Cryptographic Systems and Security

Encryption Protocols and Cryptographic Attacks

Ryder LiuLin

University of California, Berkeley

May 5, 2022

Sections

Preliminaries

Definition: Negligible: less than $\frac{1}{p(n)}$ for all polynomials $p(n)$

Definition: X and Y are computationally indistinguishable if an algorithm can only distinguish elements in X and Y with negligible probability

Definition: Something is **pseudorandom** if it is deterministic and computationally indistinguishable from a (uniform) distribution

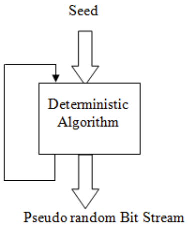


Figure: PRNG

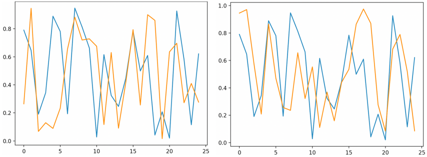


Figure: PRNG vs. True RNG

Preliminaries II

Definition: **Zero-knowledge proof**: system for demonstrating possession of knowledge without revealing knowledge

Usually, "knowledge" = solution to NP(-complete) problem

Figure: Water or vodka?

Preliminaries III

Terminology:

- Encryption scheme:
  - key generation algorithm $G(1^n) = (e, d)$
  - encryption function E
  - decryption function D, $D(d, E(e, x)) = x$
- **Private-key** encryption: $d = e =: k$
- **Public-key** encryption: $d \neq e$

Goal: Enable secure communication over insecure channel

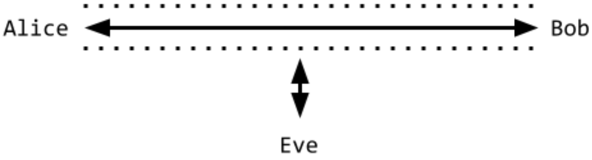


Figure: Three parties

Formalizing Security

<u>Definition:</u> (Single-message) **semantically security**: anything computable from the ciphertext is computable from the length of the plaintext

<u>Definition:</u> (Single-message) **indistinguishable encryptions**: any two ciphertexts are computationally indistinguishable

Theorem: An encryption scheme is semantically secure if and only if it has indistinguishable encryptions [Goldreich, 2009]

Security Against Interactions/Attacks

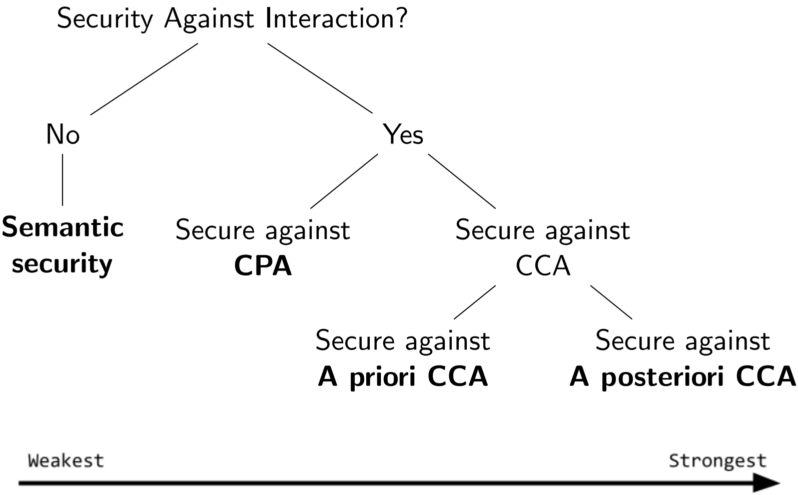
What if the attacker can interact with the encryption system?

Chosen Plaintext Attack (CPA): attacker can encrypt whatever it likes

Chosen Ciphertext Attack (CCA): attacker can decrypt (and encrypt) whatever it likes (that avoids triviality)

- **A priori CCA**: attacker must do decryption before being challenged
- **A posteriori CCA**: attacker may do decryption before and after being challenged

Security Against Interactions/Attacks



Constructions: Private-Key

Construction:

- Key: pseudorandom function f_k associated with random key k
- Encrypt: random r with $|r| = |x|$, $E_k(x) = (r, f_k(r) \bigoplus x) =: (r, y)$
- Decrypt: $D_k(r, y) = f_k(r) \bigoplus y$

Security: Secure up to a priori CCA

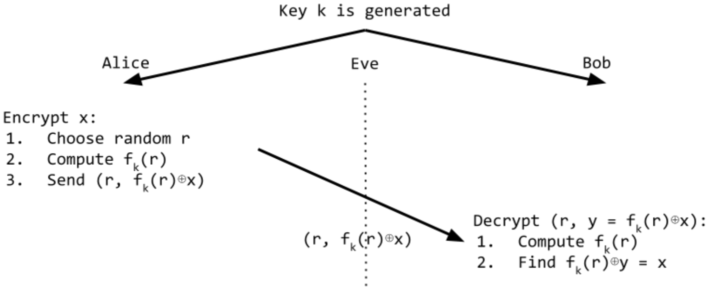


Figure: Private-key encryption system

Constructions: Public-Key

Construction:

- Key: trapdoor permutation p_α only invertible with trapdoor ${p^{-1}}_\tau$
- Encrypt: random $r \in Dom(p_\alpha)$,
  $E_\alpha(x) = (p_\alpha(r), x \bigoplus b_\alpha(r)) =: (y, \zeta)$, where b_α is a hardcore function of p_α
- Decrypt: $D_\tau(y, \zeta) = \zeta \bigoplus b_\alpha({p^{-1}}_\tau(y))$

Security: Secure up to CPA

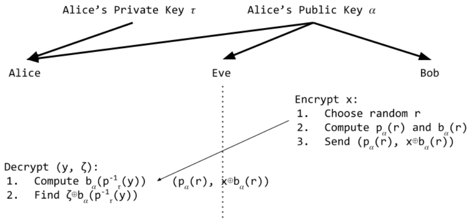


Figure: Public-key encryption system

Security Improvements

Idea: Make a posteriori CCA (almost) equivalent to CPA by making it infeasible to produce legitimate, useful ciphertext

How? Require private knowledge (either key or plaintext)

Private-key: Attach message authentication code (MAC) to ciphertext

- MAC is hard to forge (requires f_k to create)

Public-key: Attach non-interactive zero-knowledge proof (NIZK) to ciphertext

- NIZK is used to prove knowledge of plaintext
- Attacker can now only decrypt things it originally encrypted itself

References I

Goldreich, O. (2009).
Foundations of cryptography ii, basic applications.
Foundations of Cryptography, pages 373–469, Cambridge, United Kingdom. Cambridge University Press.