \text{dec}_k(r, v) = v \oplus \text{enc}_k(r)$$

Notice that CPA attacks won't work since each time I query the oracle, the oracle chooses some random r , and returns the pair. If this r is different the Adv can't use this info.

Modes of operation (Block Ciphers)

Notice the above method is length doubling. This makes it seriously problematic in a practical setting. We now provide better modes of operation that use the trick while being almost length preserving.

- 1) Cipher Block Chaining
- 2) Output Feedback Mode (OFB)
- 3) Randomized Counter Mode.

Output Feedback Mode.

Let's define $r_i = \text{enc}_k(r_{i-1})$.

Now all I need to provide is r_0 .

i.e. $c = \langle r_0, c_1, c_2, \dots, c_t \rangle$

Intuition - Assume r_i is generated using r_{i-1} as a seed. Thus the generated r_i 's are pseudorandomly generated. Even though each r_i is not actually random, no PPTM can differentiate.

Randomized Counter Mode

Notice output feed back needs sequential decoding. To decode the i th message I need to decode over before it.
i.e. performance is stateful.

Randomized Counter uses the following -

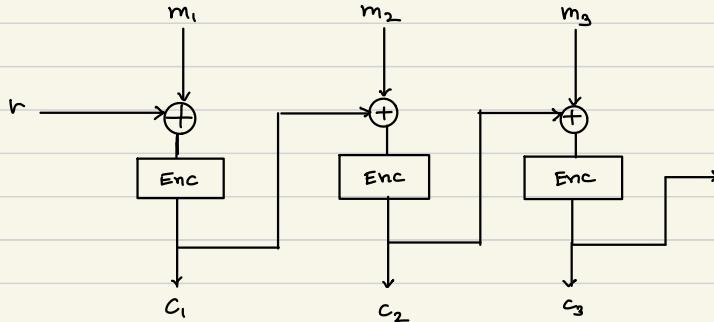
$$r_i = r_0 + i.$$

Notice that this is stateless in performance. However it is not obvious that this is secure.
proof to be done later.

Cipher Block Chaining (CBC).

For \leftarrow messages m_1, m_2, \dots, m_t
 $\langle r = c_0, c_1, c_2, \dots, c_t \rangle$

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



QUIZ Questions (2 revealed, 2 secret)

- 1) a) Show that shift/vigenere cipher is perfectly secret if one alphabet / message length \leq key word length is encrypted.
b) Secret from class notes.

- 2)
- a) Secret from class notes.
 - b) If $p-1$ can be written as $p-1 = s \cdot 2^r$ where s is odd,
then the $(r+1)$ th LSB is a hardcore predicate for
DLP.

Design a pseudorandom generator using this.

Today

- what are pseudorandom functions
- PRF iff PRGs.

Consider the prob encryption $c = \langle r, R_k(r) \oplus m \rangle$

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

Now there are a total of $2^{n \cdot 2^n}$ possible funcs.

Choose one of these uniformly at random.

However this needs at least $n \cdot 2^n$ length key.

If we have a pseudorandom function, that no PPTM can differentiate b/w a truly random function and pseudorandom function.

Unlike PRGs, notice we can't even "give the PPTM" a random function since specifying it would need $2^{n \cdot 2^n}$ which is non polynomial. We'll have to argue differently.

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

- \exists an efficient DTM that computes $F_k(x)$ (efficiency)
- \forall PPTM D , for sufficiently large n

$$|P[D(F_k(x)) = 1] - P[D(R_k(x)) = 1]| \leq \text{negl}(n)$$

↳ Input size is still n , but I can query the PRF and random function

PRF iff PRG

Thm:

PRFs exist if and only if PRGs exist.

i) PRFs \Rightarrow PRGs

$$s_i = F_k(s+i)$$

PRG $G(s) = F_k(s_0) \parallel F_k(s_1) \parallel F_k(s_2) \dots$

concatenate.

ii) PRGs \Rightarrow PRFs

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

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

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

$$\text{Hence } G(s) = G_0(s) \parallel G_1(s)$$

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

$$F_k(r) = G_{r_{n-1}} [G_{r_{n-2}} [\dots G_{r_2} [G_{r_1} [G_{r_0}(k)]] \dots]]$$

$$\text{e.g.: } |PPTM| / p(D^{F_k}(1^n) = 1) - p(D^{G}(1^n) = 1) \leq negl(n)$$

Assume to the contrary that \exists a PPTM D .

Base Case: For $|r| = 1, F_k(r) = G_{r_0}(k)$

Clearly if D can distinguish $F_k(r)$ then it can distinguish G .

Inductive Hypo: We cannot distinguish upto r of length n .

Consider r of length $n+1$.

If we can distinguish $F_k(r)$

Consider an OWF $f(x) = g^x \bmod p$
(DLP).

PKG $s_0 = S$
 $s_i = g^{s_{i-1}} \bmod p$

$$u(s) = \text{MSB}(s_1) \parallel \text{MSB}(s_2) \parallel \dots \parallel \text{MSB}(s_{2n})$$

$$F_k(r) = G_{r_{n+1}} [\dots h_{r_0}(k)] \dots]$$

$$\text{Enc}_K(m) = \langle r, F_k(r) \oplus m \rangle$$

$$\text{Enc}_K(m_0 \dots m_n) = \langle r_0, F_k(r_0 + i) \oplus m \rangle$$

Notice this entire scheme depends only on a single assumption:
i.e. OWF exist.

A specific OWF called Trapdoor OWFs. TOWFs gives rise to concepts like public key crypto and a bunch of other problems.

The assumption that TOWF exist gives rise to a world called "Crypto Mania".

If TOWF don't exist and OWFs exist this is "mini crypt".

If both don't exist, but on average case hard this is "Pessiland". (On avg, problem instances are hard).

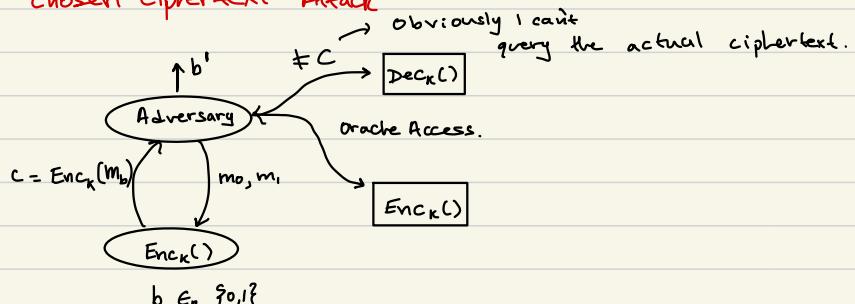
If only worst-case hard even in "Heuristica". (Hard to find a hard input)

If $P = NP$, then in "Algorithmica". This scenario is an apocalypse. Verifiers can produce proofs.

Today

- CCA Security
- MAC

CCA - Chosen Ciphertext Attack



$$b \in_R \{0, 1\}$$

$$\forall \text{ PPTM } A, \quad p(b' = b) \leq \frac{1}{2} + \text{negl}(|m_1|)$$

Consider our CPA secure modes of operation.

Choose m_0 as 0^n and m_1 as 10^{n-1}

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

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

$$c' = \langle r, \text{ToggleMSB}(m_b \oplus F_k(r)) \rangle$$

now the Adv can query the Decrypt Oracle

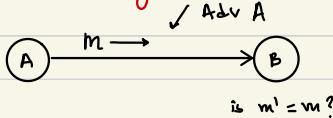
$$\begin{array}{ccc} \text{Deck}_k(c') & \xrightarrow{\text{if } b=0} & M_1 \\ & \xrightarrow{\text{if } b=1} & M_0 \end{array}$$

$$\text{Then I can choose } b' \text{ as} \\ b' = \begin{cases} 0, & \text{if } \text{Deck}_k(c') = M_1 \\ 1, & \text{if } \text{Deck}_k(c') = M_0 \end{cases}$$

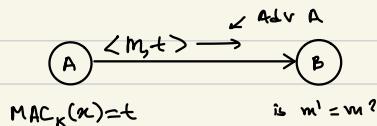
$$\Rightarrow p(b' = b) = 1.$$

In the real world: Adv has access to only an enc server. However this can be tweaked - flipping some bits. This can have consequences in areas such as money transfer.

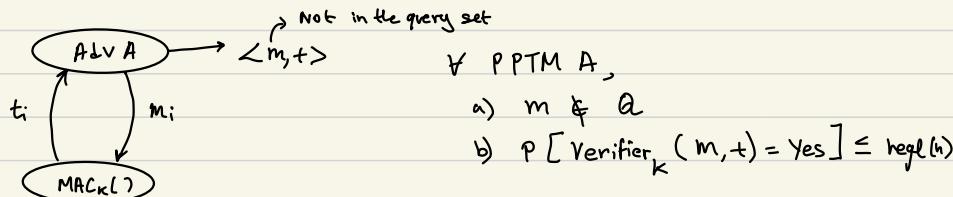
MAC - Message Authentication Codes.



B can't verify all the time. However we can define a test, s.t. not PPTM can fool the verifier.



Security of MAC



Encrypt Then Authenticate

If I have such MAC, I can get CCA security.

Consider

$$Enc_K(m) = \langle c, t \rangle$$

$\uparrow MAC_{K_2}(c)$

\downarrow

Any CPA-secure

$Enc_{K_1}(m)$

$Dec_K(m)$:

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

$Dec_{K_1}(c)$

Else

Reject.

Notice that this is CCA secure since the decryption server is useless.

Designing MAC from PRFs

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

If message size is = key length , i.e. $\{0,1\}^n$.

Then we can use $t = F_k(m)$.

But clearly this is impractical. Consider the following failed attempts -

1. $t = F_k(\oplus_i m_i)$

There are multiple way of getting the same xor.

2. $t = \langle t_1, \dots, t_n \rangle$

$$t_i = F_k(m_i)$$

Any permutation / subset of the message blocks will pass.

3. $t_i = F_k(\text{concat}(i, m_i))$

First notice, I can drop packets from the end and it will still pass.

4. $t_i = F_k(\text{concat}(1, i, m_i))$

Consider two messages -

$p_1 p_2 \dots p_e$ with tag $t_1 \dots t_n$

$q_1 q_2 \dots q_e$ with tag $s_1 \dots s_n$

$p_1 q_2 p_3 p_4 \dots p_e$ with tag $t_1 s_2 t_3 t_4 s_5 \dots s_n$

will pass.

There are actually 2^e possible interleavings. I can therefore get a lot of information.

A correct scheme

Let $t_i = F_k(\text{concat}(r, l, i, m_i))$
random no. chosen

for the entire message.
called the **Message Identifier**.

Claim: This scheme is a MAC for length n .

Proof (A): IF \exists PPTM that breaks this, it breaks F_k and hence
pseudorandomness of F_k is violated.

Proof: Suppose a PPTM A can produce $m_1, m_2 \dots m_1, t_1, t_2 \dots t_k$
↳ not in the query set.

Notice that if I query two messages that have the same r , I can
interleave the messages I can produce a valid m that is not
in the query set. However prob of getting the same r is negl.

Suppose now that any one of l, m_i were changed. Then being
able to find $F_k(\text{concat}(r, l, i, m_i))$ without knowing K ends up
contradicting the pseudorandomness of F_k .

CBC MAC

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

Let $t = t_q$ where $m = m_1 \dots m_q$

MID SEM Qn: Show that this is not CPC-Secure for variable length queries.

i.e. give two var len querys that an adv A can produce another message and its corresponding tag.

Possible Fixes - (all of these are state of art - however not used).

i) $t = F_{k_2}(t_q)$ Works even when length λ is not known.
 $t_q = \text{CBCMAC}_{k_1}(m)$ However key size doubles.

ii) $k^1 = F_k(\lambda)$
 $t = \text{CBCMAC}_{k^1}(m)$

iii) Prepend λ to message
 $t = \text{CBCMAC}_k(m')$
 $m' = \text{concat}(\lambda, m)$

Today

Today

Today

Mid Sem Review

A) Breaking Historical Ciphers

- A.1) Shift Cipher
- A.2) Mono-alphabetic substitution cipher
- A.3) Vigenere Cipher
- A.4) Kerckhoff Principle

B) Perfect Secrecy

- B.1) Definition
- B.2) Proof that Vernam cipher is P.S.
- B.3) Limitation of P.S.
 $(|K| \geq |M|)$

C) Relaxations to P.S.

- C.1) PPTM Adversary
- C.2) Negligible, non-zero error prob.
- C.3) Precise Assumptions.

D) Security against ciphertext only attacks.

- D.1) Defining Pseudorandom generators (PRGs)
- D.2) PRG \Leftrightarrow OWF (designing PRG using DLP)
- D.3) Encoder $C = g(k) \oplus m$ works.

E) CPA - Security

- E.1) Defining CPA secure encryption
- E.2) No deterministic scheme is CPA secure
- E.3) Probabilistic Enc.
- E.4) Pseudorandom Functions (PRF)
- E.5) PRF \Leftrightarrow PRG

E6) $C = \langle r, m \oplus F_k(r) \rangle$ works.

E7) Modes of operation.

E8) Feistel structure.

F) CCA-Secure and MACs

F1) Defining CCA security and MACs.

F2) MAC from PRF.

F3) CBCMAC

F4) CCA-Sec Enc from CPA-Sec Enc and secure MAC.

G) Hashing

G1) Define collision resistance.

G2) Birthday Attack

G3) Merkle - Damgård transform.

G4) Designing Hash functions from DLP.

H) Key Exchange.

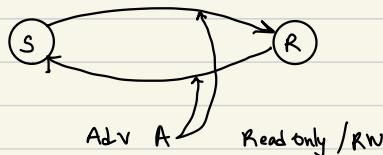
H1) Diffie Hellman

H2) DDH Assumption.

Mid Questions

H) Given a sender-receiver with one read only channel and one with read-write access (for the adversary). However we don't know which one is which.

Design a protocol to exchange a key regardless.



a) Design a modified Merkle-Damgård Transform which works when

$$h: \{0,1\}^{3n} \longrightarrow \{0,1\}^{2n} \curvearrowright \text{MMDT}$$

$$H: \{0,1\}^* \longrightarrow \{0,1\}^{2n}.$$

F) Break the basic CBCMAC scheme.

i.e. show the list of messages the adversary will query to create a message that will pass MAC.

In a normal CBCMAC, replay attacks are allowed. Why are they allowed in the textbook definition. (For ex reusing the tag is not considered a valid break, since a new message must be made by the adversary.)

E) OWF \Leftrightarrow PRG \Leftrightarrow PRF \Leftrightarrow Block Cipher.

Start from anywhere other than OWFs and build the rest.