Special Topics in Cryptography

Mohammad Mahmoody

Reminder

- Problem set 1 reminder due this Wed
- Need to solve problem 2.6 part b (from Katz-Lindell) book, as well.

Last time

- Defining encryption formally
- Information theoretic (perfect) vs. computational secrecy
- Limitation of perfect secrecy (and even its relaxations)

Today

- Secrecy based on (unproven) computational assumptions
- Pseudorandom generators (and functions)

Turing Machines

Computational Privacy/Security

running time

"probabilty of success"

A scheme is (t, ε)-secure *if every adversary running for time at most* t *succeeds in breaking the scheme with probability at most* ε.

- What it means to "break" depends on the exact security def.

$t(n) \leq n^c$ for some constant c

- Ideal: $t >$ $\boxed{\text{"feasible computation"}}$ and $\varepsilon(n) <$ $\boxed{\text{"negligible probability"}}$

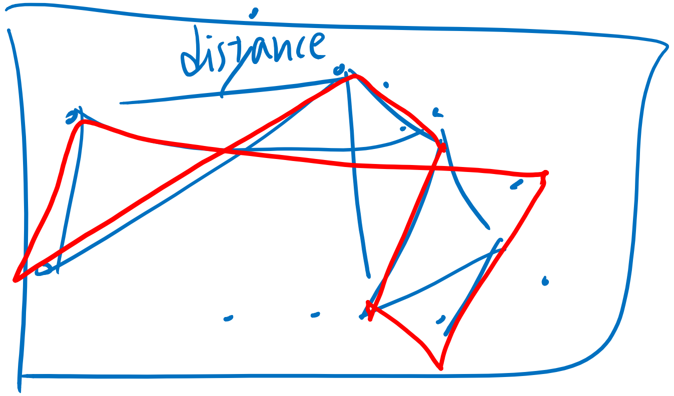
for all c, $\varepsilon(n) \leq \frac{1}{n^c}$, if n is large enough.

- Example: $t = 2^{100}$ $\varepsilon = 2^{-100}$ (age of universe $\approx 2^{80}$ seconds)

Examples ?

NP-Complete / hard : | Boolean SAT problem.

TSP:



inpt

weighted graph

find a Tour that goes to all nodes using minimal distance total.

quick feasible | run-time: time $\leq |\text{input size}|^{100}$ = $\underbrace{\text{time} \leq |\overbrace{\text{input size}}^{n}|^{\overbrace{100}^{O(1)\ \text{Some constant.}}}}_{\text{Poly-time.}}$

Conjecture: $\nexists$ $\underbrace{\text{poly-time}}_{\text{runs in time } n^{O(1)}}$ alg for TSP

run-Time $n \times n!$ easy.

2^n

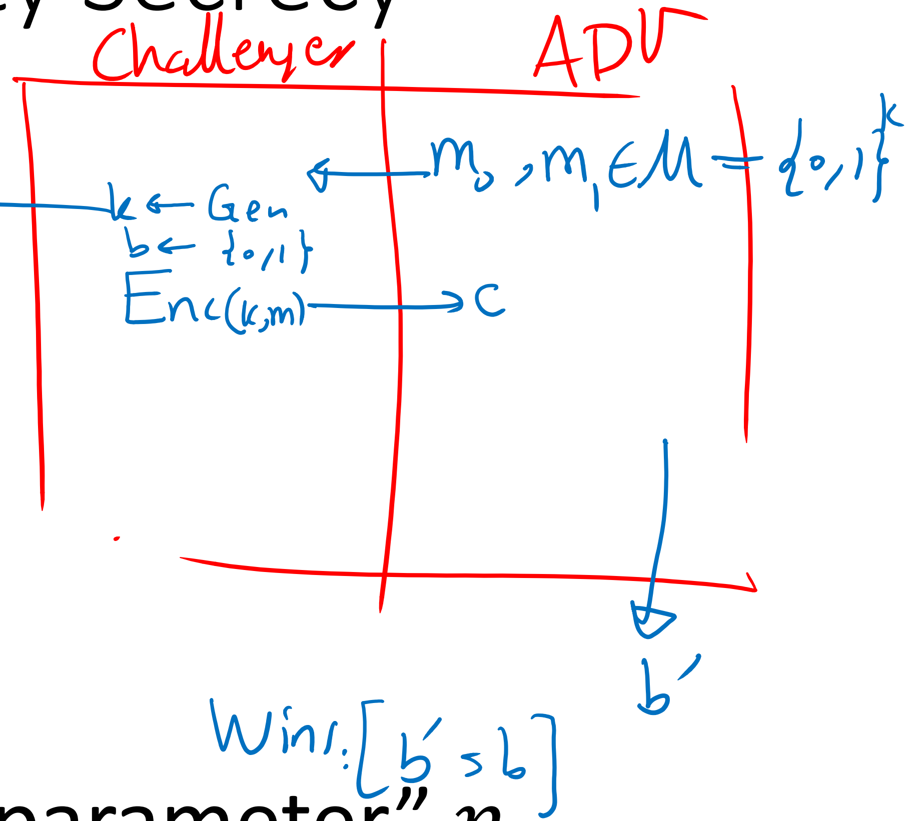
factoring large ~~Prime~~ numbers. (into primes) hope: for all $t(\cdot) = n^{\text{any constant}}$

Computational Indistinguishability Secrecy

- Eve (eavesdropping) security:
  Even if Eve knows $m \in \{m_0, m_1\}$
  she cannot **in time $t(n)$** guess m with
  probability $> \frac{1 + \varepsilon(n)}{2}$

$\varepsilon(n) < \frac{1}{\underbrace{\text{poly}(n)}_{\text{any}}}$

$n = |k|$



$t(n), \varepsilon(n)$ are functions of "security parameter" n
(e.g. key length $n = 1000$)

“Efficient” time and “Negligible” probability…

- Efficient: polynomial time over input length

$$t(n) \leqslant poly(n)$$

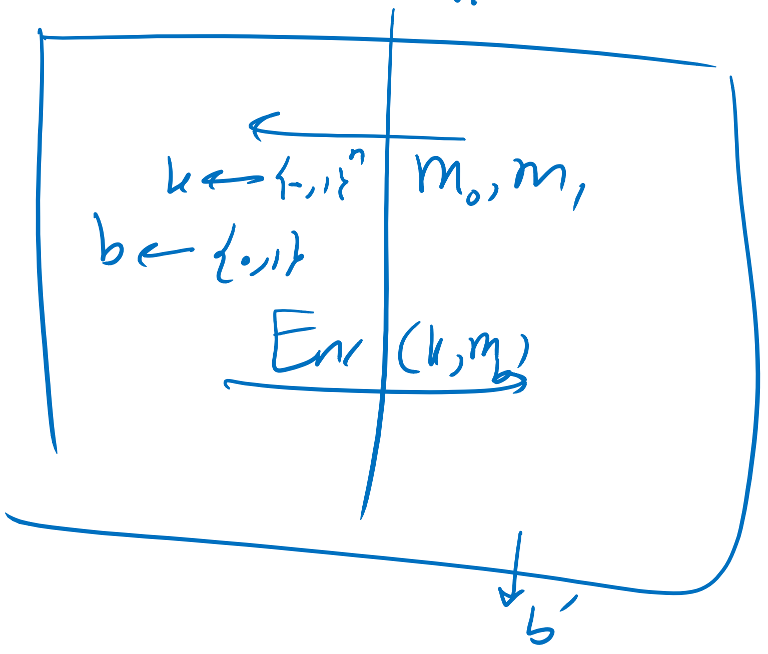
$$\exists c, \cancel{n} \quad t(n) \leqslant n^c \qquad 2^{-n}$$

- Negligible: smaller than any inverse polynomial (over input length)

$$\forall c \; \exists n_0 \quad \varepsilon(n) \leqslant \frac{1}{n^c} \quad \text{if } n \text{ is } \underbrace{\text{large enough.}}_{n > n_0}$$

Formal definitions of security

$n^{\lg(n)} > n^{100} > n^{1000}$



The adversarial indistinguishability experiment $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)$**:**

1. *The adversary* $\mathcal{A}$ *is given input* 1^n*, and outputs a pair of messages* m_0, m_1 *of the same length.*
2. *A random key* k *is generated by running* $\mathsf{Gen}(1^n)$*, and a random bit* $b \leftarrow \{0,1\}$ *is chosen. The ciphertext* $c \leftarrow \mathsf{Enc}_k(m_b)$ *is computed and given to* $\mathcal{A}$*.*
3. $\mathcal{A}$ *outputs a bit* b'*.*
4. *The output of the experiment is defined to be 1 if* $b' = b$*, and 0 otherwise. If* $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1$*, we say that* $\mathcal{A}$ succeeded.

2^n

DEFINITION 3.9 *A private-key encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has* indistinguishable encryptions in the presence of an eavesdropper *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$ *there exists a negligible function* negl *such that*

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

2^n

where the probability is taken over the random coins used by $\mathcal{A}$*, as well as the random coins used in the experiment (for choosing the key, the random bit* b*, and any random coins used in the encryption process).*

Two main issues:

- How to realize this definition?

- Is this the best definition addressing all issues?
  (No, it is still weak, but we will get back to this)

Pseudo-randomness
(random in eyes of computationally bounded)

$\underbrace{0\,1\,0\,1\,0\,1\,0\,1 \ldots 0\,1}_{100 \text{ times.}}$

2^{100}

word 1

$0\,1\,1\,1\,0\,1\,0\,1 \ldots$

more likely to be random!

2^{100}

$X = x_1 \text{——} x_n$

truly randomly generated

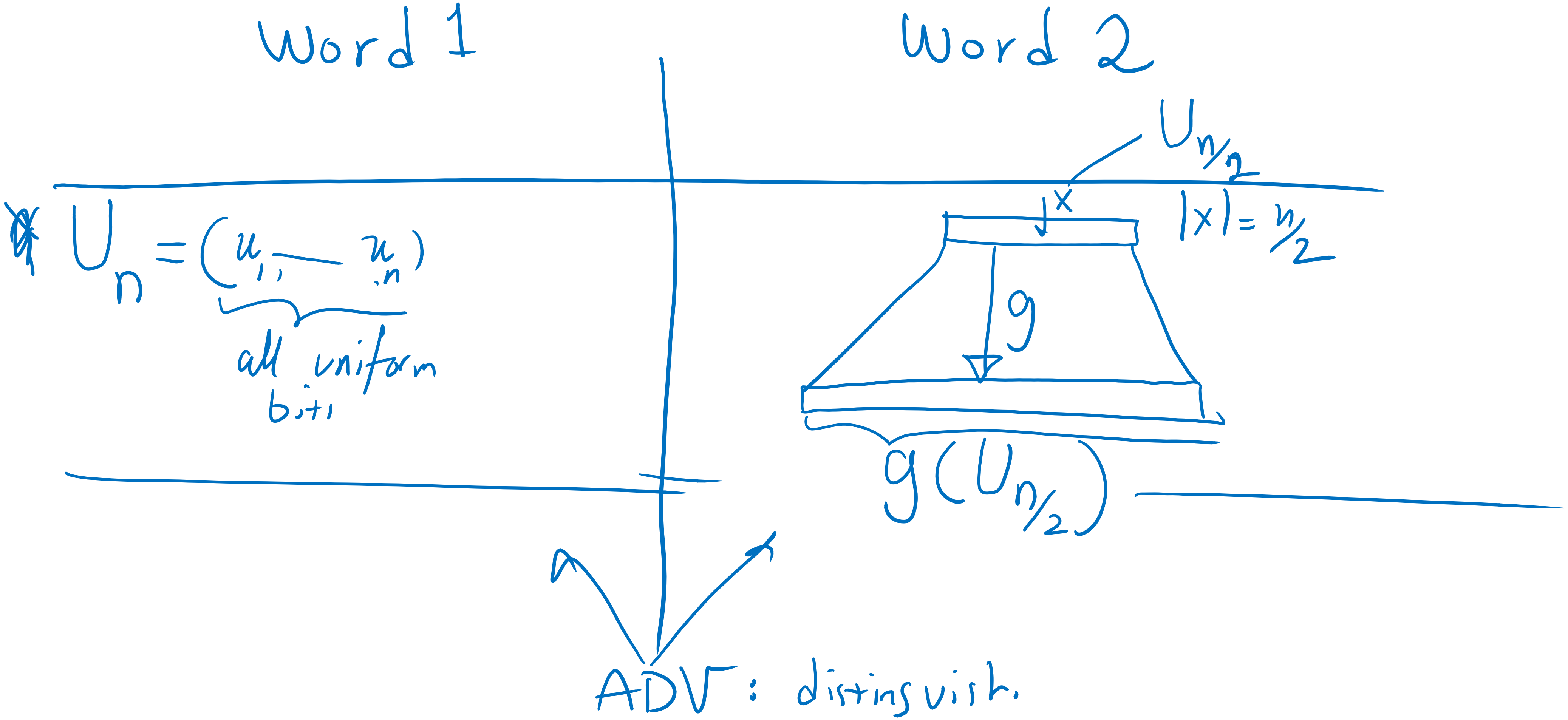
$Y = y_1 \text{——} y_n$; generated

"differently)

P

Pseudo-random generator (PRG)

- A magical tool that let us still do "one-time-pad" using short keys!



Formal definition of PRGs

efficient function $\underset{|x| = n/2}{g(x)} \longrightarrow \underbrace{\{0,1\}}_{\text{length} = n}{}^{2\cdot|x|}$

$\underbrace{|g(U_{n/2})|}_{\text{all possible outputs}} \ll 2^{n/2} \ll 2^n$

for all poly-time adv A:

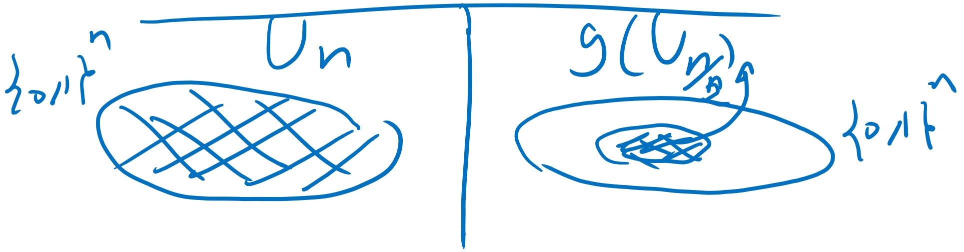
there is a negligible func $\varepsilon(\cdot)$

such that

$$\Pr[\text{Win}[\text{Adv}]] \le \frac{1}{2} + \varepsilon(n)$$

Chall.	n	ADV
$b \xleftarrow{\$} \{0,1\}$		
if $b=0$		
$y \leftarrow U_n$		
if $b=1$		
$y \leftarrow g(U_{n/2})$		
	$\xrightarrow{y}$	
	$\xleftarrow{b'}$	

win := $b = b'$



$|m| = n\text{-bit}$

Using PRGs → encrypting one long message

Compared with key

$\frac{n}{2}$ - bit

? assume the existence of a PRG $g : X \to Y \to$ n-bit

$|X| = \frac{n}{2}$-bit

Goal:

Construct an ind.-secure private-key ~~enc~~ encryption scheme (Enc, Dec, KGen(·))

Enc, Dec: work on messages of length n.

KGen(·): outputs key k at random $|k| = n/2$

B { $\mathrm{Enc}(k, m) : \underbrace{g(k)}_{k'} \oplus m$ ← bitwise

$\mathrm{Dec}(k, c) : g(k) \oplus m$

~~Enc~~

Thm: if g is a secure PRG → B is a secure SKE.

$\mathrm{Dec}(k, \mathrm{Enc}(k, m)) = m$

Completeness ✓

Equivalent definition of PRG
(applies to any ind. Security game – see PS1)

Guessing game

ADV

$b \leftarrow \{0,1\}$
if $b = 0$
$c \leftarrow X_0$
else
$c \leftarrow X_1$

c

b'

win if $b' = b$

$\Pr[\text{win}] \leq \frac{1+\varepsilon}{2}$

$\Longleftrightarrow$

World_0	World_1
$c \leftarrow X_0$	$c \leftarrow X_1$
$c \downarrow$ ADV	$c \downarrow$ ADV
$\downarrow b'$	$\downarrow b'$

$\Pr_{\text{world}_1}[\text{ADV}(c)=1] - \Pr_{\text{world}_2}[\text{ADV}(c)=1] \leq \varepsilon$

Proof of security for PRG → ind-secure encryption

Assuming g: $\underbrace{\text{secure}}_{(t,\varepsilon)}$ PRG

Goal: $B = (Enc_g, Dec_g)$ is a secure Enc.

~~(t,ε)~~ $(t/10, 10\varepsilon)$-secure

World 0	Hyb_0	Hyb_1	World 1
m_0, m_1 ADV	$m_0\ m_1$ ADV	$m_0\ m_1$ ADV	m_0, m_1 ADV
$Enc(m_0)$	$Enc'(m_0)$	$Enc'(m_1)$	$Enc(m_1)$
	OTP! $key = U_n$	OTP $key = U_n$	
$Pr[ADV \to 1] = P_0$	P'_0	P'_1	$Pr[ADV \to 1] = P_1$

ε $=$ ε

View of ADV is the same!

ε_0

$P'_0 = P'_1$

$\boxed{\text{Goal}\quad P_1 - P_0 < 2\varepsilon}$

World 0	Hyb 0
$\xleftarrow{m_0, m_1}$ ADV	$\xleftarrow{m_0, m_1}$ ADV
$\xrightarrow{m_0 \oplus g(U_{n/2})}$	$\xrightarrow{m_0 \oplus U_n}$
$\Pr\{out=1\} = p_0$	$\Pr[out=1] = p_0'$

claim $p_0' - p_0 \leq \varepsilon$

easy: $\Pr[A_2^2 \text{ output } 1 \text{ on } U_n] = p_0'$

$\Pr[A_2 \text{ output } 1 \text{ on } g(U_{n/2})] = p_0$

Proof: Suppose this is NOT the case!

Suppose A_1 is poly time t such that

$p_0' - p_0 > \varepsilon$

We construct ADversary A_2 that runs in poly Time and break security of PRG by ε!

A_2 is give y

$y \oplus m_0 \rightarrow y'$ calls $A(y')$ and outputs $A(y')$.