Graduate Cryptography January 13, 2025

Problem Set 1

Lecturer: Daniel Wichs *Due: Jan 27, 2025*

You are allowed to discuss the problems with each other, but are encouraged to try solving each problem on your own first. In either case, you must write down the solution on your own and should not share written solutions with each other. If you discussed a problem with someone else, make sure to explicitly say this in your solution.

Problem 1 (Independence, Perfect Secrecy) 5 pts

Let X, Y be random variables. Show the following three statements are equivalent:

1. X, Y are independent
2. for every x, y such that $\Pr[Y = y] > 0$ we have $\Pr[X = x | Y = y] = \Pr[X = x]$
3. for every x, y, y' s.t. $\Pr[Y = y] > 0, \Pr[Y = y'] > 0$: we have

$$\Pr[X = x | Y = y] = \Pr[X = x | Y = y'].$$

Then, use this to show that the three definitions of perfect secrecy given on the class slides are equivalent.

Problem 2 (t-wise independent hash) 10 pts

A hash function $h : \mathcal{K} \times \mathcal{U} \to \mathcal{V}$ is t-wise independent if for all t distinct values $x_1, \ldots, x_t \in \mathcal{U}$ and any $y_1, \ldots, y_t \in \mathcal{V}$ we have

$$\Pr[h(K, x_1) = y_1, \ldots, h(K, x_t) = y_t] = \prod_{i=1}^{t} \Pr[h(K, x_i) = y_i] = \frac{1}{|\mathcal{V}|^t}$$

where K is a random variable that's uniform over $\mathcal{K}$.

Use the ideas we saw in class about polynomials over a finite field $\mathbb{F}$ in the construction of Shamir secret sharing to construct such a scheme for any t with $\mathcal{K} = \mathbb{F}^t$ and $\mathcal{U} = \mathcal{V} = \mathbb{F}$.

Show how to use the above to construct a message authentication code (MAC) which can be securely used to authenticate up to $(t-1)$ messages.

Problem 3 (Statistical Distance) 10 pts

We define the *statistical distance* between two distributions X, Y as $\mathsf{SD}(X,Y) = \max_{\mathcal{T}} |\Pr[X \in \mathcal{T}] - \Pr[Y \in \mathcal{T}]|$ where the max is over all sets $\mathcal{T}$.

Part A: Consider a computationally unbounded adversary $\mathcal{A}$ who gets a sample from either X or Y and wants to distinguish between by outputting 1 if it gets a sample from X, but not if it gets a sample from Y. Show that if $\mathsf{SD}(X,Y) \leq \varepsilon$ then the adversary cannot distinguish with better than ε probability:

$$|\Pr[\mathcal{A}(X) = 1] - \Pr[\mathcal{A}(Y) = 1]| \leq \varepsilon.$$

First show this for a deterministic $\mathcal{A}$ and then extend your argument to a randomized $\mathcal{A}$.

Part B: Show that for any function $G : \{0,1\}^n \to \{0,1\}^{n+1}$ the statistical distance between $G(U_n)$ and U_{n+1} is at least $1/2$ where U_ℓ denotes the uniform distribution over $\{0,1\}^\ell$.

Part C: Show that statistical distance obeys the triangle inequality: for any X, Y, Z it holds that $SD(X,Z) \leq SD(X,Y) + SD(Y,Z)$.

Part D: Show that for any (even inefficient, randomized) function f and any random variables X, Y we have $SD(f(X), f(Y)) \leq SD(X,Y)$.

Part E: Let X be uniformly random over $\{1,\ldots,n\}$ and let Y be uniformly random over $\{1,\ldots,n+1\}$. What's the statistical distance $SD(X,Y)$?

Problem 4 (Relaxing Perfect Secrecy) 15 points

Recall that perfect secrecy says that for every pair of messages $m_0, m_1 \in \mathcal{M}$ and every ciphertext $c \in \mathcal{C}$ we have $\Pr[\mathsf{Enc}(K, m_0) = c] = \Pr[\mathsf{Enc}(K, m_1) = c]$. Shannon's lower bound on perfect secrecy lead us to consider computational security against a polynomial-time attacker. Let's go back and try to relax the definition of perfect secrecy in other more statistical ways and see what we get.

- Idea 1: Define "weak ε-statistical security" if for for every pair of messages $m_0, m_1 \in \mathcal{M}$ and every ciphertext $c \in \mathcal{C}$ we have

$$|\Pr[\mathsf{Enc}(K, m_0) = c] - \Pr[\mathsf{Enc}(K, m_1) = c]| \leq \varepsilon.$$

Show that this notion does not provide any real guarantee of security even when ε is small, by constructing a scheme that achieves this notion of security with (say) $\varepsilon = 2^{-128}$, but every ciphertext completely reveals the message.

- Idea 2: Define "ε-statistical security" if for every pair of messages $m_0, m_1 \in \mathcal{M}$ and every subset $\mathcal{T} \subseteq \mathcal{C}$ we have:

$$|\Pr[\mathsf{Enc}(K, m_0) \in \mathcal{T}] - \Pr[\mathsf{Enc}(K, m_1) \in \mathcal{T}]| \leq \varepsilon.$$

In the terminology of the previous problem, this says:

$$\mathsf{SD}(\mathsf{Enc}(K, m_0), \mathsf{Enc}(K, m_1)) \leq \varepsilon.$$

Show that, unfortunately, "strong ε-statistical security" will not let us beat the Shannon bound on key size by much. In particular show that any scheme with $|\mathcal{K}| \leq |\mathcal{M}|/2$ (i.e., key is only 1-bit shorter than message) there is some $m_0, m_1 \in \mathcal{M}$ and $\mathcal{T} \subseteq \mathcal{C}$ such that

$$|\Pr[\mathsf{Enc}(K, m_0) \in \mathcal{T}] - \Pr[\mathsf{Enc}(K, m_1) \in \mathcal{T}]| \geq \frac{1}{2}.$$

You can assume that encryption is deterministic for simplicity; for an extra challenger, generalize to randomized encryption.

Problem 5 (Two-time Security?) 15 pts

Part A: Here is a natural way to define perfectly secret encryption for two messages. For any two pairs of messages (m_0, m_1) and (m'_0, m'_1) and for any ciphertexts c_0, c_1 we have

$$\Pr[\mathsf{Enc}(K, m_0) = c_0, \mathsf{Enc}(K, m_1) = c_1] = \Pr[\mathsf{Enc}(K, m'_0) = c_0, \mathsf{Enc}(K, m'_1) = c_1]$$

Show that no encryption scheme with a deterministic encryption procedure can satisfy this definition. Then generalize to the case where encryption is a randomized algorithm (i.e., has access to additional randomness beyond the secret key K).

Part B: To overcome the limitation in part A, we first relax the problem by considering statistical security as in the previous problem. We require that for all $(m_0, m_1), (m'_0, m'_1) \in \mathcal{M} \times \mathcal{M}$ and all functions $\mathcal{A}$ that output 1 bit:

$$\Pr[\mathcal{A}(\mathsf{Enc}(K, m_0), \mathsf{Enc}(K, m_1)) = 1] - \Pr[\mathcal{A}(\mathsf{Enc}(K, m'_0), \mathsf{Enc}(K, m'_1)) = 1] \leq \varepsilon.$$

Show that, even with this relaxation, no encryption scheme with a deterministic encryption procedure can satisfy the above with $\varepsilon < 1$.

Part C: We relax the problem further by considering randomized encryption schemes where, for a fixed k, m the encryption procedure $\mathsf{Enc}(k, m)$ can use additional randomness to create the ciphertext. We require perfect correctness so that for all $m \in \mathcal{M}, k \in \mathcal{K}$: $\Pr[\mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m] = 1$ where the probability is over the randomness of the encryption procedure. Show that there exists a randomized encryption scheme that achieves statistical security (as in part B) for arbitrarily small ε.

(Hint: Use t-wise independent hash functions from problem 2 with $t = 2$. Think of using the hash function to generate a one-time pad.)

Part D: Another idea to overcome the limitation in part A is to augment the encryption/decryption procedures so that they also takes the index $b \in \{0, 1\}$ of the message as an input - to encrypt m_0, m_1 we compute $\mathsf{Enc}(K, m_0, 0), \mathsf{Enc}(K, m_1, 1)$ and to decrypt c_0, c_1 we compute $\mathsf{Dec}(k, c_0, 0), \mathsf{Dec}(k, c_1, 1)$. We define pefect security as:

$$\Pr[\mathsf{Enc}(K, m_0, 0) = c_0, \mathsf{Enc}(K, m_1, 1) = c_1] = \Pr[\mathsf{Enc}(K, m'_0, 0) = c_0, \mathsf{Enc}(K, m'_1, 1) = c_1].$$

Construct a simple scheme that meets the above notion of security.

Problem 6 (Refreshing Secret Sharing) 10 pts

A secret is shared across n computers using Shamir Secret Sharing with a threshold $t \geq 2$ (t parties learn nothing, $t + 1$ can recover the secret). Every morning, a determined hacker can choose to compromise any one of the computers. The computer stays compromised for an entire day meaning that the adversary can see everything that happens on it during that day. However, by the following morning the hack is guaranteed to be discovered and the attacker is booted off from the computer. The attacker can then hack a new computer that morning (potentially the same one as the previous morning) and so it goes day after day for ever.

We want to make sure the attacker never learns the shared secret. To do so, we want to have a protocol that the n computers can run once a day to "refresh" their shares. The attacker sees everything that happens on the compromised computer during the run of the protocol. Design a protocol to solve this problem and argue that it is secure.