LectureLet 11

Sep 28, 2020

Last time
- PRG
- 2-world
- OTP' (OTP+PRG)

G: PRG

Gen

$k \leftarrow Gen(1^n)$

Enc (expanding the key)

$c = G(k) \oplus m$

π

Dec

$m = G(k) \oplus c$

(G is a PRG) $\Rightarrow$ (Scheme is secure)

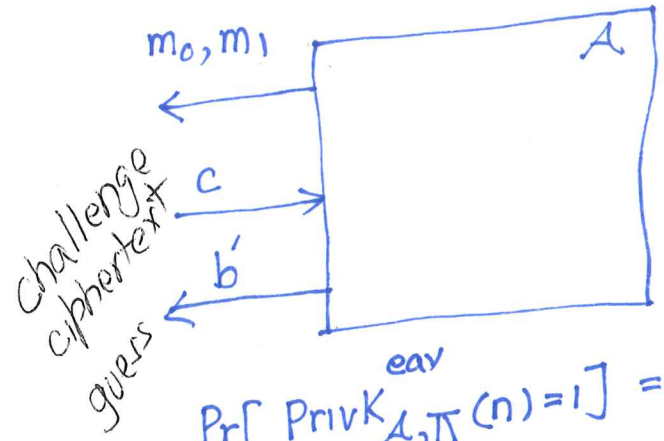
Use new definition

(scheme not secure) $\Rightarrow$ (G is not a PRG)

assume an attacker

$A \Rightarrow B$

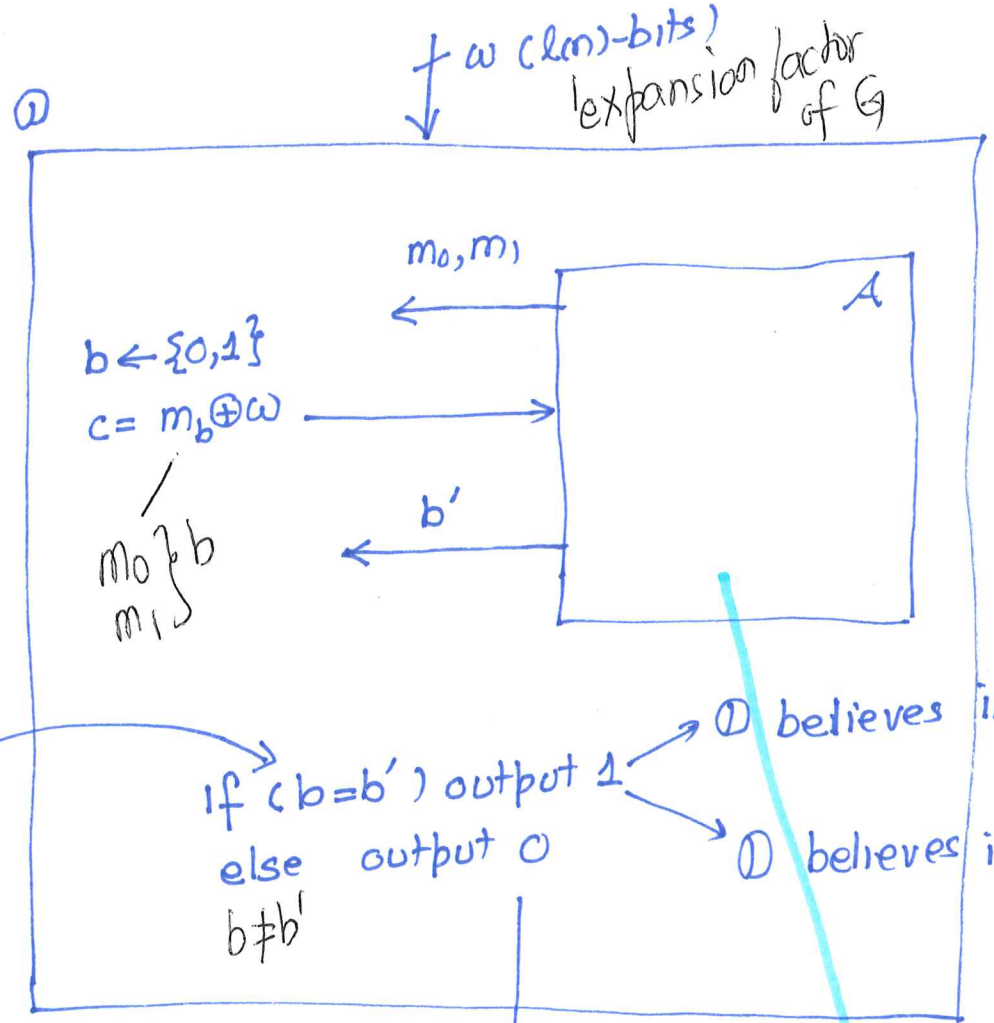
$\neg B \Rightarrow \neg A$



$\Pr[PrivK^{eav}_{A,\pi}(n) = 1] = \frac{1}{2} + f(n)$ $\neq$ negl. (say $f(n) = \frac{1}{n^{1000}}$)

$1/2 + \frac{1}{n^{1000}}$

Use A to find a distinguisher D for G.



D plays the indistinguishability game with A

prob. A winning the game.

world 0

seed

random n-bit ~~scheme~~.

$w = G(s)$

D behaves like TT

| OTP+PRG

$\Pr[D(G(s))=1] = \frac{1}{2} + f(n)$

world 1

$w \leftarrow \{0,1\}^{l(n)}$

D behaves like OTP

| perfect secrecy

$\Pr[D(w)=1] = \frac{1}{2}$

$\frac{1}{2} + f(n) - \frac{1}{2} = f(n) \neq \text{negl}(n)$

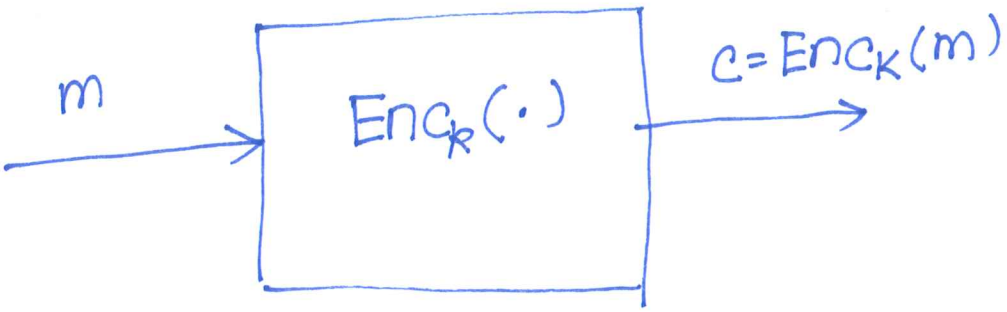
D is a distinguisher. PRG G

so far: ciphertext only

Chosen Plaintext Attack

adv. A capability to obtain encryptions of msgs of its choosing

Oracle access



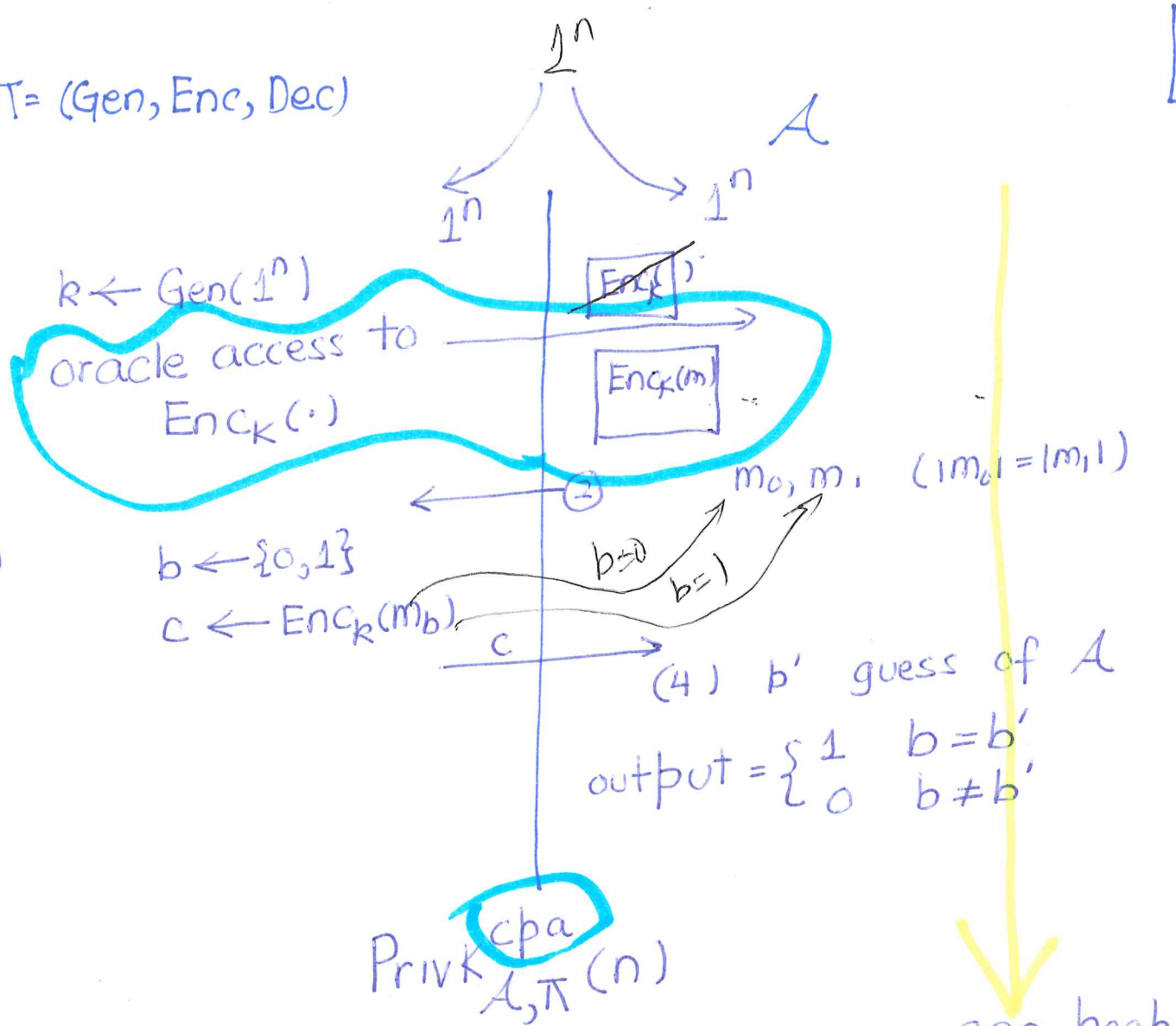
k

Blacbox access to A

Can't look inside.

I/o behavior of black box

$\Pi = (Gen, Enc, Dec)$



$\Pi = (Gen, Enc, Dec)$ is <u>CPA-secure</u> if for all PPT adversaries $\mathcal{A}$ there is negligible function negl s.t

$$Pr[PrivK^{cpa}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + negl(n)$$

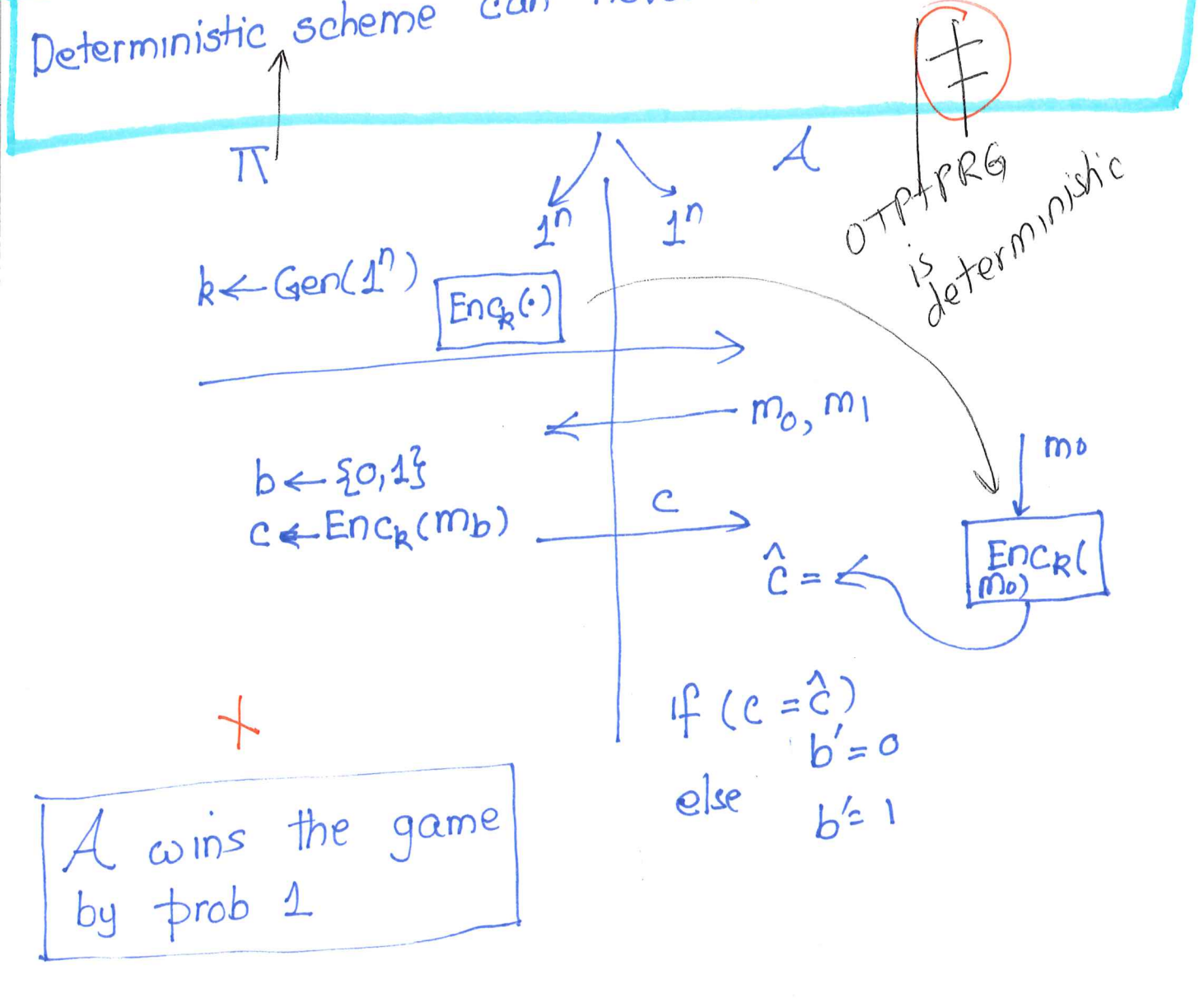
Deterministic encryption scheme

- encrypting the same msg again gives us the same ciphertext.

m = "I love beer"

$$= \begin{cases} Enc_k(m) \\ \vdots \\ Enc_k(m) \end{cases}$$

Deterministic scheme can never be CPA-secure.



A wins the game by prob 1

Last time

Proof for OTP+PRG → new indist game

CPA security → oracle access $(Enc_k(\cdot))$

Lecture Let 12

OCT 2, 2026

key

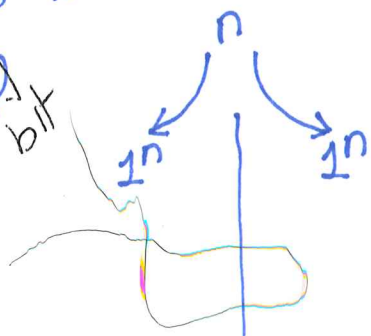
"left-or-right" oracle ~~$LR_{k,b}$~~

$LR_{k,b}(m_0, m_1)$ ⇝ returns $c \leftarrow Enc_k(m_b)$

generalization of $Enc_k(\cdot)$ oracle

$\Pi = (Gen, Enc, Dec)$

bit



$\mathcal{A}$

use once? previous CPA game

① $k \leftarrow Gen(1^n)$

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

oracle access

$\mathcal{A}$

use as many times as possible

$LR_{k,b}(\cdot,\cdot)$

$b=0$ $LR_{k,0}$

$b=1$ $LR_{k,1}$

guess $\mathcal{A}$ b'

$$output = \begin{cases} 1 & b = b' \\ 0 & b \neq b' \end{cases}$$

$\mathcal{A}$ wins

$PrivK^{LR\text{-}cpa}_{\mathcal{A},\Pi}(n)$ — cpa

Private-key encryption Π is CPA-secure ~~with~~ for multiple encryptions if for all PPT $\mathcal{A}$ there is a negligible function negl such that

$$Pr[PrivK^{LR\text{-}cpa}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + negl(n)$$

CPA-Secure $\Rightarrow$ CPA-secure with multiple encryptions.

Lecture Let 11

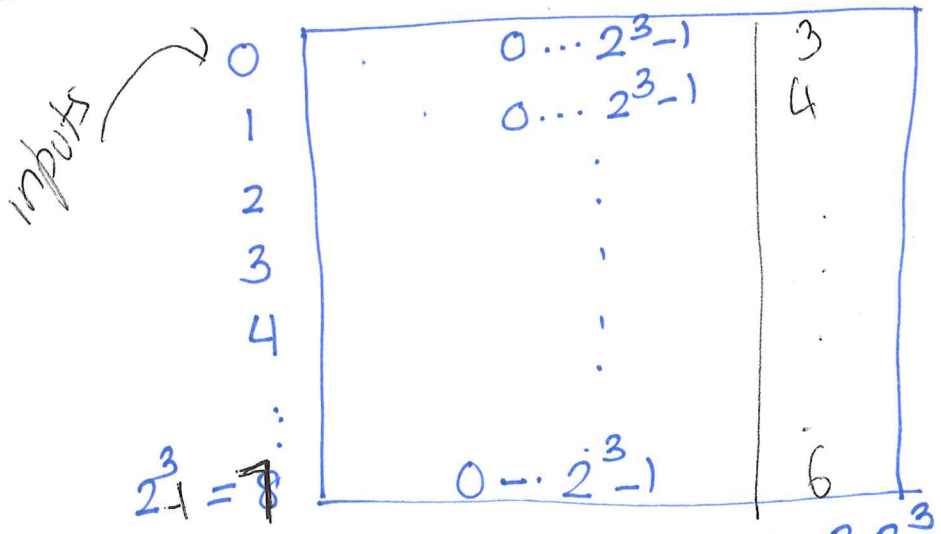
~~OPT~~ OTP + PRG $\neq$ CPA-secure (deterministic)

$f_n : \{0,1\}^n \rightarrow \{0,1\}^n$ ← all funcs from n-bits to n-bits

all n-bit strings $0 .. 2^n-1$

$0 .. 2^n-1$

$n=3$

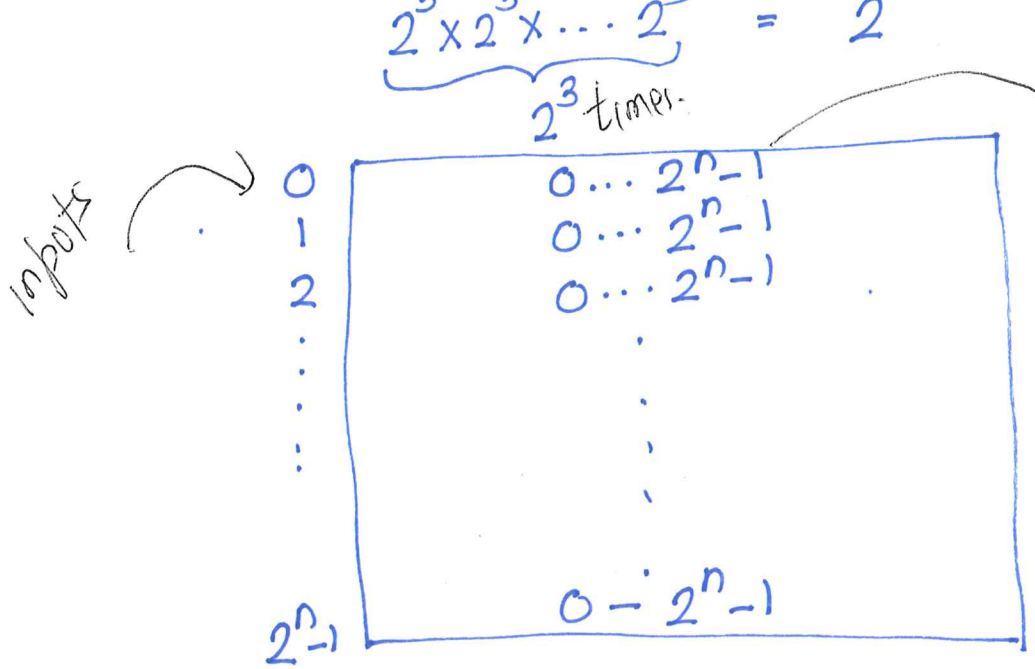


How many fns from 3 bits → 3 bits

How many ways can I fill this truth table?

$2^3 \times 2^3 \times \cdots 2^3 = 2^{3 \cdot 2^3}$

2^3 times.



$$\underbrace{2^n \times 2^n \times \cdots \times 2^n}_{2^n \text{ times}} = 2^{n \cdot 2^n} = |F_n|$$

P_n: all 1-1 functions

$= |2^n!|$ arrangements / permutations of 2^n objects

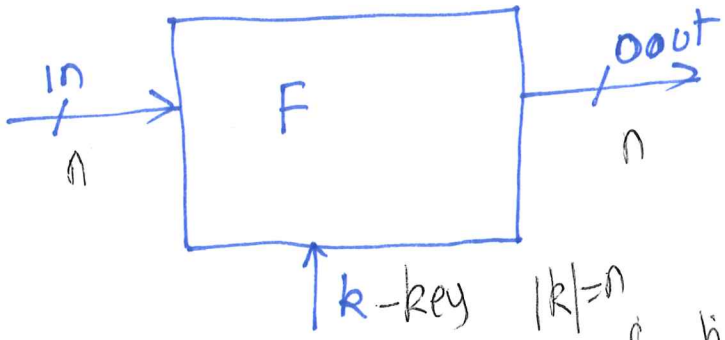
PRG:

pseudo random function

$F_k : \{0,1\}^* \longrightarrow \{0,1\}^*$

k ← key

- efficient
- length preserving



different key $k \Longrightarrow$ different function (keyed function)

Previous definition PRG does not work — subtle point

Size of $\{F_k$ is exponential in n

$O(2^n)$ a PPT A can't even read the truth table.

$O(n^2)$

Oracle access to the rescue.

$F: \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$

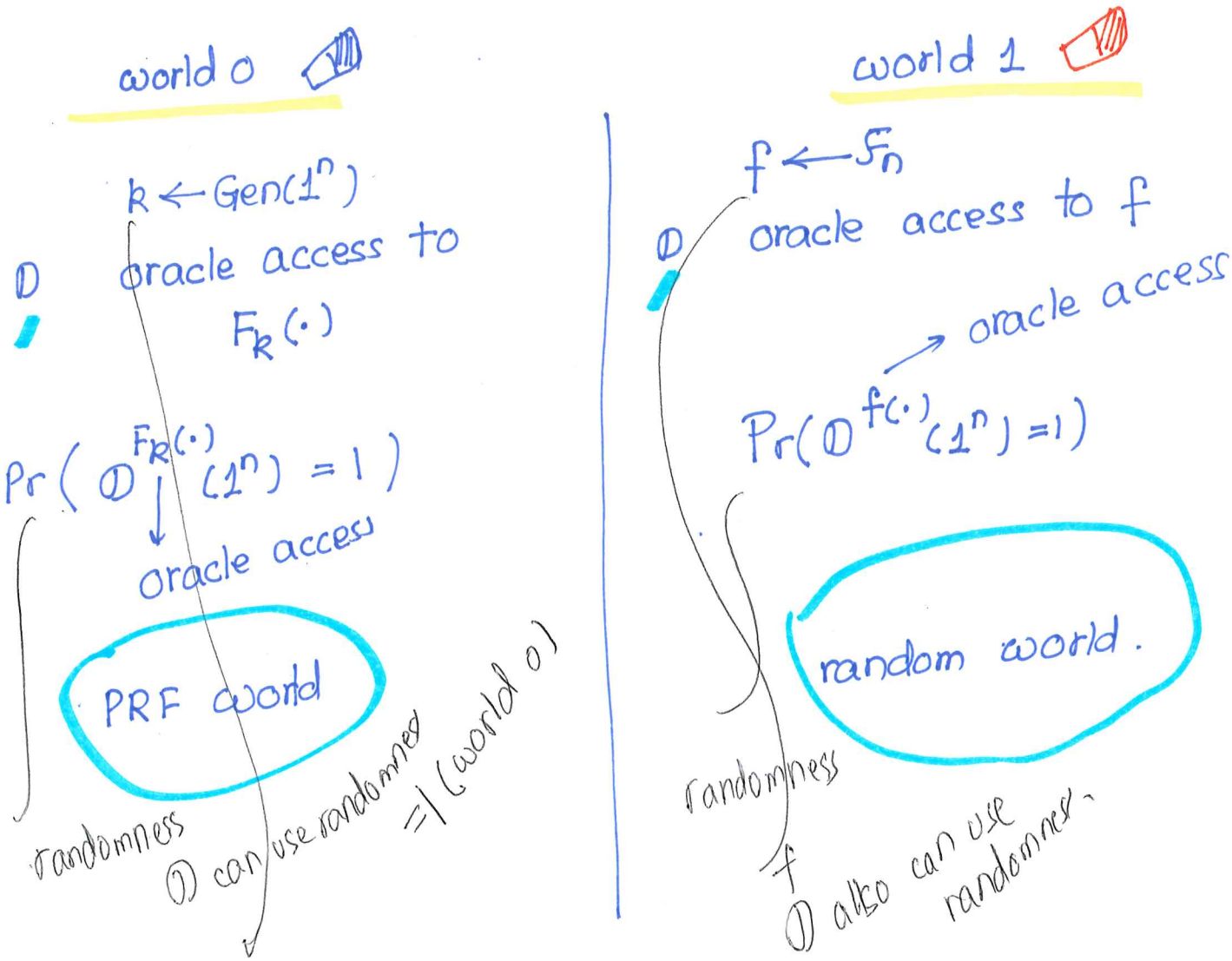
key $\uparrow$, $x \downarrow$, $y \downarrow$

$F(k,x) = y$

$F(10\cdots1, \cdot) = \cdot$

- efficient, length-preserving

D: distinguisher



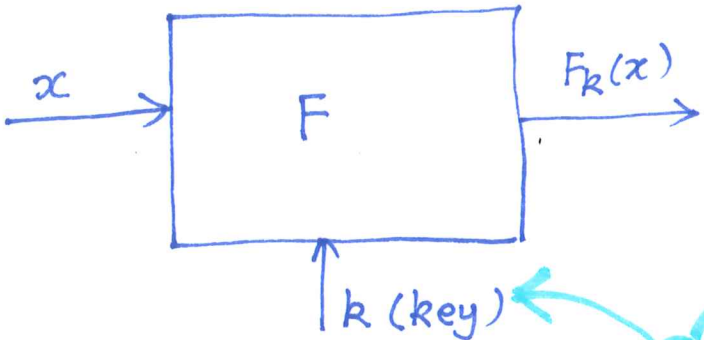
For all PPT D

$$|Pr(D^{F_k(\cdot)}(1^n)=1) - Pr(D^{f(\cdot)}(1^n)=1)| \leq negl(n)$$

Last time
- PRG's
- # of fns $\leq 2^{n2^n}$
- Two-world PRF

Lecturelet 13

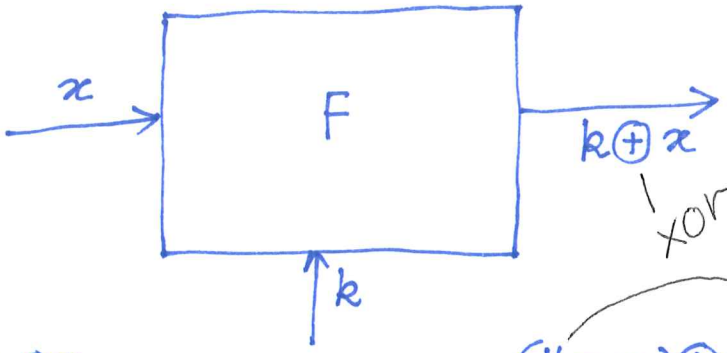
Oct 5, 2020



$F_R(\cdot)$ ← key

key cannot be revealed

Ex



$F(x_1) = y_1 = k \oplus x_1$

$F(x_2) = y_2 = k \oplus x_2$

$y_1 \oplus y_2 = (k \oplus x_1) \oplus (k \oplus x_2)$

$= x_1 \oplus x_2$ ← (*) not likely for a random fn

$x_1 \neq x_2$

$y_1 = O(x_1), y_2 = O(x_2)$

$y_1 \oplus y_2 = x_1 \oplus x_2$

$y_1 \oplus y_2 \neq x_1 \oplus x_2$ otherwise.

oracle

$\mathcal{D}^O(1^n) = \begin{cases} 1 & \text{"world 0"} \\ 0 & \text{"world 1"} \end{cases}$

world 0 (PRF world)

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

$O = F_k(\cdot)$ — oracle is the function $F_k(x) = k \oplus x$

$\Pr[D^{F_k(\cdot)}(1^n) = 1] = 1$ (*)

D always outputs 1

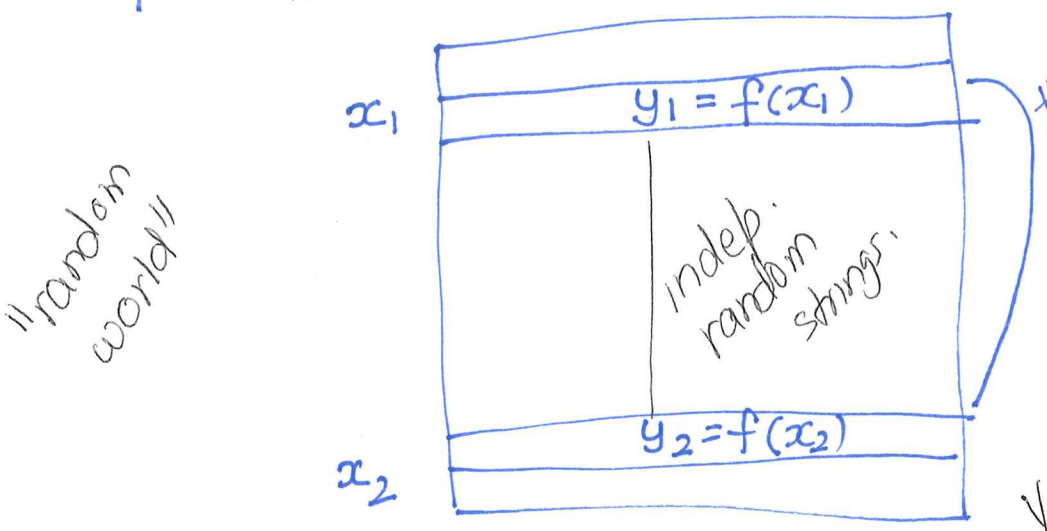
$y_1 \oplus y_2 = x_1 \oplus x_2$

$F_k(x) = k \oplus x$

$y_1 \oplus y_2 = x_1 \oplus x_2$ (proved earlier)

world 1

$f \leftarrow F_n$ (random function)



$y_1 \oplus y_1 \oplus y_2 = x_1 \oplus x_2 \oplus y_1$

$y_2 = x_1 \oplus x_2 \oplus y_1$ — $1/2^n$

y_2: $1/2^n$; $x_1 \oplus x_2 \oplus y_1$: fixed

$\Pr(f(x_2) = x_1 \oplus x_2 \oplus y_1) = 1/2^n$

$\Pr[D^{f(\cdot)}(1^n) = 1] = 1/2^n$ (**)

$(*) - (**) = 1 - \frac{1}{2^n} \neq \text{neg}!$

$F_k(\cdot)$ permutation/one-to-one

$F_k^{-1}(\cdot)$: inverse of $F_k(\cdot)$

$$F_k^{-1}(F_k(x)) = x$$

Block ciphers → PRFs

block size: 128 bits
Key size: 128, 192, 256 bits } AES

Goal: use $F_k(\cdot)$ to construct CPA-secure scheme

First attempt

Gen: $k \leftarrow \text{Gen}(1^n)$ $\{0,1\}^n$

Enc: $c = F_k(m)$

Dec: $m = F_k^{-1}(c)$ — $F_k(\cdot)$ is 1-to-1

Sanity check ✓

The scheme is deterministic ✗

$\neq$ CPA secure.

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

Enc: $m \in \{0,1\}^n$

choose $r \in \{0,1\}^n$ uniformly randomly

$c := \langle r, F_k(r) \oplus m \rangle$

PRF

k: key

$2n$ bits

Dec: $c = \langle r, s \rangle$

$m := F_k(r) \oplus s$

Sanity check:

$$F_k(r) \oplus (F_k(r) \oplus m) = 0 \oplus m = m$$

Is it deterministic?

$c := \langle r, F_k(r) \oplus m \rangle$

$\neq$ (except with $1/2^n$ prob)

$c' := \langle r', F_k(r') \oplus m \rangle$

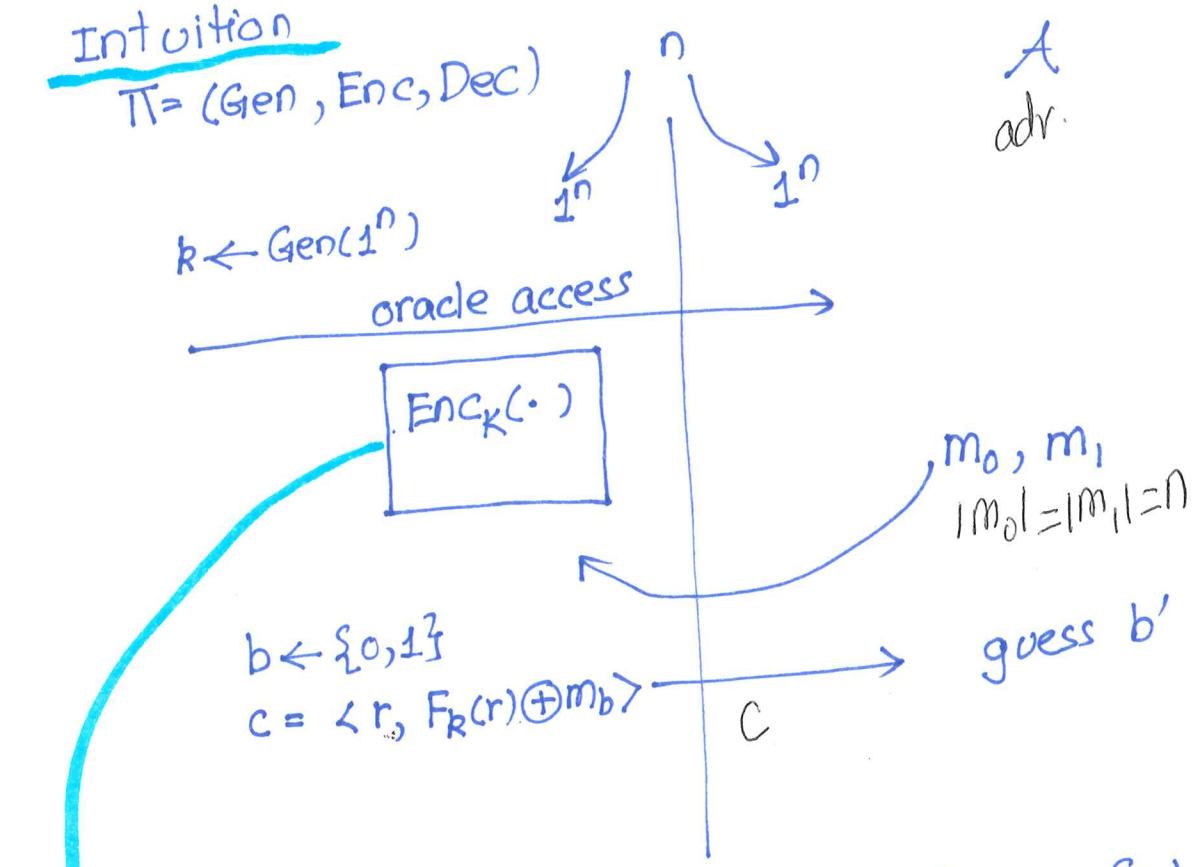
negl.

$1 - \frac{1}{2^n}$

The scheme is CPA-secure

Intuition

$\Pi = (Gen, Enc, Dec)$



A: encrypt m_0 several times using oracle

$c_0 = \langle r_0, F_k(r_0) \oplus m_0 \rangle, \ldots\ldots \langle r_{\ell-1}, F_k(r_{\ell-1}) \oplus m_0 \rangle = c_{\ell-1}$

use this oracle

check whether $c_0 = c$, or, $\ldots$, $c_{\ell-1} = c$ — T → guess $b'=0$; F → guess $b'=1$

$P(r = r_0 \text{ or } \ldots r = r_{\ell-1}) \leq \frac{\ell}{2^n}$ negl.

bad for us, good A

otherwise no help for A

Last time
- Ex. of PRF
- CPA secure scheme
- Proof intuition

LectureLet 14

Oct 6, 2020

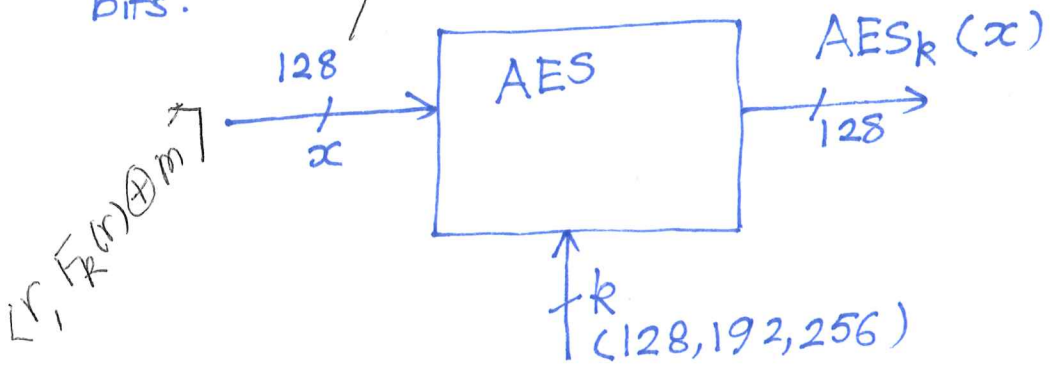
PRF

$F_K(\cdot)$

Block cipher (AES) — Advanced Encryption cipher

Encrypt → fixed size blocks

block size: 128 bits
key size: 128, 192, 256 bits.



Modes of operation

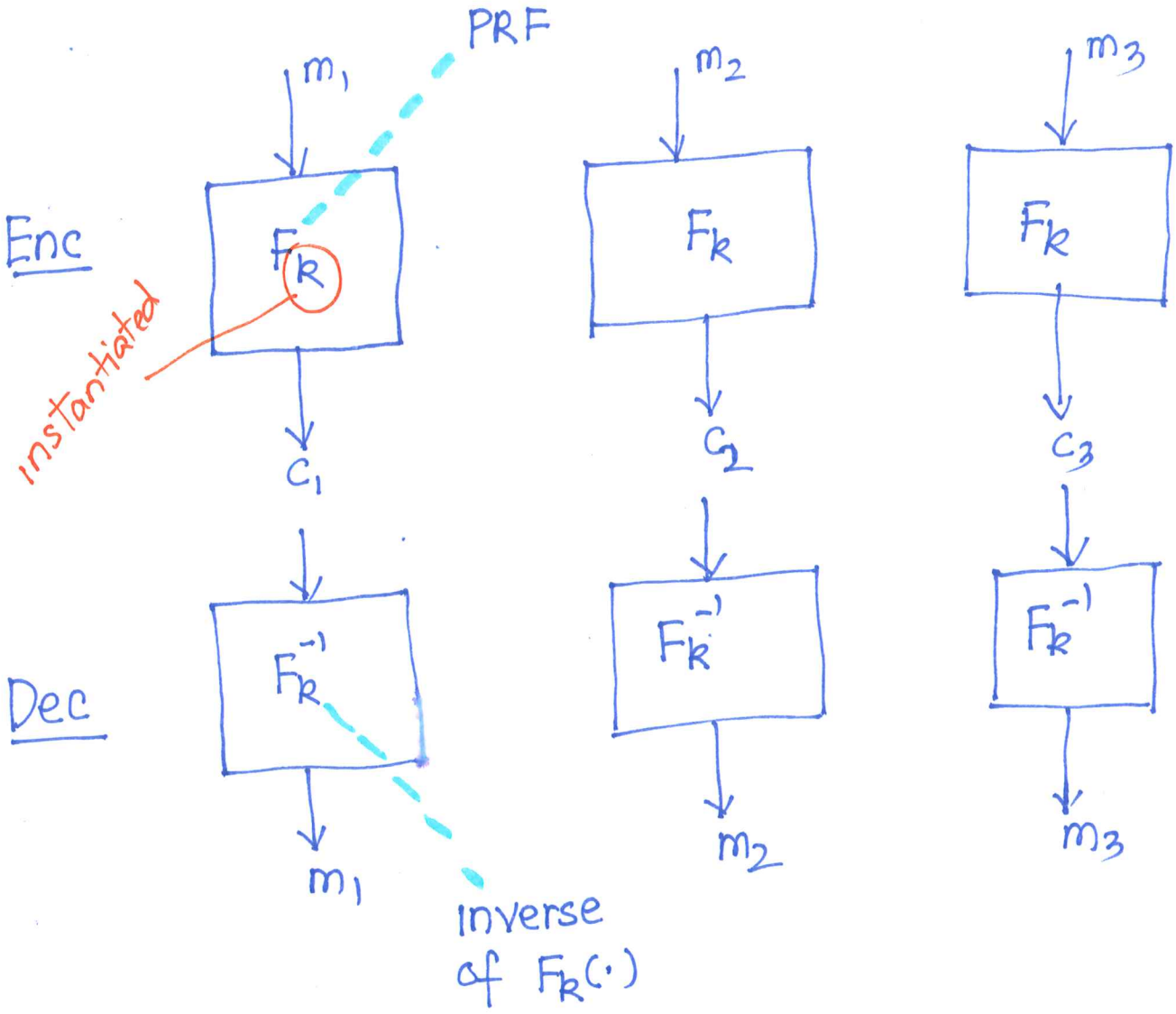
PRF with fixed block size (Ex: 128 bits) → MOP → Encrypt large Messages (Ex: 400MB)

$m = m_1 \; m_2 \; \cdots \; m_k$

msg (400MB)

divide into blocks (Ex: 128-bit blocks)

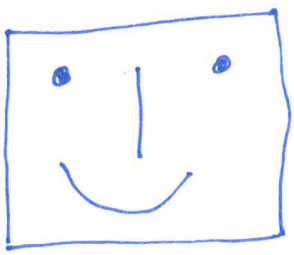
Electronic Code Book (ECB)



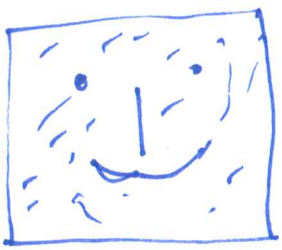
Weakness

- deterministic $\Rightarrow$ NOT CPA-secure

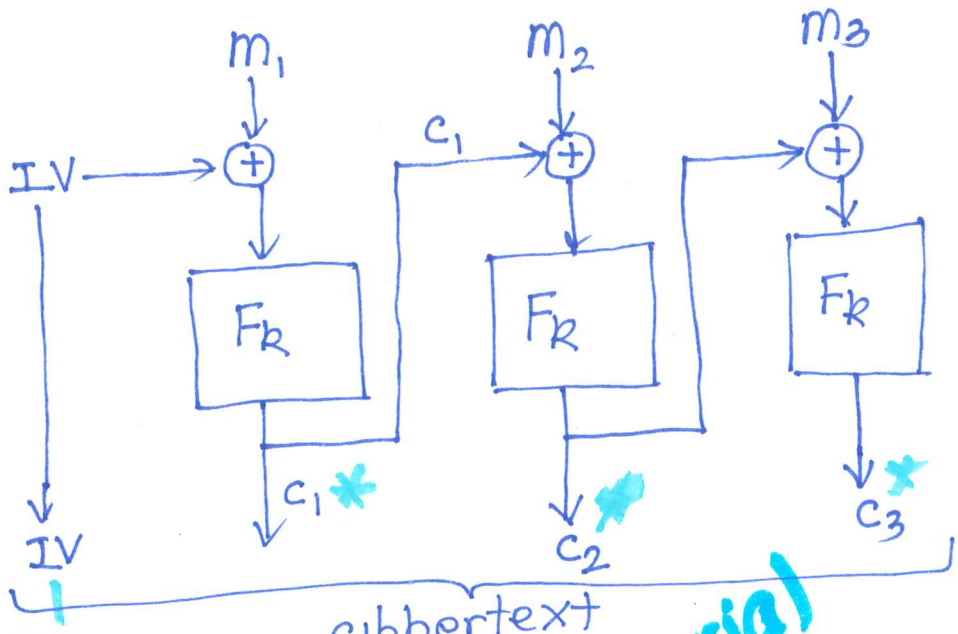
- $m_1 = m_2 \Rightarrow c_1 = c_2$



ECB ----→



Cipher Block Chaining (CBC)



Enc

Dec (HW)

$[r, F_k(r) \oplus m]$

Initial value (picked randomly each time)

Good things :)

- Not deterministic and CPA secure
- $m_1 = m_2$

$c_1 = F_k(m_1 \oplus IV)$) $\neq$

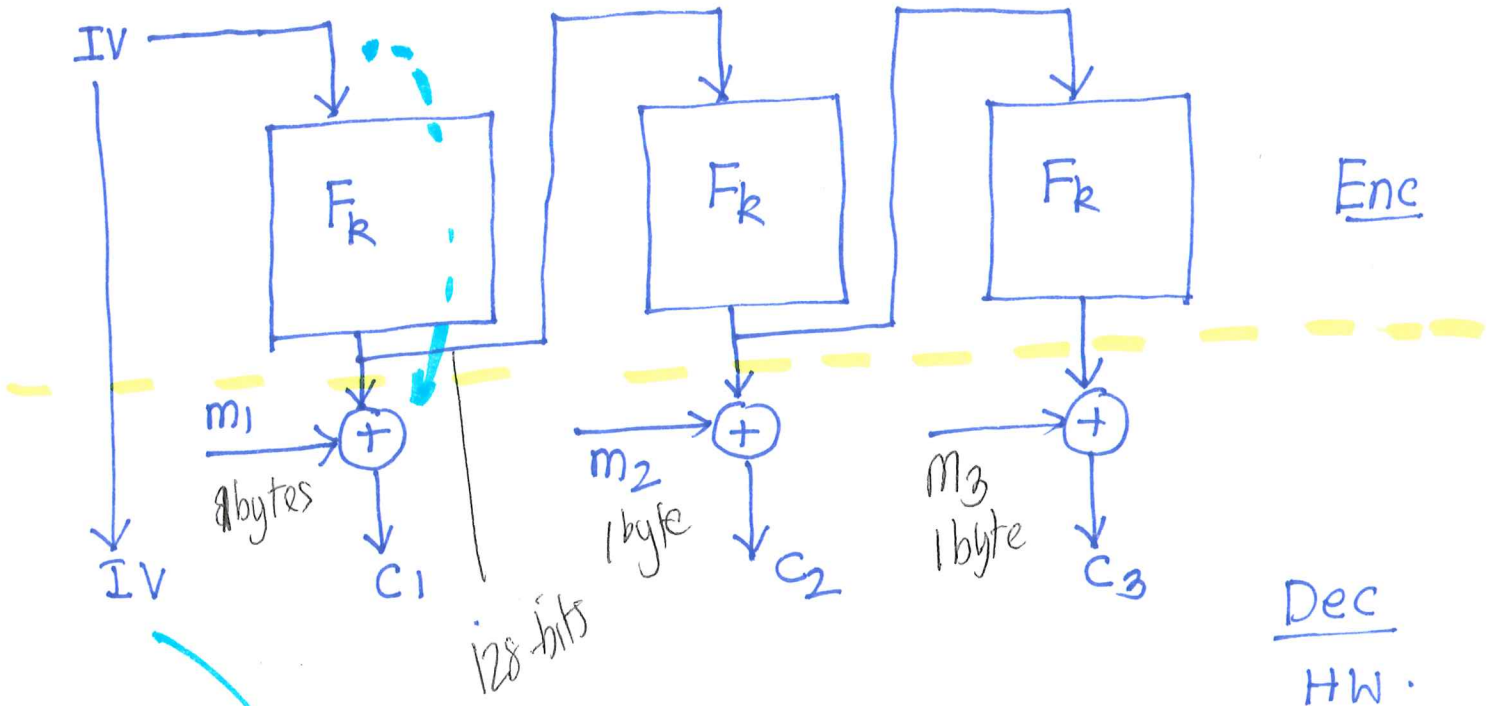
$c_2 = F_k(c_1 \oplus m_2)$

Bad things :(

errors propagate

Parallelizing is hard

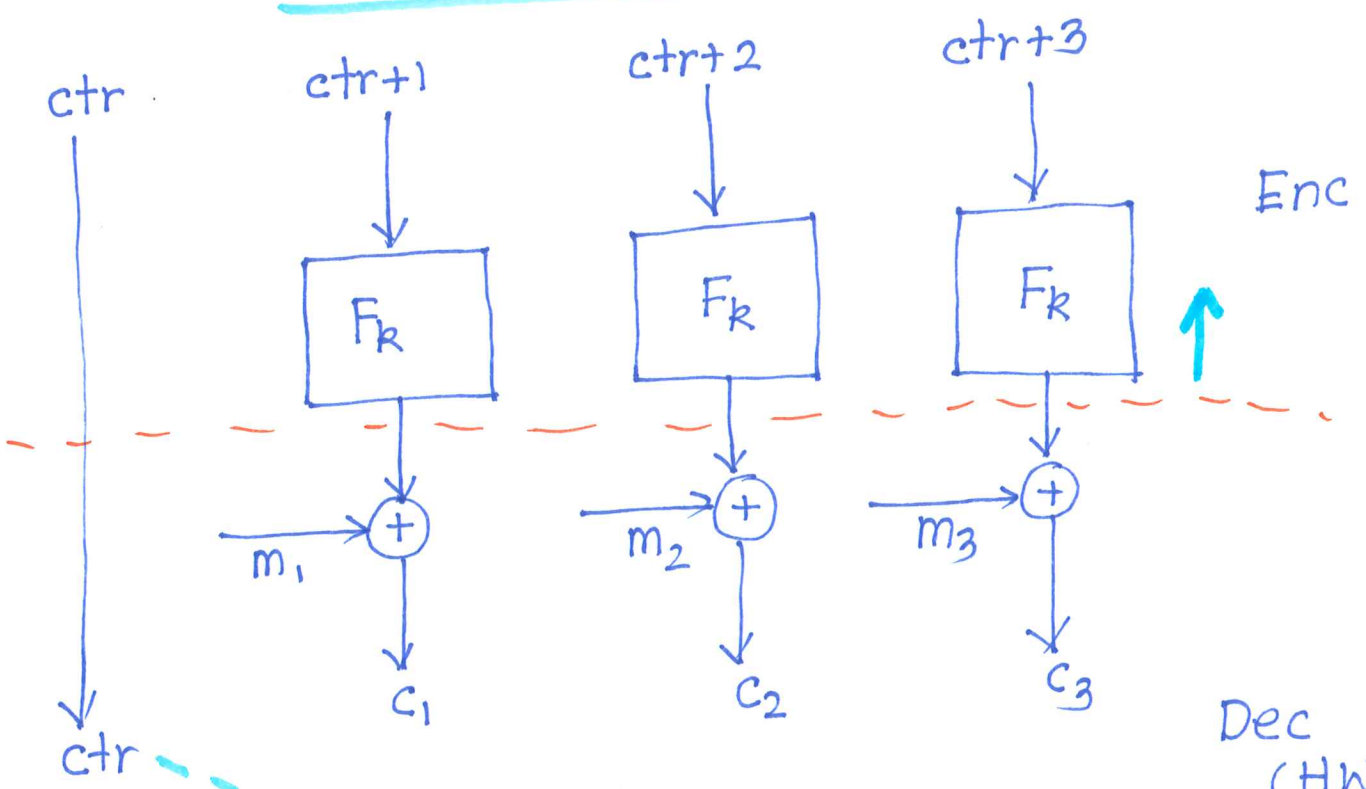
Output Feedback mode (OFB)



Good things :)

- Not deterministic / CPA secure
- Parallelizable
  - everything above only depends on IV, k
  - can precompute
- Can be used as stream cipher
  - video streaming

Counter mode (CTR)



pick randomly

Good things

- parallelizable
- CPA secure
- Simple to analyze
- Read the proof (Not required for the class) ✗

AES
- Rijndahl

OFB-AES

CTR-AES — PRF

MOP

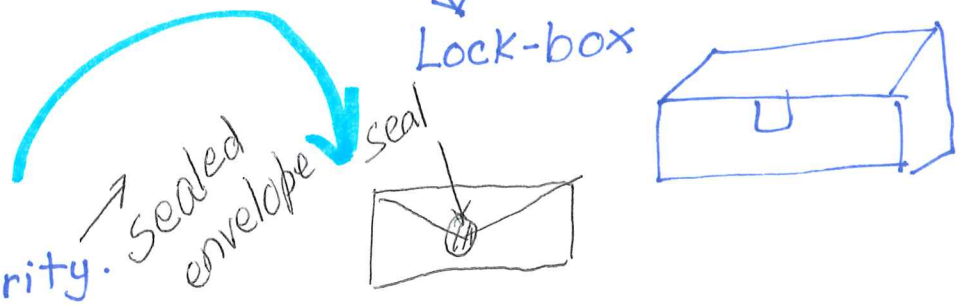
Oct 9, 2020

Last time
- Modes of operation (MOO)
- ECB, OFB, CBC, CTR
  X

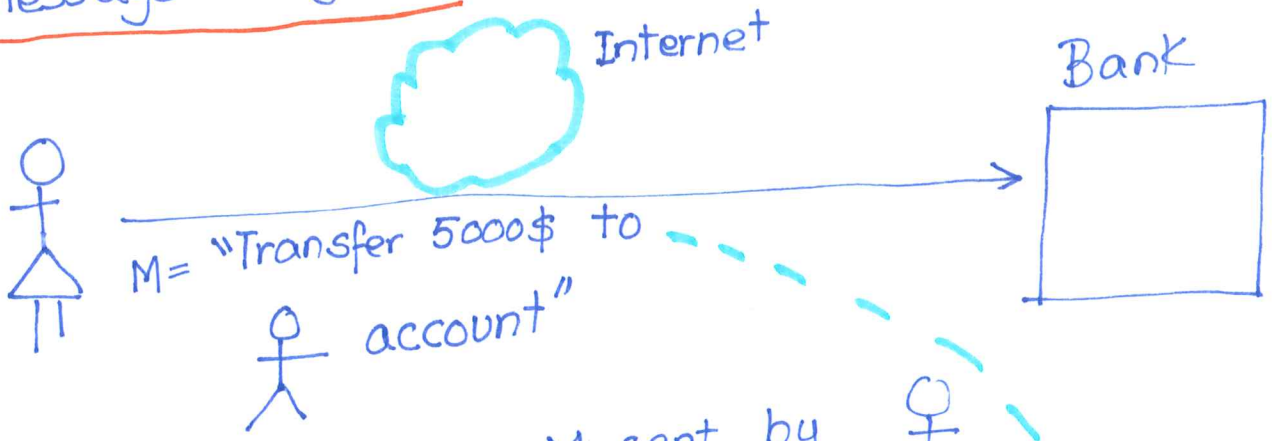
Lecture Let 15

Encryption schemes provide secrecy

physical analogy



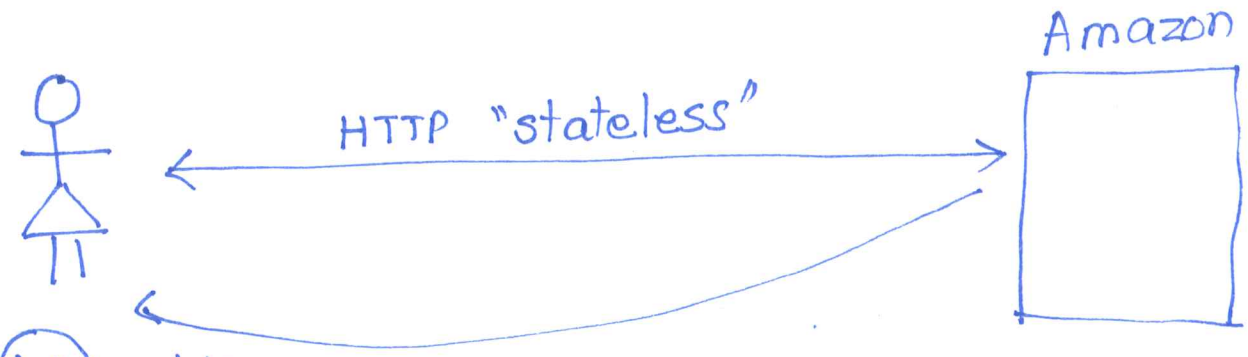
Message Integrity.



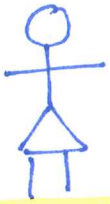
- Authentication: was M sent by
- Integrity: did some one modify M enroute to the bank

"Transfer 10000 $ to
account"

- error correcting codes (Ecc)
  - not enough
  - only corrects random errors (malicious)



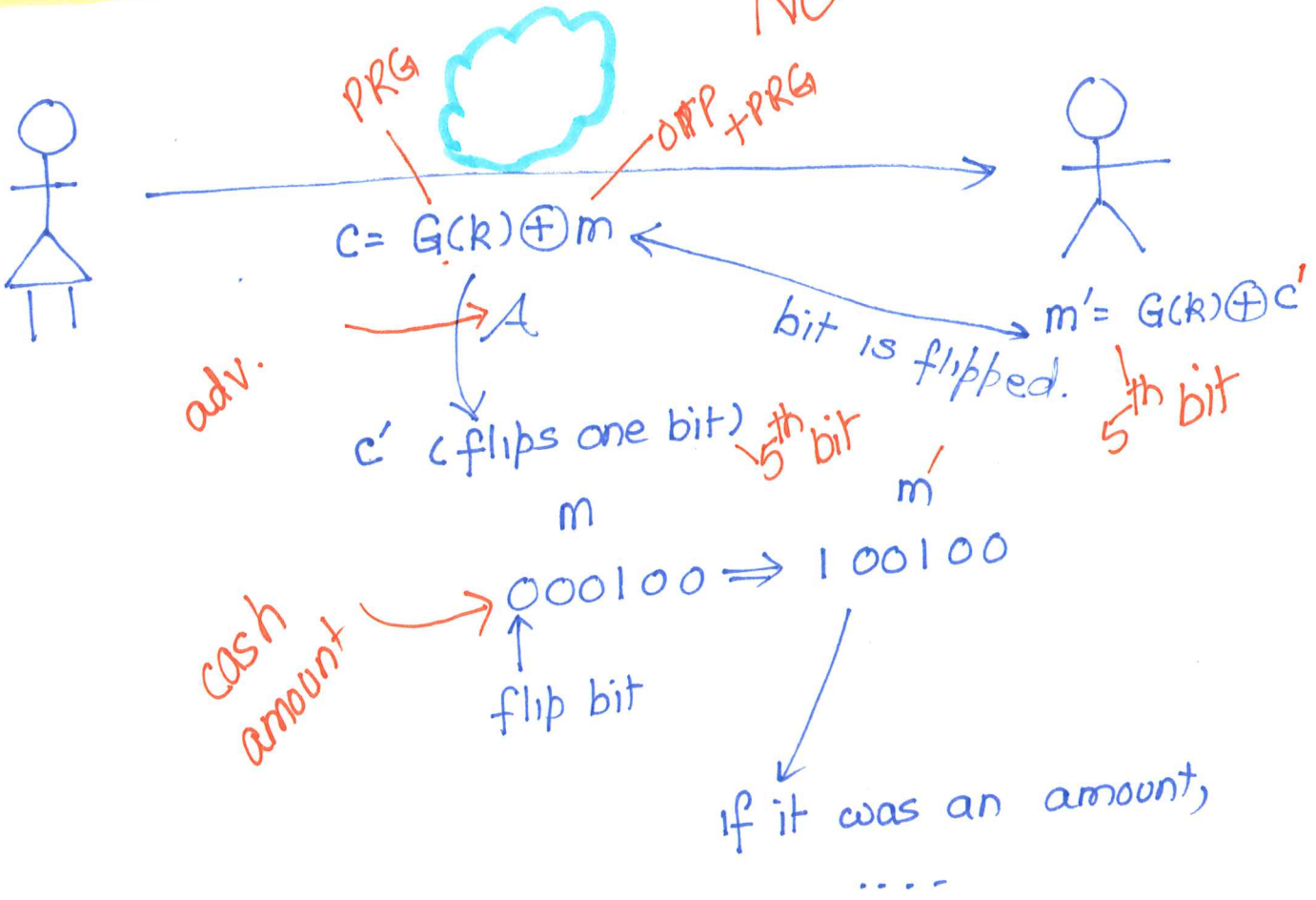
cookie
-state of the shopping cart
-prices of items



should not be able to tamper with the cookie

Encryption does not provide msg integrity.?

NO



Message Authentication Code (MAC)

security parameter
Key size

Gen: $k \leftarrow Gen(1^n)$

$|k| \geq n$ ← key

↓ Key size in bits (usually $|k| = n$)

encryption scheme

Mac (tag generation)

$t \leftarrow mac_k(m)$

tag

$m \in \{0,1\}^*$ — arbitrary len

$t \in \{0,1\}^{\ell(n)}$

fixed-length Mac

$\ell(n) = n^3$

$n = 1000, 1000^3$

Vrfy (deterministic verification)

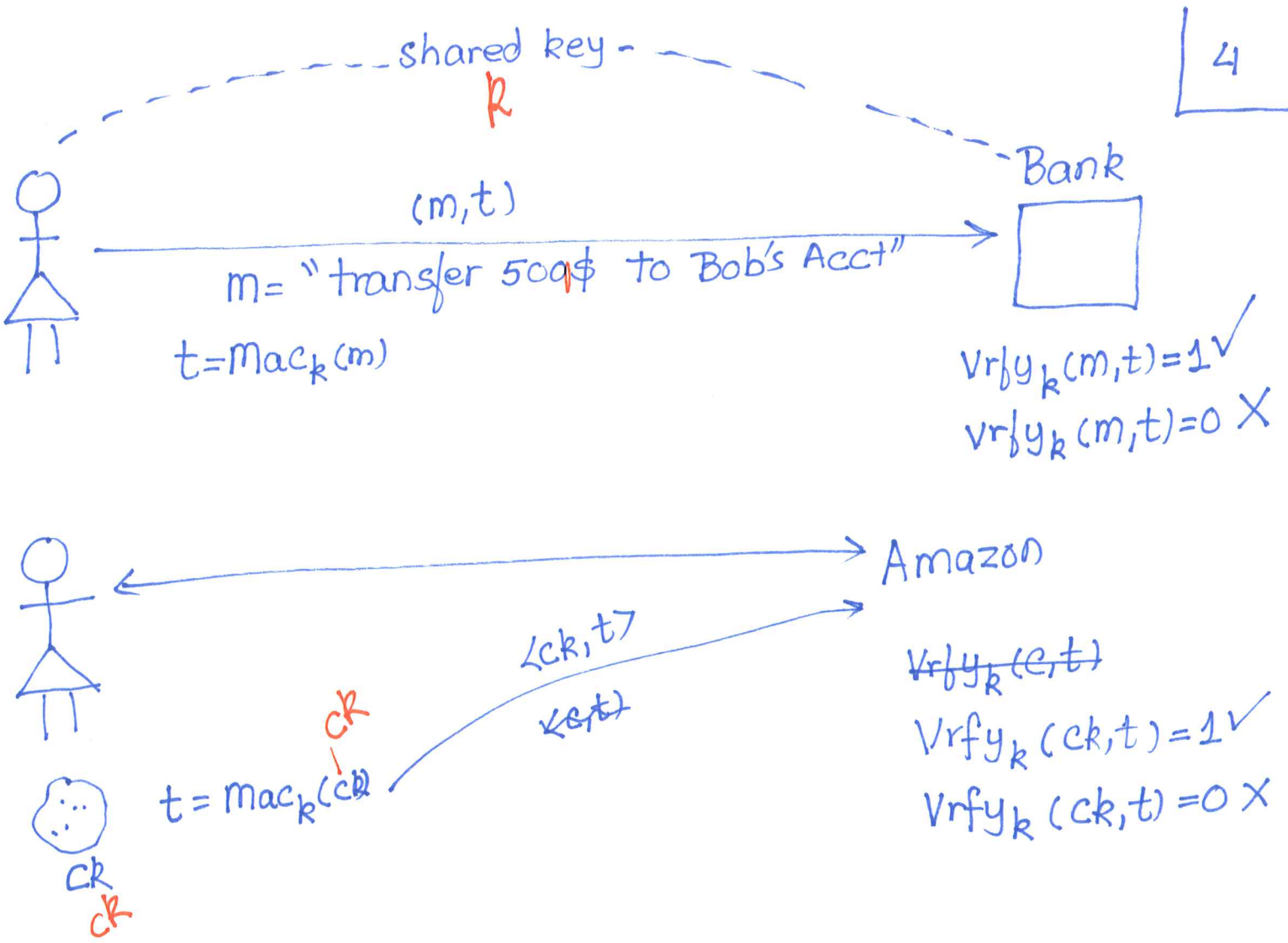
$b := Vrfy_k(m, t)$ — msg, tag

→ 1 (valid) ✓

→ 0 (invalid) ✗

Sanity check:

$Vrfy_k(m, mac_k(m)) = 1$ ✓



Canonical verification

Bank

- compute $t' = Mac_k(m)$
- check $t' = t$?

$\langle m, t \rangle$ →

$t' = t$ → 1 ✓

$t \neq t'$ → 0 ✗

Next time

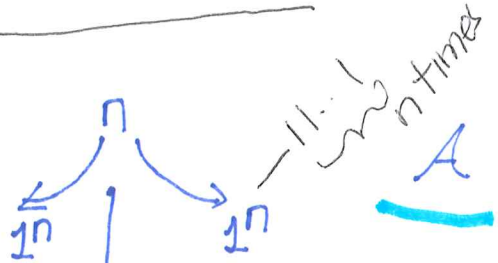
Security of Mac

Last time
- message integrity
- Def of mac

Lecture Let 16

Date 10/11/20

$\Pi = (Gen, Mac, Vrfy)$



① $k \leftarrow Gen(1^n)$

② Oracle Access → Q: set of queries A asks of the oracle

$Mac_k(\cdot)$

queries → $Q = \{m_1, \ldots, m_\ell\}$

answers ← $t_1, \ldots, t_\ell$

outputs: (m,t) ← forgery

$\notin$

- A succeeds (output of game 1)
  - [1] $Vrfy_k(m,t) = 1$
  - [2] $m \notin Q$
- A does not succeed (output of game 0)

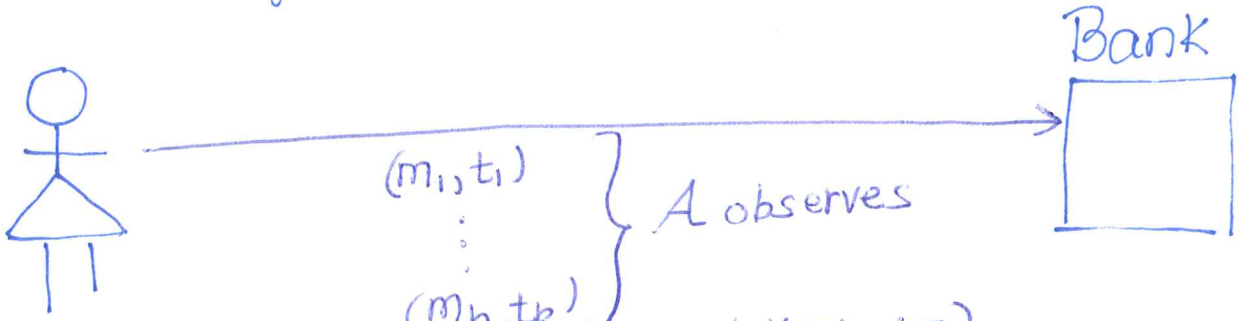
A winning?

$\text{Mac-forge}_{A,\Pi}(n)$

$\Pi = (Gen, Mac, Vrfy)$ is secure if for all PPT adversaries A, there is a negligible function negl s.t

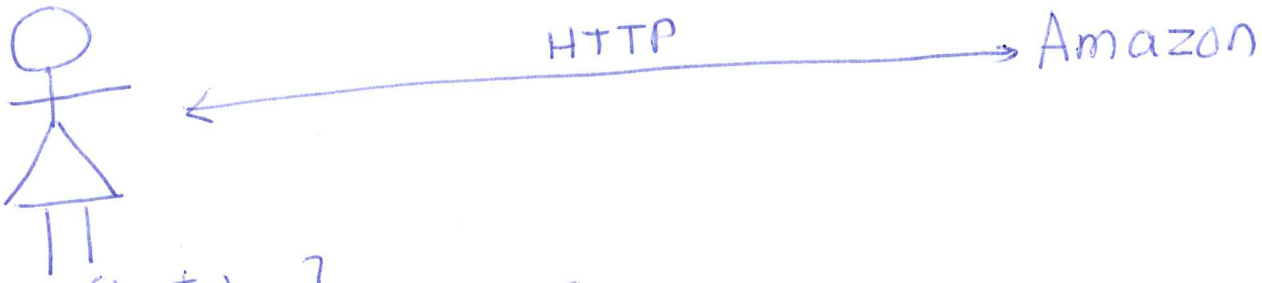
$$Pr[\text{Mac-forge}_{A,\Pi}(n) = 1] \leq negl(n)$$

Does the definition make sense?

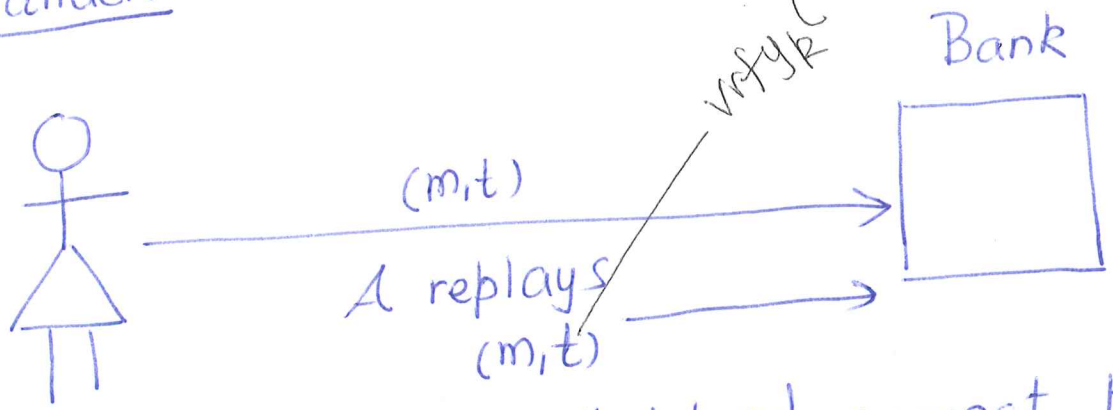


A forges (m, t)

$m \neq m_1, \ldots, m_k \neq m_k$



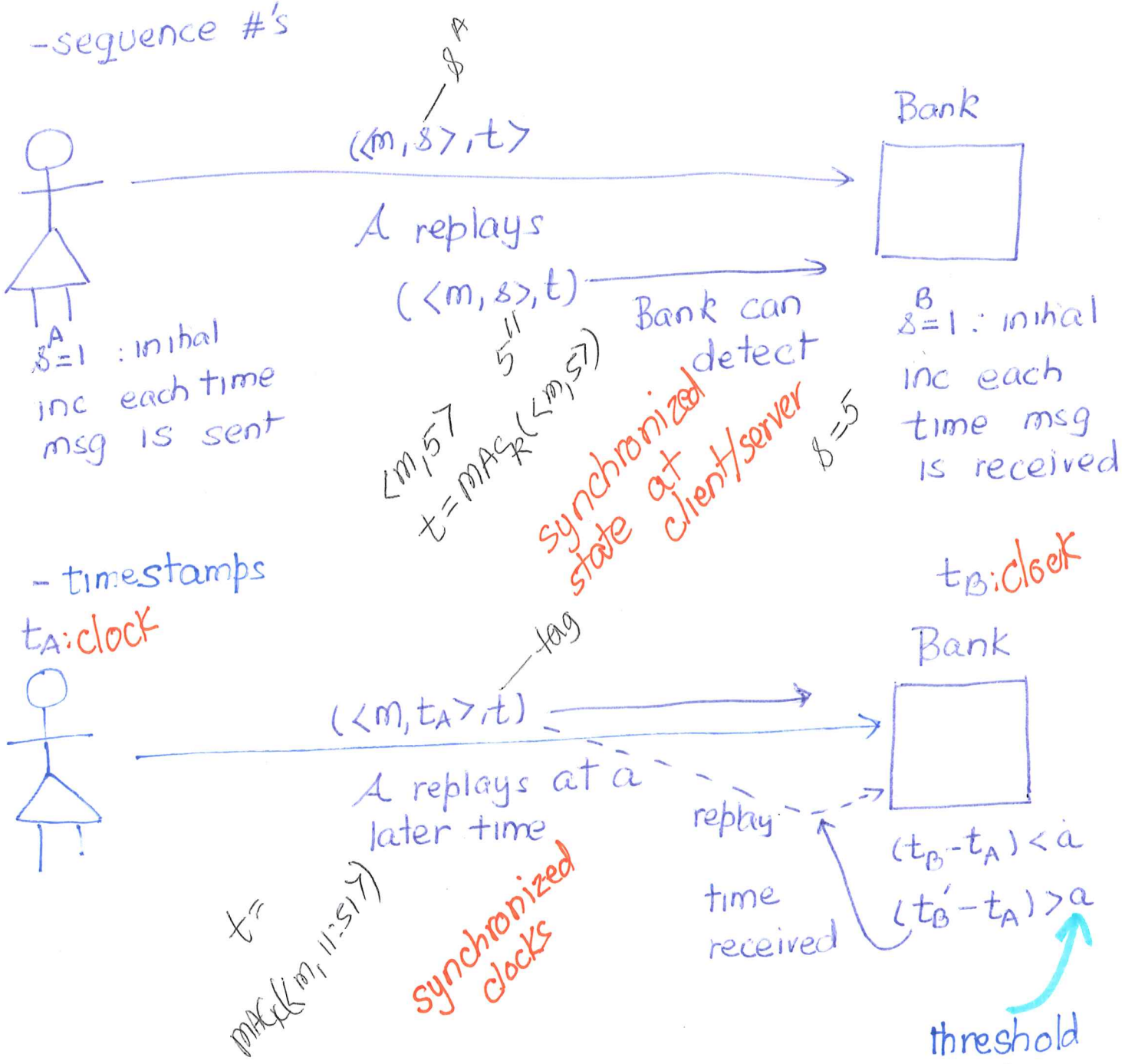
Replay attack



Macs don't defend against that

Two solutions

- sequence #'s



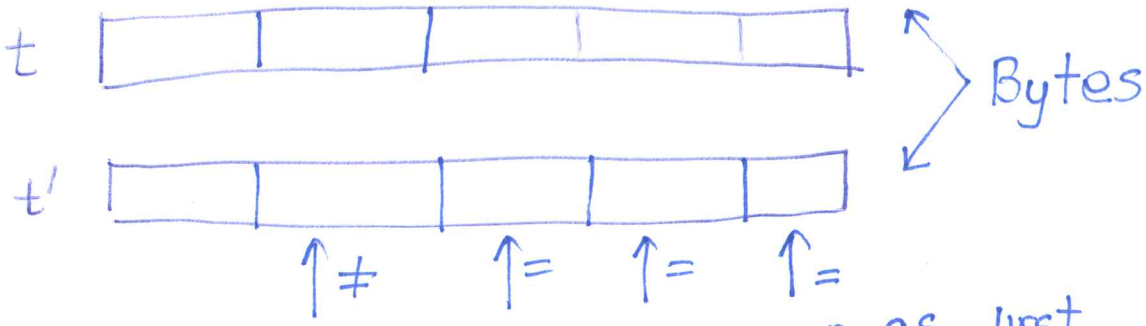
Canonical verification

$$Vrfy_k(m,t) = \begin{cases} t = t' & 1 \\ t \neq t' & 0 \end{cases}$$

$t' = Mac_k(m)$

comparison

Timing attacks.



↳ rejects as soon as first byte ≠

tag for m

(m, 0x00) → Vrfy procedure
(m, 0x01) → take byte which takes longest time
⋮
(m, 255)

↑ first byte, all other bytes 0

Say b_0 (we know first byte)

(m, ... 0 b_0)
(m, ... 1 b_0) → take byte which takes longest time
⋮
(m, ... ; 255 b_0)

↑ first byte b_0 ... $b_1 b_0$

Know second byte b_1

- Keep going - ...

Timing attack Xbox