Homework 1

Cryptography & Network Security
CS 406 : Spring 2021

Released: Thu Jan 28
Due: Sun Feb 14

CPA-Secure Symmetric-Key Encryption [Total 100 pts]

1. **Secure Computation with Perfect Secrecy** [15 pts]

   This problem considers a puzzle from Lecture 0, involving three parties, Alice, Bob and Carol. Alice and Bob are given *inputs* $x, y \in D$ (for some finite domain D) and Carol wishes to learn $f(x, y)$ (for some function $f : D \times D \to Z$).

   We shall consider protocols that proceed as follows (specified in terms of a finite set R and three functions g_A, g_B, g_C). After Alice and Bob receive their inputs x and y respectively:

   - Alice picks $r \leftarrow R$ uniformly at random and sends r to Bob.
   - Alice sends a message $\alpha = g_A(x, r)$ to Carol and Bob sends $\beta = g_B(y, r)$ to Carol, where $g_A, g_B : D \times R \to Q$.
   - Carol outputs $z = g_C(\alpha, \beta)$, where $g_C : Q \times Q \to Z$.

   By the nature of the protocol, Alice and Bob learn nothing about each other's inputs (note that r is chosen independently of x).

   (a) State the perfect correctness requirement of the protocol formally, in terms of the sets D, R, and the functions f, g_A, g_B, g_C.

   (b) Formalize a perfect secrecy requirement that Carol learns nothing other than $f(x, y)$ in this protocol, by filling in the blanks below.

   $$\forall \underline{\hspace{6cm}} \quad \Pr_{r \leftarrow R}\left[\underline{\hspace{6cm}}\right] = \Pr_{r \leftarrow R}\left[\underline{\hspace{6cm}}\right]$$

   Hint: Carol should not be able to differentiate between (x, y) and (x', y') such that $f(x, y) = f(x', y')$.

   (c) Suppose $f(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise.} \end{cases}$ Let R be the set of all permutations over D (so that $|R| = |D|!$), $g_A = g_B = g$ where $g(w, r) = r(w)$ (i.e., apply the permutation r to w). What should g_C be so that the protocol meets the correctness requirement? Also, prove that the perfect secrecy condition above is met.

   (d) Give a secure protocol for the case when $f : \{0, 1\} \times \{0, 1\} \to \{0, 1\}$ is the AND function (i.e., $f(x, y) = 1$ iff $x = y = 1$). No proof is required.

   Hint: You can use the protocol from the previous part that computes $f_{\text{eq}} : D' \times D' \to \{0, 1\}$, for an appropriately chosen D'. Alice and Bob would locally map $x, y \in D$ to $x', y' \in D'$, before invoking the protocol for f_{eq}.

2. **IND-CPA and Perfect Correctness $\Longrightarrow$ SIM-CPA** [15 pts]

   Show that a perfectly correct encryption scheme that is IND-CPA secure is SIM-CPA secure.

   Hint: You can use a simulator similar to the one we used for showing the analogous result for IND-Onetime and SIM-Onetime. Use a reduction to argue that if the simulation is not good against some PPT adversary and environment, then you can break the IND-CPA security.

3. **Hybrid Argument: Next-Bit Unpredictability implies Pseudorandomness.** [20 pts]

Let Y denote a distribution ensemble over $\{0,1\}^n$, where n is a polynomial function of the security parameter k. Y is said to be *next-bit unpredictable* if, for all PPT algorithms B, $\max_{i\in[n]} |\Pr_{y\leftarrow Y_k}[B(y_1^{i-1}) = y_i] - 1/2|$ is negligible. Y is said to be *pseudorandom* if for all PPT A, $|\Pr_{y\leftarrow Y_k}[A(y) = 0] - \Pr_{y\leftarrow Un}[A(y) = 0]|$ is negligible, where U_n denote the uniform distribution over $\{0,1\}^n$.

Given a PPT distinguisher A, define a PPT predictor B to be as follows:

On input $z \in \{0,1\}^{i-1}$, pick $b \leftarrow \{0,1\}$, $r \leftarrow \{0,1\}^{n-i}$ and output $A(z||b||r) \oplus b$. (Here || denotes concatenation)

For each $i \in [n]$, define the distribution H_i over n-bit strings as the distribution of the string z produced by taking $y \leftarrow Y_k$, $r \leftarrow U_{n-i}$, and letting $z := y_1^i||r$. Note that $H_0 = U_n$ and $H_n = Y_k$. For parts (a)-(e), fix an $i \in [n]$.

(a) Let $\alpha(z) := \Pr_{y\leftarrow Y_k}[y_1^{i-1} = z]$ and $q(z,b) := \Pr_{r\leftarrow\{0,1\}^{n-i}}[A(z||b||r) = 0]$ for all $z \in \{0,1\}^{i-1}$ and $b \in \{0,1\}$. Compute $\Pr_{y\leftarrow H_{i-1}}[A(y) = 0]$ in terms of these two functions.

(b) Also, let $\beta(z) := \Pr_{y\leftarrow Y_k}[y_i = 0 \mid y_1^{i-1} = z]$. Now, compute $\Pr_{y\leftarrow H_i}[A(y) = 0]$.

(c) For each $z \in \{0,1\}^{i-1}$, let $\gamma(z) := \Pr[B(z) = 0]$ (where the probability is over the randomness of the algorithm B above). Compute $\gamma(z)$ in terms of the quantities $q(z,0)$ and $q(z,1)$.

(d) Show that $\Pr_{y\leftarrow Y_k}[B(y_1^{i-1}) = y_i \mid y_1^{i-1} = z] = \frac{1}{2} + 2(\beta(z) - \frac{1}{2})(\gamma(z) - \frac{1}{2})$, for each $z \in \{0,1\}^{i-1}$.

(e) From the above parts establish that

$$|\Pr_{y\leftarrow Y_k}[B(y_1^{i-1}) = y_i] - \frac{1}{2}| = |\Pr_{y\leftarrow H_i}[A(y) = 0] - \Pr_{y\leftarrow H_{i+1}}[A(y) = 0]|.$$

(f) Using the fact that part (e) holds for all $i \in [n]$, show that

$$|\Pr_{y\leftarrow Y_k}[A(y) = 0] - \Pr_{y\leftarrow U_n}[A(y) = 0]| \le n \cdot \max_{i\in[n]} |\Pr_{y\leftarrow Y_k}[B(y_1^{i-1}) = y_i] - \frac{1}{2}|.$$

If Y is next-bit unpredictable, then the RHS above is negligible (n being polynomial in k), and hence so is the LHS. Since this holds for every PPT adversary A, we conclude that if Y is next-bit unpredictable, then it is pseudorandom.

In the lecture we saw that pseudorandomness implies next-bit unpredictability. The above completes the argument that the two definitions are equivalent.

4. **Impossibility of deterministic CPA-secure encryption.** Suppose a symmetric key encryption scheme has a deterministic encryption algorithm. Give an adversary in the IND-CPA experiment for SKE to show that this scheme cannot be CPA-secure. [5 pts]

A consequence of the above is that the so-called "Electronic Code Book" mode of using a block-cipher is not an IND-CPA secure SKE scheme.

5. **One-Timeness of One-Time Pad.** Consider a deterministic "two-message encryption scheme" to be a function $\mathsf{Enc}^2 : \mathcal{K} \times \mathcal{M} \times \mathcal{M} \to \mathcal{C} \times \mathcal{C}$. [5 pts]

(a) Define perfect secrecy for such an encryption scheme.

(b) Let $\mathcal{M} = \mathcal{K} = \mathcal{C}$ be the set of n-bit strings. Let $\mathsf{Enc}^2(K, m_1, m_2) = (K \oplus m_1, K \oplus m_2)$, where $\oplus$ is bit-wise xor-ing. Prove that this is **not** perfectly secret, according to your definition.

In particular, using a one-time pad to encrypt two messages will break perfect secrecy.

6. **Statistical Indistinguishability.** [10 pts]

Recall that for two distributions X and Y over n-bit strings, the *statistical difference* (a.k.a. variational distance) between them is denoted by

$$\Delta(X,Y) = \max_{S\subseteq\{0,1\}^n} |\Pr_{x\leftarrow X}[x \in S] - \Pr_{x\leftarrow Y}[x \in S]|.$$

(Alternately, this can be phrased in terms of a statistical test T, which checks if $x \in S$ for some subset S.)

(a) Suppose $G : \{0,1\}^k \to \{0,1\}^n$ is a deterministic function, where $n > k$. Let X be the distribution of the output of $G(s)$ when $s \leftarrow \{0,1\}^k$ is chosen uniformly at random. Let Y be the uniform distribution over $\{0,1\}^n$. Show that $\Delta(X,Y) \geq \frac{1}{2}$. Conclude that the output of a pseudorandom random generator is quite distinguishable from a truly random distribution, if computationally unbounded distinguishers are considered.

(b) Suppose X_k and Y_k are distributions over 2-bit strings (for all integers $k > 0$). Further suppose that for all values of k, $\Delta(X_k, Y_k) \geq 0.1$. Show that X_k and Y_k are *not* computationally indistinguishable.

You may use *non-uniform* PPT distinguishers. i.e., describe a family of distinguishers D_k, each of which runs in time polynomial in k such that $|\Pr_{x \leftarrow X_k}[D_k(x) = 0] - \Pr_{x \leftarrow Y_k}[D_k(x) = 0]| \geq \epsilon(k)$ for some function ϵ that is not negligible.

[Extra Credit] Can you show that X_k and Y_k are in fact distinguishable by a *uniform* PPT distinguisher.

7. **PRG and PRF.** True or False (give reasons): [12 pts]

(a) If $G : \{0,1\}^k \to \{0,1\}^n$ is a PRG, then so is $G' : \{0,1\}^{k+\ell} \to \{0,1\}^{n+\ell}$ defined as $G'(x \circ x') = G(x) \circ x'$ where $x \in \{0,1\}^k$, $x' \in \{0,1\}^\ell$, and $\circ$ denotes concatenation.

(b) If $F : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^n$ is a PRF, then so is:

i. $F' : \{0,1\}^k \times \{0,1\}^{m+\ell} \to \{0,1\}^{n+\ell}$ defined as $F'(s; x \circ x') = F(s; x) \circ x'$ where $s \in \{0,1\}^k$, $x \in \{0,1\}^m$, $x' \in \{0,1\}^\ell$.

ii. $F' : \{0,1\}^{k+\ell} \times \{0,1\}^m \to \{0,1\}^{n+\ell}$ defined as $F'(s \circ s'; x) = F(s; x) \circ s'$ where $s \in \{0,1\}^k$, $x \in \{0,1\}^m$, $s' \in \{0,1\}^\ell$.

8. **Block Ciphers** [8 pts]

(a) Consider an adversary in the IND-CPA experiment against a symmetric key encryption algorithm implemented using a block-cipher in the CTR mode. Describe a brute-force strategy for the adversary to recover the encryption key.

(b) A PetaFLOPS computer can execute 10^{15} floating point operations per second. If the adversary uses a 100 PetaFLOPS computer, and the block-cipher used is DES (which uses 56 bit keys), how long would your brute-force strategy take on the average to recover the key? You may suppose that a single evaluation of a block-cipher (DES or AES) takes 10 FLOPs.

What if the block-cipher used is AES with 128-bit keys?

(c) **[Extra Credit]** The triple-DES (3DES) is a block-cipher that uses the DES block-cipher three times, with three different keys. The output of 3DES with key (K_1, K_2, K_3), on input x is defined as $\text{3DES}_{(K_1,K_2,K_3)}(x) := \text{DES}_{K_1}(\text{DES}^{-1}_{K_2}(\text{DES}_{K_3}(x)))$ where DES_K and DES^{-1}_K stand for the application of the DES block-cipher in the forward and reverse directions. Since DES has 56 bit keys, 3DES has 168 bit keys.

As before, your goal is to design a key-recovery algorithm for an adversary in the IND-CPA experiment for an SKE scheme using 3DES in CTR mode. Your algorithm can also invoke the DES block-cipher locally as a black-box (in either forward or reverse directions) with keys of your own choice.

Can you devise a key-recovery algorithm which invokes the DES block-cipher computation "only" about 2^{112} times. How much memory does your algorithm use?

9. **One-way, but every single bit of the preimage is predictable:** [10 pts]

For any function $f : \{0,1\}^* \to \{0,1\}^*$, define a function g_f as follows $g_f(x, S) = (f(x|_S), S, x|_{\bar{S}})$, where S is a subset of $\{1, 2, \ldots, |x|\}$ of size $\lfloor |x|/2 \rfloor$ (represented as a bit vector of length $|x|$ with $\lfloor |x|/2 \rfloor$ 1's). Here $x|_S$ denotes the string obtained by choosing only those bits from x whose indices are in S and $x|_{\bar{S}}$ is the string containing the remaining bits.

(a) Show that if f is a one-way function, then so is g_f. You may assume that f is length-preserving (i.e., $|f(x)| = |x|$ for all x).

(b) Show that no single bit of the input is a hard-core bit for g_f.