Cryptography
CS 555

Topic 4: Computational Security

Recap

- Perfect Secrecy, One-time-Pads

Theorem: If (Gen,Enc,Dec) is a perfectly secret encryption scheme then

$$|\mathcal{K}| \geq |\mathcal{M}|$$

What if we want to send a longer message?

K1,K2,K3

K1,K2,K3

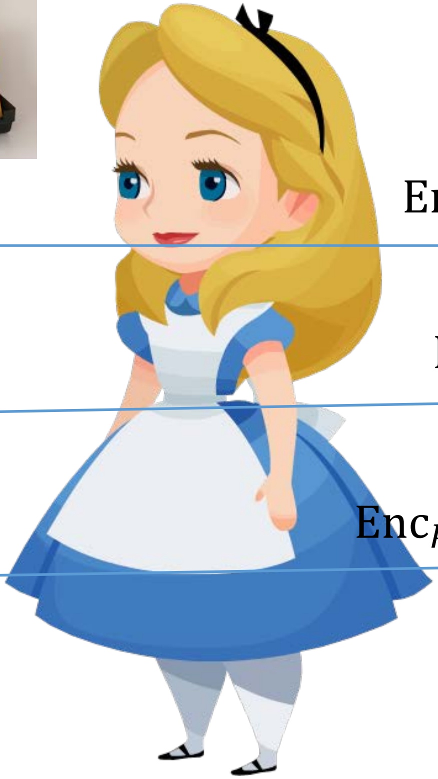
$\mathrm{Enc}_{k1}(\text{"}Dear\ Alice, I\ wrote\ this\ poem\ for\ you\text{"})$

$\mathrm{Enc}_{k2}(\text{"}Roses\ are\ red, \ldots.\text{"})$

$\mathrm{Enc}_{k3}(\text{"}I\ am\ out\ of\ space, but\ the\ rest\ was\ awesome\text{"})$

What if we want to send many messages?

K1,K2,K3



K1,K2,K3

$\mathrm{Enc}_{k1}(\text{"}Whats\ up, Alice?\text{"})$

$\mathrm{Enc}_{k2}(\text{"}Not\ too\ much, you?\text{"})$

$\mathrm{Enc}_{k3}(\text{"}Just\ chilling\ out?\text{"})$

Can we save their relationship?

K1,K2,K3

K1,K2,K3

$\text{Enc}_{k1}(\text{"}Whats\ up, Alice?\text{"})$

$\text{Enc}_{k2}(\text{"}Not\ too\ much, you?\text{"})$

$\text{Enc}_{k3}(\text{"}Just\ chilling\ out?\text{"})$

Perfect Secrecy vs Computational Security

- Perfect Secrecy is Information Theoretic
  - Guarantee is independent of attacker resources

- Computational Security
  - Security against computationally bounded attacker
    - An attacker with infinite resources might break security
  - Attacker might succeed with very small probability
    - Example: Lucky guess reveals secret key
    - Very Small Probability: $2^{-100}, 2^{-1000}, \ldots$

Today's Goal

- Define computational security in presence of eavesdropper who intercepts a single (long) message

If you don't understand what you want to achieve, how can you possibly know when (or if) you have achieved it?

- ~~Show how to build a symmetric encryption scheme with computational security in the presence of an eavesdropper.~~
- ~~Define computational security against an active attacker who might modify the message~~
- ~~Define computational security for multiple messages in presence of an eavesdropper~~

Concrete Security

"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"

- Example: $t = 2^{60}$ CPU cycles
  - 9 years on a 4GHz processor
  - < 1 minute on fastest supercomputer (in parallel)
- Full formal definition needs to specify "break"
- Important Metric in Practice
  - **Caveat 1**: difficult to provide/prove such precise statements
  - **Caveat 2**: hardware improves over time

Asymptotic Approach to Security

A scheme is secure if every *probabilistic polynomial time* (ppt) adversary "succeeds" with *negligible* probability.

- Two Key Concepts
  - Polynomial time algorithm
  - Negligible Function

Definition: A function $f\colon \mathbb{N} \longrightarrow \mathbb{R}_{\geq 0}$ is negligible if for every positive polynomial p there is an integer N>0 such that for all n > N we have

$$f(n) < \frac{1}{p(n)}$$

Asymptotic Approach to Security

Definition: A function $f\colon \mathbb{N} \longrightarrow \mathbb{R}_{\geq 0}$ is negligible if for every positive polynomial p there is an integer N>0 such that for all n > N we have

$$f(n) < \frac{1}{p(n)}$$

Intuition: If we choose the security parameter n to be sufficiently large then we can make the adversaries success probability very small (negligibly small).

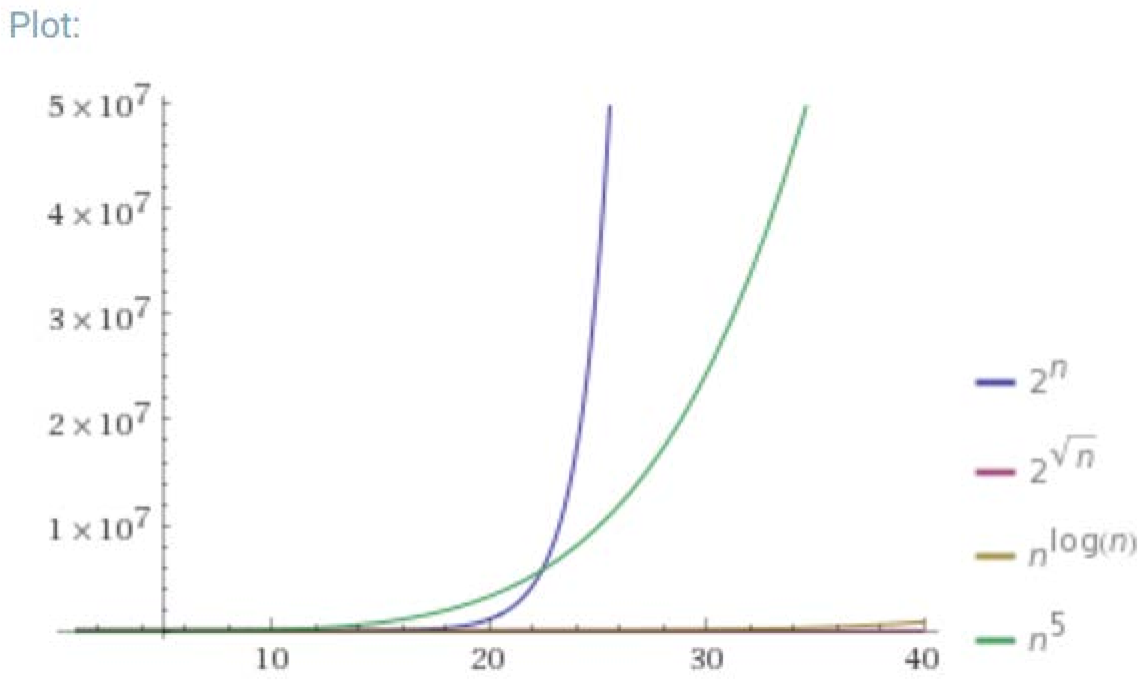
Asymptotic Approach to Security

Definition: A function $f\colon \mathbb{N} \longrightarrow \mathbb{R}_{\geq 0}$ is negligible if for every positive polynomial p there is an integer N>0 such that for all n > N we have

$$f(n) < \frac{1}{p(n)}$$

Which functions below are negligible?

- $f(n) = 2^{-n}$
- $f(n) = n^{-3}$
- $f(n) = 2^{-1000} 1000 n^{1000}$
- $f(n) = 2^{100} 2^{-\sqrt{n}}$
- $f(n) = 2^{-\log n}$



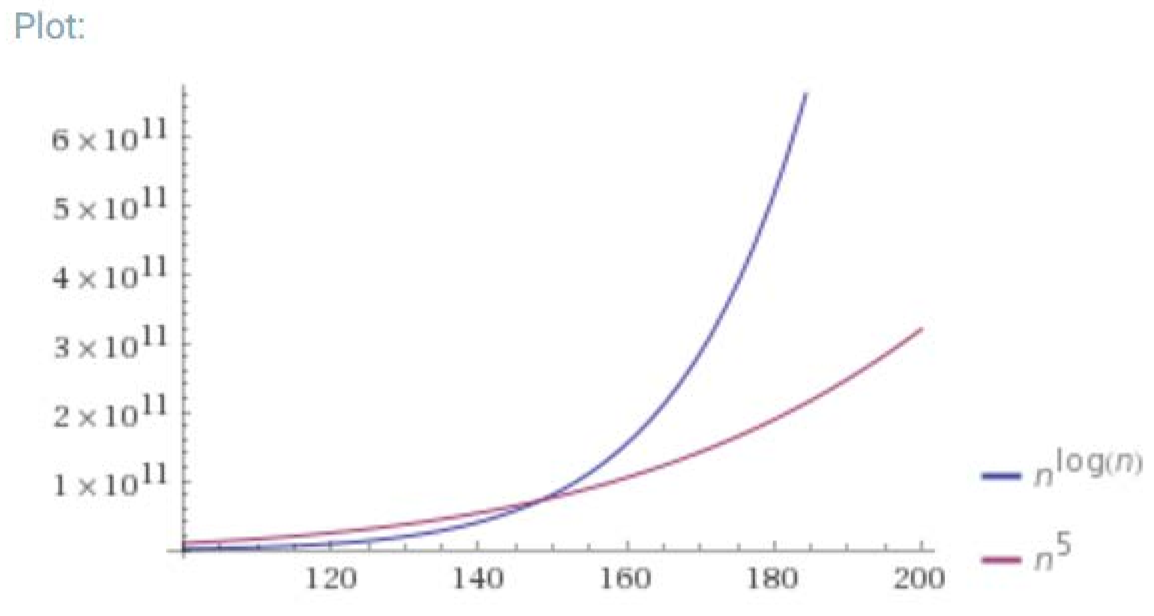
Asymptotic Approach to Security

Definition: A function $f\colon \mathbb{N} \longrightarrow \mathbb{R}_{\geq 0}$ is negligible if for every positive polynomial p there is an integer N>0 such that for all n > N we have

$$f(n) < \frac{1}{p(n)}$$

Which functions below are negligible?

- $f(n) = 2^{-n}$
- $f(n) = n^{-3}$
- $f(n) = 2^{-1000}1000n^{1000}$
- $f(n) = 2^{100}2^{-\sqrt{n}}$
- $f(n) = 2^{-\log n}$



Asymptotic Approach to Security

Definition: An (randomized) algorithm A runs in polynomial time if there exists a polynomial p such that for every n-bit input x, A(x) terminates in at most p(n) steps in expectation.

Intuition: If an algorithm A does not run in polynomial time then, for sufficiently large n, it will quickly become impractical for any attacker to run the algorithm A.

Asymptotic Approach to Security

A scheme is secure if every *probabilistic polynomial time* (ppt) adversary "succeeds" with *negligible* probability.

- **General Attack 1:** Test all possible secret keys $\mathrm{k}' \in \mathcal{K}$
  - Doesn't run in polynomial time, since $|\mathcal{K}| = 2^n$
- **General Attack 2:** Select random key $\mathrm{k}' \in \mathcal{K}$, check if it is correct (otherwise output $\perp$ for "fail").
  - Only successful with negligible probability 2^{-n}

Advantages of Asymptotic Approach

- **Closure**
  - If subroutine B runs in polynomial time and algorithm A makes p(n) queries to B then A also runs in polynomial time.
  - If f and g are negligible functions then h(n) = f(n)+g(n) is a negligible function
  - If p is a positive polynomial, and f is a negligible function then the function g(n)=f(n)p(n) is also negligible.
- **Church-Turing Thesis**: “reasonable” model of computations are all polynomially equivalent.
- **Implication**: No need to worry about different models of computation (circuits, random access machines, etc…)
- **Disadvantage:** Limited guidance on how big to make security parameter n in practice.

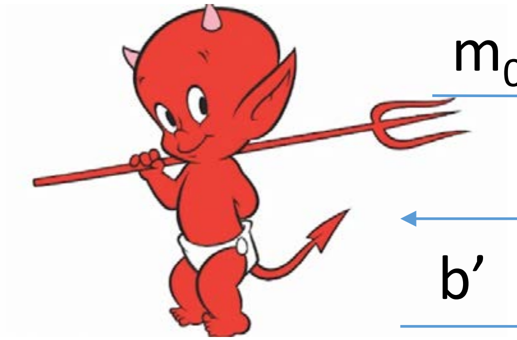
Private Key Encryption Syntax (Revisited)

- Message Space: $\mathcal{M}$
- Key Space: $\mathcal{K}$
- Three Algorithms
  - $\text{Gen}(\mathbf{1^n}; R)$ (Key-generation algorithm)
    - Input: **1^n (security parameter in unary)** + Random Bits
    - Output: Secret key $\mathrm{k} \in \mathcal{K}$
  - $\text{Enc}_{\mathrm{k}}(m; \boldsymbol{R})$ (Encryption algorithm)
    - Input: Secret key $\mathrm{k} \in \mathcal{K}$ and message $\mathrm{m} \in \mathcal{M}$ + Random
    - Output: ciphertext *c*
  - $\text{Dec}_{\mathrm{k}}(c)$ (Decryption algorithm)
    - Input: Secret key $\mathrm{k} \in \mathcal{K}$ and a ciphertex c
    - Output: a plaintext message $\mathrm{m} \in \mathcal{M}$ *or* $\perp$ ***(i.e "Fail")***

Requirement: all three algorithms run in probabilistic polynomial time

- Invariant: $\text{Dec}_k(\text{Enc}_k(m))=m$

Adversarial Indistinguishability Experiment



m_0, m_1 →

← **c**

b' →

Random bit b

K = Gen(.)

$c = Enc_K(m_b)$

ppt attacker

negligible function

$\forall$  $\Pr\Big[$  $Guesses\ b' = b\Big] \leq \frac{1}{2} + \mu(n)$

Adversarial Indistinguishability Experiment

Formally, let $\Pi = (Gen, Enc, Dec)$ *denote the encryption scheme, call the experiment* $PrivK^{eav}$ *and define a random variable*

$$PrivK_{A,\Pi}^{eav} = 1 \quad if\ b = b'$$
$$PrivK_{A,\Pi}^{eav} = 0 \quad otherwise$$

Π *has indistinguishable encryptions in the presence of an eavesdropper if for all PPT adversary* A*, there is a* Negligible function μ such that $\Pr[PrivK_{A,\Pi}^{eav} = 1] \leq \frac{1}{2} + \mu(n)$

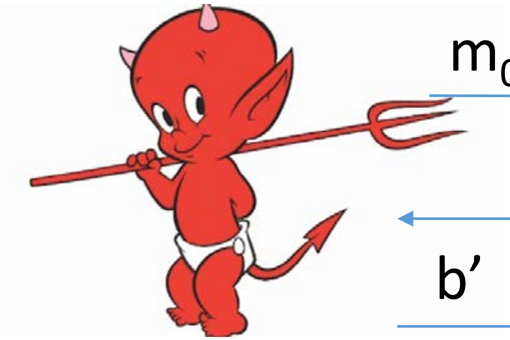
om bit b

en(.)

$c_K(m_b)$

 [] = 2 (

Semantic Security



m_0, m_1 →

← **c**

b' →

Random bit b

K = Gen(.)

$c = Enc_K(m_b)$

ppt attacker

negligible function

$\forall$  $\Pr\Big[$  $Guesses\ b' = b\Big] \leq \frac{1}{2} + \mu(n)$

Aside: Message and Ciphertext Length

- In the previous game we typically require that $|m_0|=|m_1|$. Why?

- It is <u>impossible</u> to support arbitrary length messages while hiding all information about plaintext length

- Limitation: When could message length be sensitive?
  - Numeric data (5 figure vs 6 figure salary)
  - Database Searches: number of records returned can reveal information about the query
  - Compressed Data: Short compressed string indicates that original plaintext has a lot of redundancy. (CRIME attack on session cookies in HTTPS)

Implications of Indistinguishability

Theorem 3.10: Let (Gen, Enc, Dec) be a fixed-length private key encryption scheme for message of length ℓ that satisfies indistinguishability (prior definition) then for all PPT attackers A and any $\mathrm{i} \leq \ell$ we have

$$\Pr\left[A\left(1^n, \mathrm{Enc}_K(m)\right) = m^i\right] \leq \frac{1}{2} + \mathrm{negl}(n)$$

Where the randomness is taken over $\mathrm{K} \leftarrow \mathrm{Gen}(1^n)$, <u>uniform</u> $\mathrm{m} \in \{0,1\}^\ell$ and the randomness of Enc and A.

h(m) background knowledge the attacker might have about m.

Definition 3.12: Let (Gen, En... fixed-length private key encryption scheme for message of le... antically secure if for all PPT attackers A t... t for any PPT algorithm Sample all any ... f and h we have

$$|\Pr[A(1^n, \mathrm{Enc}_K(\ldots), h(m)) = f(m)]$$

A... en get to see an encry... on ...! Just the length of ...!

Semantic Security

Definition 3.12: Let (Gen, Enc, Dec) be a fixed-length private key encryption scheme for message of length ℓ. We say that the scheme is semantically secure if for all PPT attackers A there exists a PPT algorithm A' such that for any PPT algorithm Sample all any polynomial time computable functions f and h we have

$$|\Pr[A(1^n, \mathrm{Enc}_K(m), h(m)) = f(m)] - \Pr[A'(1^n, |m|, h(m)) = f(m)]|$$
$$\leq \mathrm{negl}(n)$$

Where the randomness is taken over $\mathrm{K} \leftarrow \mathrm{Gen}(1^n)$, $\mathrm{m} \leftarrow \mathrm{Samp}(1^n)$ and the randomness of Enc, A and A'.

Theorem 3.13: Both security definitions (indistinguishable encryptions/semantic security) are equivalent.

Coming Up…

- Before Next Class (Friday)
  - Read: Katz and Lindell 3.3
  - Constructing Secure Encryption Schemes

- Homework 1 released Friday