Cryptography

Lecture 1

Arpita Patra

Welcome to the second Half of the Course ☺

Course Homepage: http://drona.csa.iisc.ernet.in/~arpita/Cryptography15.html

Katz & Lindell: Introduction to Modern Cryptography, 2nd Edition, CRC Press.

Buy one, a great book.

Evaluation Policy (2nd Half)

Use it for a better cause than grading:

Enhance our learning process

Nurture qualities of a good cryptographer

Four Pillars of Evaluation

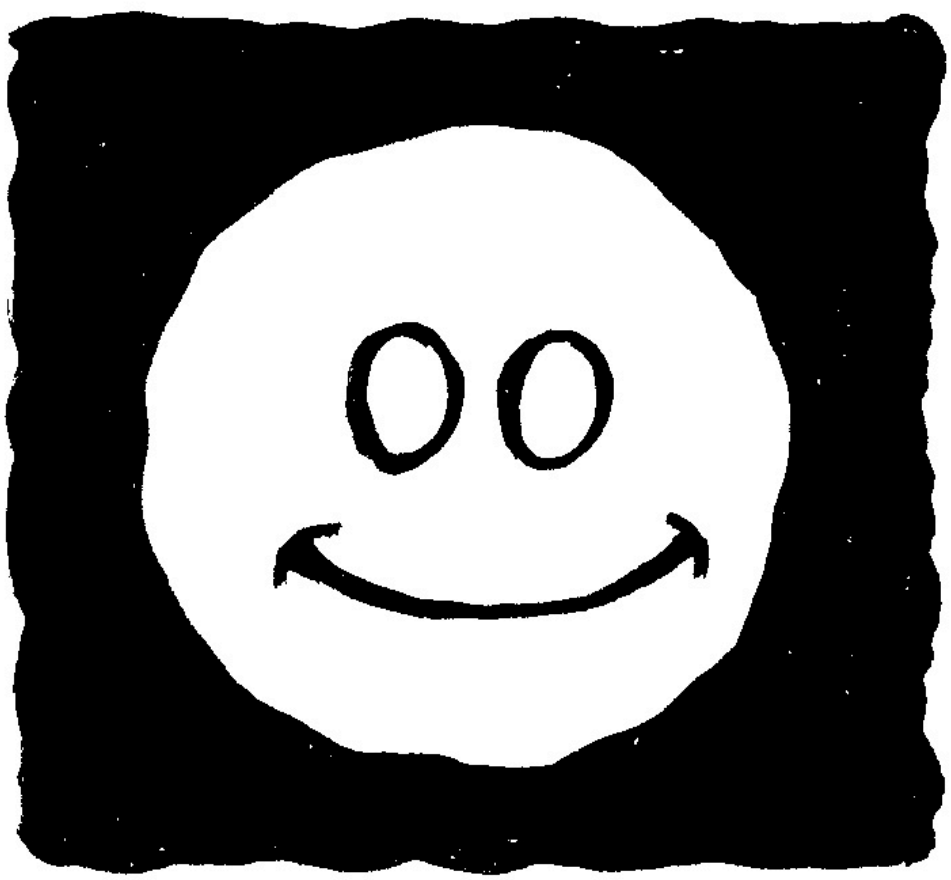
	Scribe (5%)	Chalk & Talk (10%)	Assignments (20%)	Final Exam (15%)
Depth			✔	✔
Breadth		✔		
Presentation- Oral		✔	✔	
Presentation- Written	✔	✔	✔	✔
Class Attentiveness	✔			
	2 students / lecture	2 sessions / week: Friday 8-9:30 am	1 evaluation / three weeks	1 evaluation

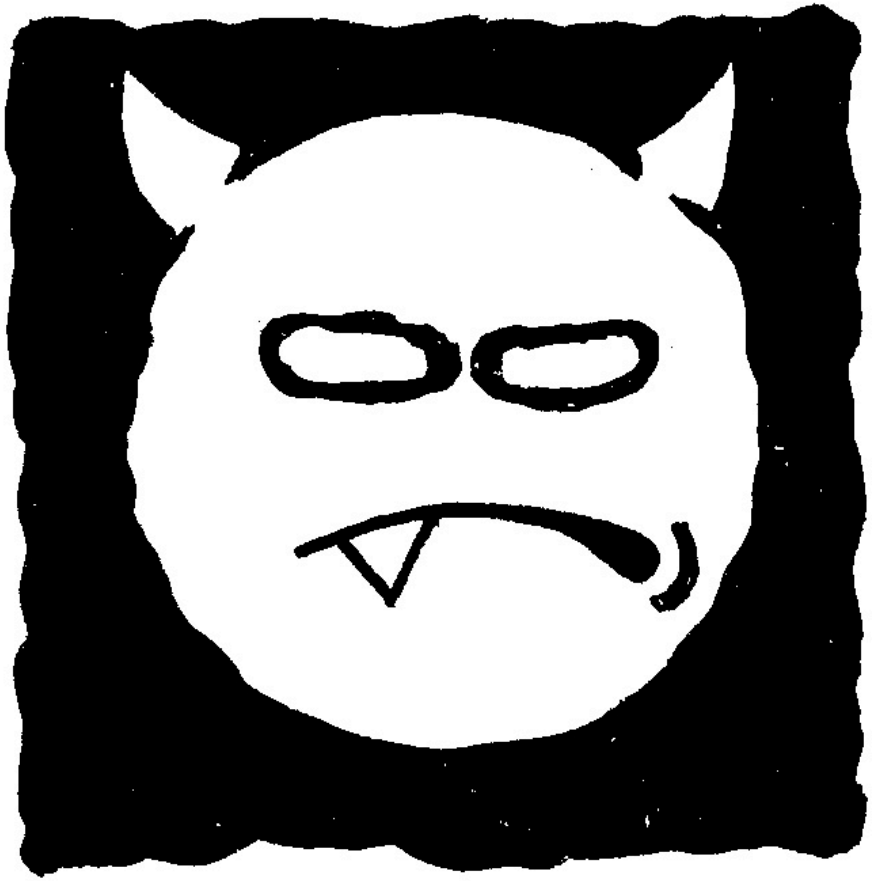
Let's start the exciting Journey of Cryptography

Well, you have already started.

Let's see where we stand so far and where we are heading to

Why Cryptography?





Tool to control access to information; tool to enforce policies who can learn/ influence information

Ethical Cryptographer: Protect Good from Bad

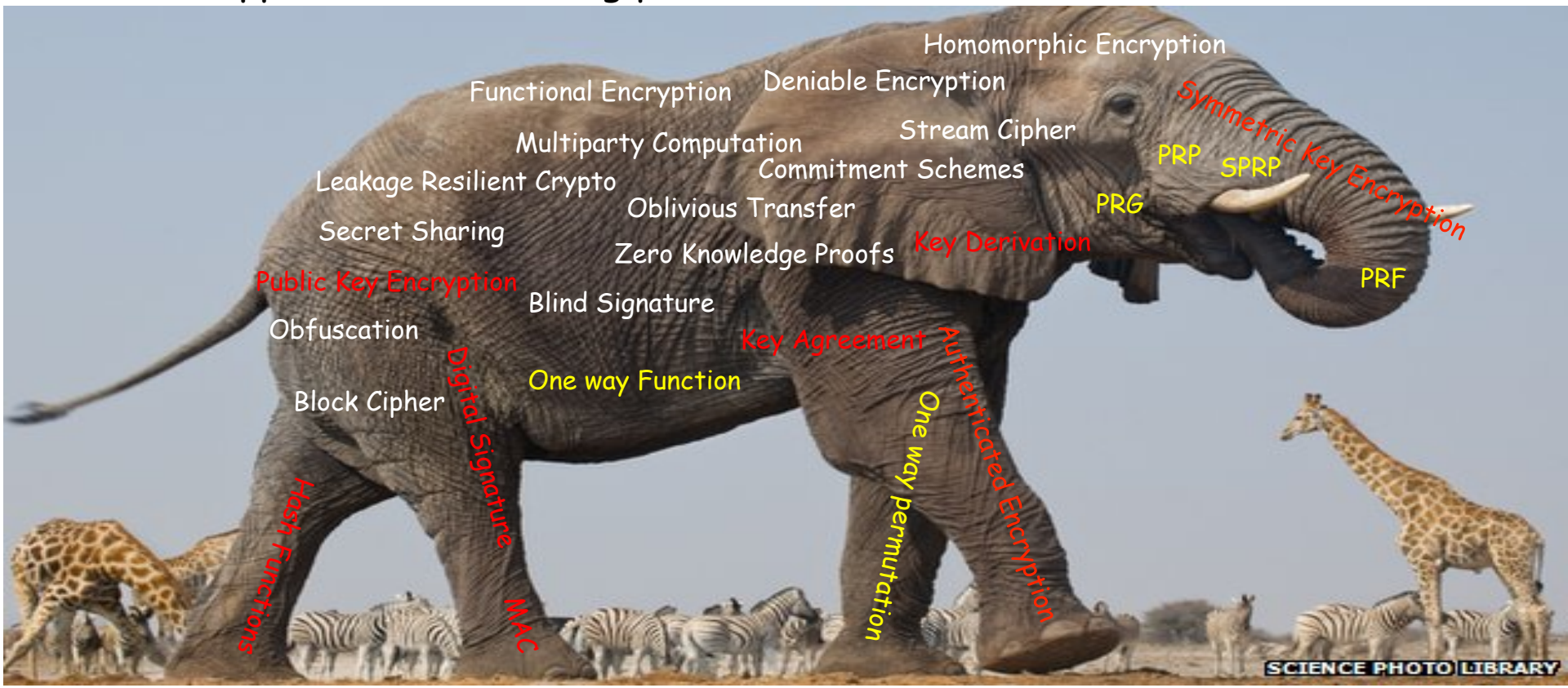
Crypto: Past and Present

Until 1970s	After 1970s and Until Now
Message Communication	Authenticated Message Communication
	E-cash
	Activism with Safety
	Secure Storage, Disk encryption
	Secure Information Retrieval
	Secure outsourcing to Cloud
	…….and the list goes on
	All these fall under the purview of Cryptography

Cryptography: Study of mathematical techniques for securing digital information, systems and distributed computation against adversarial attacks

Crypto Zoo

Real life Applications: Including previous ones



Assumptions: Number Theoretic/crude assumptions like AES/DES are secure PRF schemes, SHA 3 secure hash function etc.

Slide Idea courtesy: Manoj Prabhakaran

Impagliazzo's Crypto World: Crytomania & Minicrypt

Minicrypt assumes One way Function (OWF) exists and encompasses all that can be built from OWF: Symmetric Key Cryptography, Signatures, Commitments….

Cryptomania assumes Oblivious Transfer (OT) exists and encompasses all that can be built from OT: Almost everything that you can imagine in crypto

A big chunk of Cryptomania will be covered in my next course CSA E0 312 "Secure Computation" – Aug-Dec'15

Three Key Principles of Modern Crypto

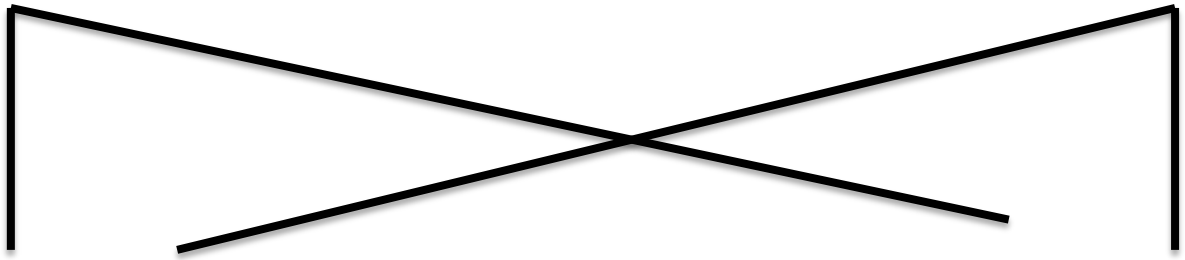
Perfect security

- Principle 1:
  - Formulation of rigorous and precise definition of security capturing requirement. Just having intuition is not enough.
- Principle 2:
  - Rely on well-studied assumption.
- Principle 3:
  - Construct scheme with rigorous proof of security

Our lookout in this course

Secure Message Communication

Message Authentication/Integrity



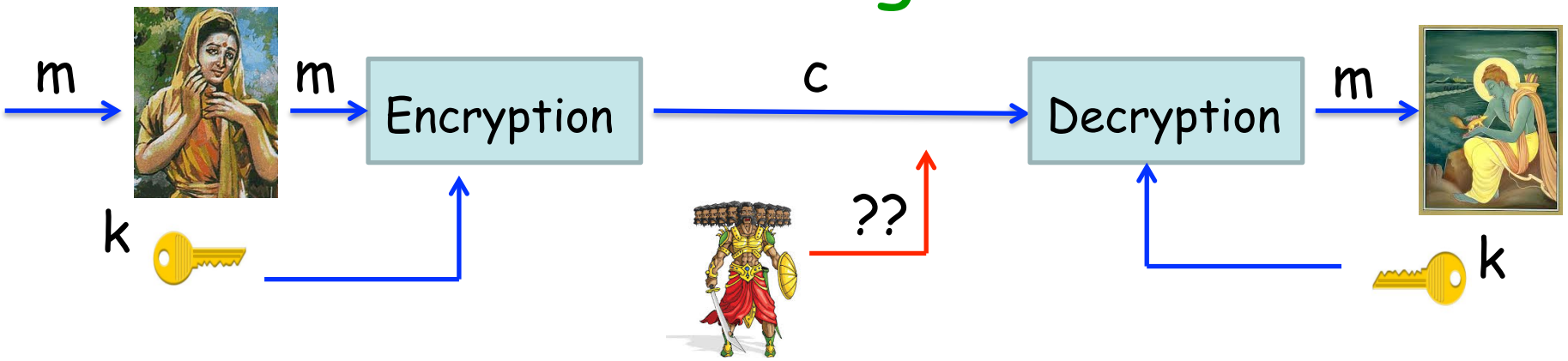
Symmetric/Private/
Shared Key Setting

Asymmetric/Public
Key Setting

Hash Function – How to compress a message
Key Agreement – How to agree on a key

>> How to construct a formal definition
>> Various Formal Security Definitions
>> Constructions and Security Proof

Secure Communication in Private Key Setting



- Secret key k shared in advance (by "some" mechanism)
- m is the plain-text
- c is the cipher-text (scrambled message)
- Symmetry: same key used for encryption and decryption

Syntax of SKE

1. Key-generation Algorithm (Gen()):
   - Outputs a key k chosen according to some probability distribution determined by the scheme;
   - MUST be a Randomized algorithm

2. Encryption Algorithm ($Enc_k(m)$); m from {0,1}*:
   - $c \leftarrow Enc_k(m)$ when randomized and $c := Enc_k(m)$ when deterministic
   - Deterministic/Randomized algorithm

3. Decryption Algorithm ($Dec_k(c)$):
   - Outputs $m := Dec_k(c)$
   - Usually deterministic

Syntax of SKE

- Any cipher defines the following three space (sets):
  1. Key space ($\mathcal{K}$):
     - Set of all possible keys output by algorithm Gen
     - Usually Gen selects a key k uniformly at random from $\mathcal{K}$
  2. Plain-text (message) space ($\mathcal{M}$):
     - Set of all possible "legal" message (i.e. those supported by Enc)
  3. Cipher-text space ($\mathcal{C}$):
     - Set of all cipher-texts output by algorithm Enc
     - The sets $\mathcal{K}$ and $\mathcal{M}$ together define the set $\mathcal{C}$
- Any cipher is defined by specifying (Gen, Enc, Dec) and $\mathcal{M}$

Towards Defining Security of SKE

Two components of a security definition:

Threat:
- >> Who is your threat?
- >> Who do you want to protect from?
- >> Cultivate your enemy a.k.a adversary in crypto language.
- >> Look out in practical scenarios / be an adversary
- >> Unless you know your adv, no hope of defeating him

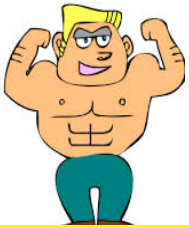
Break:
- >> What are you afraid of losing?
- >> What do you want to protect?
- >> If you don't know what to protect then how to do you when or if you are protecting it?
- >> Means different for different task

Threat Model

How powerful computationally?

What are his capabilities?

Unbounded Powerful



$Pr[M = m \mid C = c] = Pr[M = m] \quad \forall m, c$

Posteriori probability that m is encrypted in c

a priori probability that m will be sent

Key as large as the message

Fresh key for every encryption

Computationally Bounded

Two relaxations:

- >> Bounded Adversary
- >> Probability of sucess

A small key will do

Key reuse is permitted.

Computational Security of SKE

Computational security --- two relaxations to perfect-security

1. Bounded resource/ efficient adversaries
2. Adversaries can break the scheme with some very small probability, which is so small that we do not bother

Is it necessary to incorporate these two relaxations ?

YES Absolutely!
Will see towards the end of this class

Making "Efficient" and "Very Small" Precise- Asymptotic Approach

>> Security parameter n --- publicly known (part of the scheme)

Ru

Our SKE cannot be broken with probability better than 2^{-n} by adversary with n^{10} running time.

1. "Feasible" /"Effi Poly time (PPT)" means :

   - running tim constant c

Ex: 2^{-n}
Recall the properties of negl functions

2. " Very Small" / "negligible" means those f(n) :

   - for every constant c, there exists some N, such that $f(n) < n^{-c}$, for all $n > N$
   - "eventually becomes 0", "grows slower than any inverse poly"

Choosing n Carefully is Very Essential

A designer claims that an adversary running for n^3 minutes can break his scheme with probability $2^{40}\ 2^{-n}$

- $2^{40}\ 2^{-n}$ is negligible --- hence secure scheme
  - But what value of n to select while implementing ?
- If $n \leq 40$ then an adversary working for 40^3 minutes (6 weeks) can break the scheme with probability 1 --- completely useless
- n = 50 ?: adversary working for 50^3 minutes (3 months) succeed with probability 1/1000 --- may be unacceptable
- n = 500: adversary working for 200 yrs can break the scheme with probability 2^{-460} --- definitely acceptable

n = Knob

User's running time is also increasing ☹

Adv's job becomes harder ☺

min → max

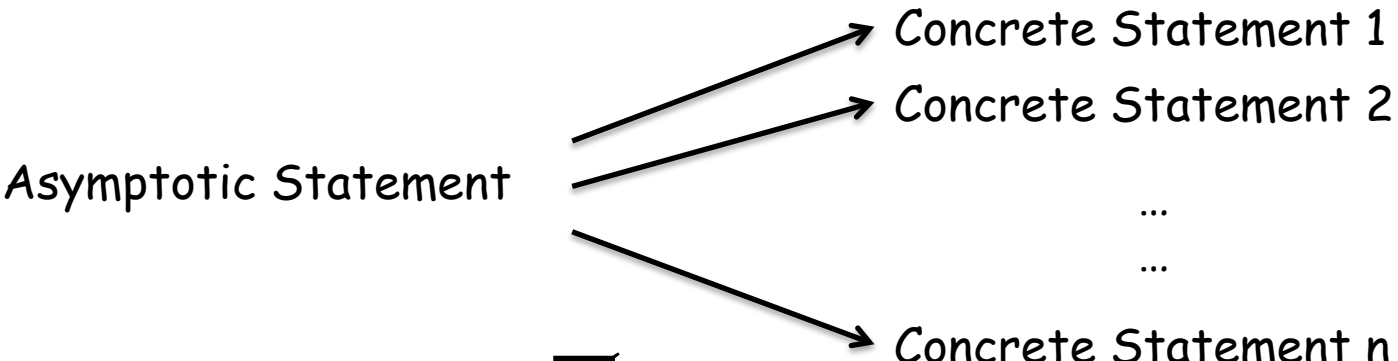
n

Concrete Approach

>> Set the value of n

>> Run users and adversary on specific machines

No adversary running for 5 yrs on 4GHz Machine can break the scheme with probability better than 2^{-60}



Asymptotic Approach ☑

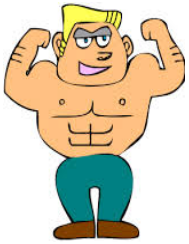
Concrete Approach ☒

Threat Model

How powerful computationally?

What are his capabilities?

Unbounded Powerful



Perfect/Shannon Security

Key as large as the message

Fresh key for every encryption

Computationally Bounded

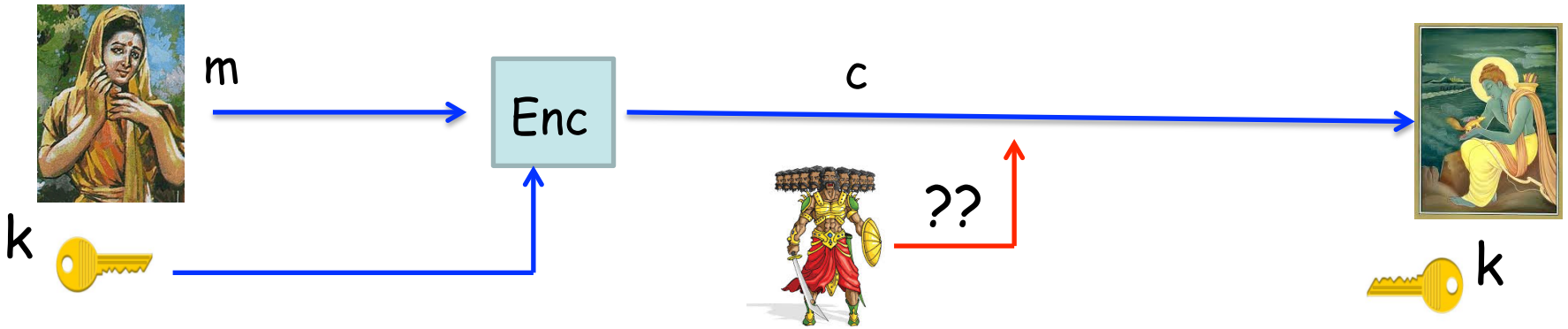
Computational Security

A small key will do

Key reuse is permitted.

Threat Model

Weakest: Ciphertext-only Attack



If you are thinking this is the best attack that can be mounted. Time for you to retire. Adversary is way more smarter ☺

- Attacker has acce her-texts)
- Goal: to determine xt(s)
- Passive/Eavesdropper in nature
  - Easy to mount

Threat Model

- Bounded Computing Power; Negligible error probability
- Ciphertext-only Attack
- Randomized Adversary

Randomness is absolute necessity in Crypto; it is practical and Good guys use randomness often. Why not adversary?

Good to be liberal in terms of giving more power to adversary

What constitutes your "Break"?

Break: >> What are you afraid of losing?
>> What do you want to protect?
>> If you don't know what to protect then how to do you when or if you are protecting it?
>> Means different for different crypto task

A1>> Secret key ?

Then Enc(m) = m is secure

A2>> Entire Message?

Then Enc(m) leaking msb is secure m is your salary

A3>> No additional info about the message irrespective of prior information?

Right Notion

Semantic Security Notion for SKE

S. Goldwasser and S. Micali. Probabilistic Encryption. Journal of Computer and System Sciences, 28(2): 270-299, 1984

Captures the ciphertext does not give any additional information about the message to the adversary irrespective of prior knowledge about m.

h(m): external info about m; history function

f(m): additional partial information about m that adv wants to compute

Syntax of SKE

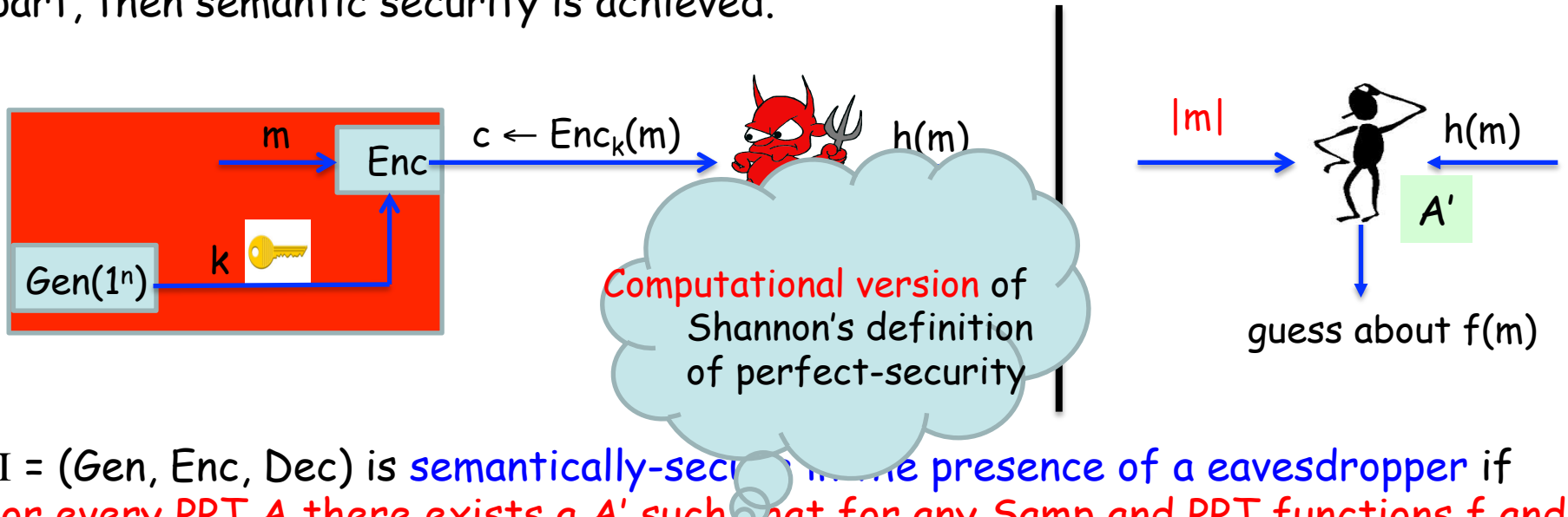
1. Key-generation Algorithm ($Gen(1^n)$):
   - MUST be a Randomized algorithm
   - Outputs a key k chosen according to some probability distribution determined by the scheme; |k| >= n
2. Encryption Algorithm ($Enc_k(m)$): m from {0,1}*:
   - Deterministic/Randomized algorithm
   - c ← $Enc_k(m)$ when randomized and c:=$Enc_k(m)$ when deterministic
3. Decryption Algorithm ($Dec_k(c)$):
   - Usually deterministic
   - Outputs m:= $Dec_k(c)$

If (Gen, Enc, Dec) is defined over message space $\{0,1\}^{l(n)}$ then fixed-length SKE

For message of length l(n)

Semantic Security

Two worlds: In one adv gets ciphertext and in another it does not. If the difference between probabilities of guessing f(x) in the both worlds are negligibly apart, then semantic security is achieved.



Π = (Gen, Enc, Dec) is semantically-secure in the presence of a eavesdropper if for every PPT A there exists a A' such that for any Samp and PPT functions f and h:

$$\left| \Pr[A(1^n, c, h(m)) = f(m)] - \Pr[A'(1^n, |m|, h(m)) = f(m)] \right| \leq \text{negl}(n)$$

Probability taken over
>> uniform k,
>> m output by Samp(1^n),
>> the randomness of A and
>> the randomness of Enc

Probability taken over
>> m output by Samp(1^n) and
>> the randomness of A'

Indistinguishability Based Definition

Common Feature: Experiment- a game between a challenger and an adversary

Eavesdropping indistinguishability experiment $\text{PrivK}^{co}_{A,\Pi}(n)$ $\Pi = (\text{Gen}, \text{Enc}, \text{Dec}), \mathcal{M}$

Attacker A

I can break Π

Run time: Poly(n)

$m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$

(freedom to choose any pair)

$c \leftarrow \text{Enc}_k(m_b)$

$b' \in \{0, 1\}$

(Attacker's guess about encrypted message)

Challenger

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

Let me verify

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

$b = b'$ $\quad \text{PrivK}^{co}_{A,\Pi}(n) \quad$ $b \neq b'$

1 --- attacker won

0 --- attacker lost

All Security Definitions will be in Ind style

SEMANTIC Security ≈ IND Security

(First Chalk & Talk Topic)

Π has indist … is co-secure if for every PP…

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

… the randomness used by A and the challenger

Equivalent Formulation of Ind Definition

Attacker A

I can break Π

Run time: Poly(n)

Π = (Gen, Enc, Dec), $\mathcal{M}$, n

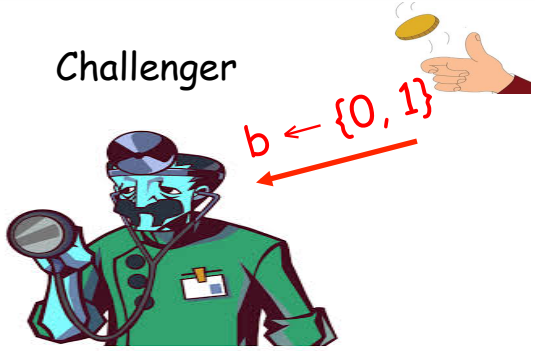
$m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$

(freedom to choose any pair)

$c \leftarrow Enc_k(m_b)$

$b' \in \{0, 1\}$

(Attacker's guess about encrypted message)



Challenger

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

Let me verify

$Gen(1^n)$

Game Output

$b = b'$

1 --- attacker won

$b \neq b'$

0 --- attacker lost

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

Intuition behind the definition ?

>> Attacker should behave in the same way irrespective of of m_0 or m_1

>> What does same behavior mean ? --- Attacker just outputs a bit

>> Same behavior means that attacker outputs 1 with al most the same probability in each case (irrespective of whether it sees an encryption of m_0 or m_1)

Equivalent Formulation

$\text{PrivK}^{co}_{A,\Pi}(n, b)$: the eavesdropping experiment with m_b selected by challenger

$\text{Output}(\text{PrivK}^{co}_{A,\Pi}(n, b))$: output bit of the attacker during $\text{PrivK}^{co}_{A,\Pi}(n, b))$

>> How to formalize that attacker's strategy gives the same output in each case

Π = (Gen, Enc, Dec) is CO-secure if for every PPT adversary A, there is a negligible function negl, such that :

$$\left| \Pr[\text{Output}(\text{PrivK}^{co}_{A,\Pi}(n, 0)) = 1] - \Pr[\text{Output}(\text{PrivK}^{co}_{A,\Pi}(n, 1)) = 1] \right| \leq \text{negl}(n)$$

Π = (Gen, Enc, Dec) is co-secure if for every PPT adversary A, there is a negligible function negl, such that :

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

Ingredient for co-secure SKEs

- OTP ? But key as large as message
- Need Better solution- Shorter key for big message
- Which symmetric-key primitive would serve our purpose?
- Pseudorandom Generators (PRGs)

M. Blum, S. Micali. How to Generate Cryptographically strong sequences of pseudo-random bits. SIAM Journal of Computing, 13(4), 850-864, 1984

A. C.-C. Yao. Theory and Applications of Trapdoor Functions. FOCS, 80-91, 1982.

Random Strings/Pseudorandom Strings

Pseudorandom string looks like a uniformly distributed string

"looking entity" runs in polynomial time

0100010101010100010101010

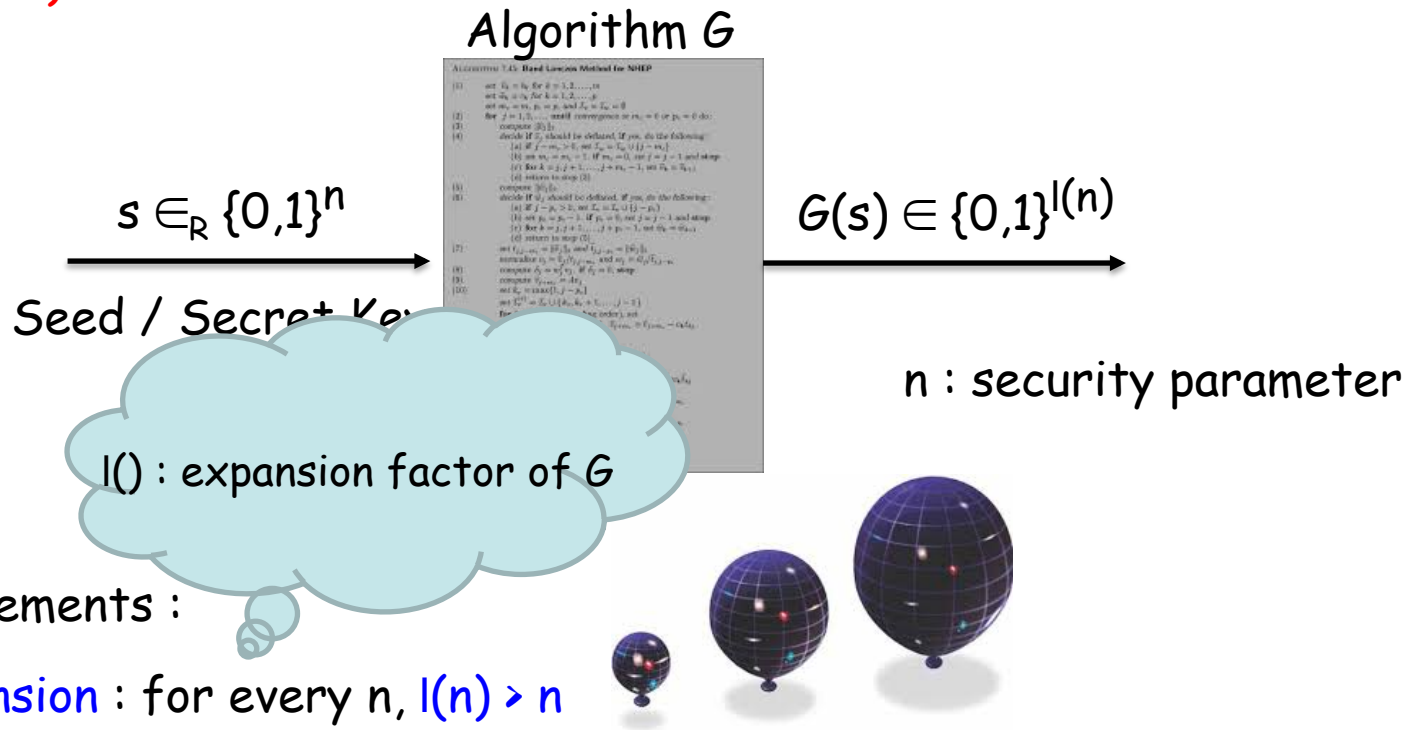
"pseudorandom" or "random" ?

Randomness/Pseudorandomness is a property of a distributions on strings

Random Generator / Pseudorandomgenerator is a property of a process/algorithm that generates strings according to the respective distributions.

Pseudorandom Generators (PRGs)

A deterministic algorithm that "expands" a truly random short string into a long pseudorandom string (that looks like a random string for a PPT Distinguishaer)



- Requirements :

  1. Expansion : for every n, $l(n) > n$

  2. Pseudorandomness : $G(s)$ "looks like" a truly random string

Mathematical formulation of pseudorandomness ? --- Indistinguishability game

PRG Security

U : uniform distribution over $\{0,1\}^{l(n)}$

PPT distinguisher D

A string of length l(n) please

y

How I selected it ?

Challenger

Oracle

A random string of length l(n) plz

$y \in_R \{0,1\}^{l(n)}$

b= 0

b= 1

$s \in_R \{0,1\}^n$

$y := G(s)$

G

G: Probability distribution over $\{G(s): s \in_R \{0,1\}^n\}$

G is a PRG if for every PPT D, there is a negligible function negl

$$\left| \Pr_{r \in_R \{0,1\}^{l(n)}}[D(r) = 1] - \Pr_{s \in_R \{0,1\}^n}[D(G(s)) = 1] \right| \leq negl(n)$$

Probability taken over
>> Random Choice of r
>> the randomness of D

Probability taken over
>> Random Choice of s
>> the randomness of D

Is designing PRG easy?

$s \in_R \{0,1\}^n$ →  → $G(s) = ss'$

$s' = s_1 \oplus s_2 \oplus \ldots \oplus s_n$

Expansion factor: n+1

- Is G a PRG? --- trivial to design a distinguisher D

$y \in \{0,1\}^{n+1}$ → random or generated by G ?

D

s the final bit of y XOR of the preceding bits ?

Yes → D outputs 1 — y generated by G

No → D outputs 0 — y random

- With how much probability D distinguishes a random and pseudorandom string ?
- If y generated by G
  - D outputs 1 with probability 1

  $\Pr [D(G(s)) = 1] = 1 \quad s \in_R \{0,1\}^n$
- If y is truly random
  - D outputs 1 with probability $\frac{1}{2}$

  $\Pr [D(r) = 1] = \frac{1}{2} \quad r \in_R \{0,1\}^{n+1}$

$$\left| \Pr [D(r) = 1] - \Pr [D(G(s)) = 1] \right| = \frac{1}{2}$$ Non-negligible

Existence of PRG

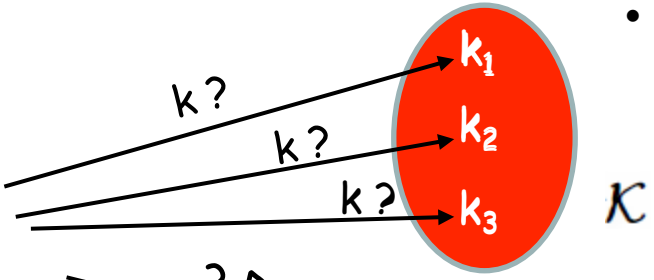
- Do PRG exists ?
- OWF + hardcore bit → PRG
  - Provably secure

- Several practical PRGs (Stream Ciphers)
  - Not provably secure (but no good distinguishers found till now)
  - High practical efficiency compared to provably-secure PRGs

Thank You!

Computational Security : Necessity of the Relaxations

- Practical crypto : many messages encrypted using a single short key
- Relaxation I : security only against efficient attackers --- Why ?
- Let $|\mathcal{K}| < |\mathcal{M}|$
- Let the attacker launch a known-plaintext attack against the scheme



$(m_1, c_1), (m_2, c_2), \ldots, (m_t, c_t)$: $c_i = Enc_k(m_i)$

? $Dec_{k_1}(c_i) = m_i$, for all i

? $Dec_{k_2}(c_i) = m_i$, for all i

? $Dec_{k_3}(c_i) = m_i$, for all i

Yes

- Attacker can try decrypting each cipher-text with all possible keys until it finds a matching key

---brute-force search

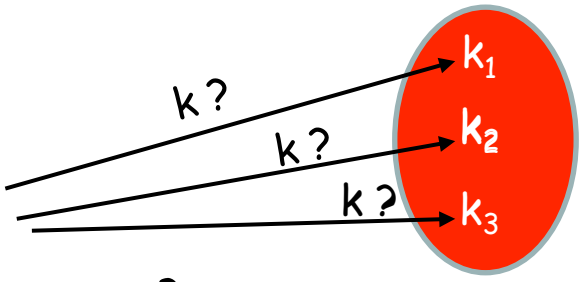
- $O(|\mathcal{K}|)$ time

Hurray : I got the key

- Need to bound the running time of the attacker to disallow it to carry brute-force search attack

Computational Security : Necessity of the Relaxations

- Practical crypto : many messages encrypted using a single short key
- Relaxation II : Attacker allowed to break the scheme with some very small probability --- Why ?
- Let $|\mathcal{K}| < |\mathcal{M}|$
- Attacker launches a known-plaintext attack



- Attacker randomly guess a key $k \in \mathcal{K}$ and checks whether it is the matching key --- O(1) time

$(m_1, c_1), (m_2, c_2), \ldots,$
(m_t, c_t): $c_i = Enc_k(m_i)$

? $Dec_{k_2}(c_i) = m_i$, for all i

Yes

- Need to allow a very small probability of success without considering it a break

Hurray : my guess was correct

- Probability : $1 / |\mathcal{K}|$