COL759 Cryptography Semester 1, 2023-24

Problem Set 3

Instructor: Venkata K *Due Date: 20 October 2023*

Instructions:

- Assignment must be done in groups of size at most 2. Each group must submit one pdf on Gradescope, and mention the partner's name (if any).
- All questions are compulsory in this assignment.
- All solutions must be typeset in LaTeX. For the coding questions, provide a brief explanation of your approach and upload the relevant files on Gradescope.
- (Optional) Discuss how much time was spent on each problem. This will not be used for evaluation. We will use this for calibrating future assignments.

Notations:

- Let p be a prime. $\mathbb{Z}_p$ denotes the set $\{0, 1, \ldots, p-1\}$, and $\mathbb{Z}_p^* = \{y \in \mathbb{Z}_p : \gcd(y, p) = 1\} = \{1, 2, \ldots, p-1\}$.
- For a composite number n, $\mathbb{Z}_n$ denotes the set $\{0, 1, \ldots, n-1\}$, and $\mathbb{Z}_n^* = \{y : \gcd(y, n) = 1\}$.

1. (6 marks) **CPA with Very Weak Ciphertext Integrity**

   Let $\mathcal{E} = (\mathsf{Enc}, \mathsf{Dec})$ be a private-key encryption scheme with key space $\mathcal{K}$, message space $\mathcal{M}$. We say that $\mathcal{E}$ satisfies CPA with very weak ciphertext integrity if it is CPA secure, and additionally, for any p.p.t. adversary $\mathcal{A}$,

   $$\mathsf{Pr}_{k \leftarrow \mathcal{K}, \mathcal{A}} \left[\mathsf{ct} \leftarrow \mathcal{A}() \wedge \mathsf{Dec}(k, \mathsf{ct}) \neq \perp \right]$$

   is negligible. Informally, an adversary (without receiving any information about the key) should not be able to output a valid ciphertext.

   You are given a PRF $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$. Construct an encryption scheme with key space and message space $\{0,1\}^n$, satisfying CPA with very weak ciphertext security. You are allowed at most two uses of F per encryption.

2. (8 marks) **Encryption Scheme with Threshold Decryption**

You are running an organization with n employees, and wish to have secure communication with the individuals. Let $\mathcal{E} = (\mathsf{Enc}, \mathsf{Dec})$ be a CPA secure encryption scheme with key space $\mathcal{K}$, message space $\mathcal{M} = \{0,1\}^{\geq 1}$.

Each person in the organization chooses an encryption key $k \leftarrow \mathcal{K}$ and sends it to you. To send a message m to the i^{th} person, you compute $\mathsf{ct}_i \leftarrow \mathsf{Enc}(k_i, m)$. Occasionally, you want to encrypt a message for persons $\{i, j\}$, and it should be possible to recover the message only if both parties are present (that is, both keys k_i and k_j are essential for decryption). For this purpose, you define two new algorithms $\mathsf{Enc\text{-}two}$ and $\mathsf{Dec\text{-}two}$ with the following syntax:

- $\mathsf{Enc\text{-}two}(k_i, k_j, m)$: The encryption algorithm takes two keys k_i, k_j and a message m. It outputs a ciphertext.
- $\mathsf{Dec\text{-}two}(k_i, k_j, \mathsf{ct})$: The decryption algorithm takes two keys k_i, k_j and a ciphertext. It outputs $y \in \mathcal{M} \cup \{\perp\}$.

These two algorithms must satisfy correctness:

$$\forall k_1, k_2 \in \mathcal{K}, \forall m \in \mathcal{M}, \mathsf{Dec\text{-}two}(k_1, k_2, \mathsf{Enc\text{-}two}(k_1, k_2, m)) = m.$$

To send a message m for pair $\{i, j\}$ where $i < j$, you compute $\mathsf{Enc\text{-}two}(k_i, k_j, m)$ and send it to person i and person j.

Defining security is a bit tricky since there are n parties present. For simplicity, let us consider $n = 4$, and suppose an adversary has corrupted parties 1 and 4. We require that the adversary must learn nothing about the messages that were meant for pairs other than $(1, 4)$. We will simplify it further: the adversary must learn nothing about messages meant for the pairs $(1, 2)$, $(2, 3)$ and $(3, 4)$. We capture this via the following security game.

- **Challenge Phase:** Challenger picks $k_2, k_3 \leftarrow \mathcal{K}$. The adversary sends keys k_1, k_4, as well as challenge messages $(m^0_{1,2}, m^0_{2,3}, m^0_{3,4})$ and $(m^1_{1,2}, m^1_{2,3}, m^1_{3,4})$. Challenger samples $b \leftarrow \{0,1\}$, computes $\mathsf{ct}_{1,2} \leftarrow \mathsf{Enc\text{-}two}(k_1, k_2, m^b_{1,2})$, $\mathsf{ct}_{2,3} \leftarrow \mathsf{Enc\text{-}two}(k_2, k_3, m^b_{2,3})$ and $\mathsf{ct}_{3,4} \leftarrow \mathsf{Enc\text{-}two}(k_3, k_4, m^b_{3,4})$.
- **Encryption Queries:** The adversary can make polynomially many encryption queries. Each query consists of a message m and an index-pair $\{i, j\} \in \{\{1,2\}, \{2,3\}, \{3,4\}\}$. The challenger computes $\mathsf{ct} \leftarrow \mathsf{Enc\text{-}two}(k_i, k_j, m)$ and sends to the adversary.
- **Guess:** Finally, the adversary sends its guess b', and wins if $b = b'$.

Use Enc and Dec to define $\mathsf{Enc\text{-}two}$ and $\mathsf{Dec\text{-}two}$. Prove that if $(\mathsf{Enc}, \mathsf{Dec})$ is CPA secure, then no p.p.t. adversary has non-negligible advantage in the security game defined above.

3. (8 marks) **One-time secure MACs, and Upgrading One-Time MACs to Many-Time MACs**

Similar to one-time semantic security for encryption schemes, we can also define one-query security for MACs. A MAC scheme $\mathcal{I} = (\mathsf{Sign}, \mathsf{Verify})$ with key space $\mathcal{K}$, message space $\mathcal{M}$ is one-query unconditionally secure if, for any adversary (even computationally unbounded ones), the adversary's winning probability in the following game is negligible:

- Challenger picks a MAC key $k \leftarrow \mathcal{K}$.
- Adversary sends a signing query for message m, and receives $\mathsf{Sign}(k, m)$.
- Adversary must output m' and signature σ'. It wins if $(m', \sigma') \neq (m, \sigma)$ and $\mathsf{Verify}(k, m', \sigma') = 1$.

(a) Construct a one-query unconditionally secure MAC with message space $\{0,1\}^n$, and appropriately chosen key space. The signing algorithm must be very efficient, and not use any arithmetic operations. Compute an upper bound on the winning probability of any adversary $\mathcal{A}$.

(b) Let $\mathcal{I}_1 = (\mathsf{Sign}_1, \mathsf{Verify}_1)$ be a one-time secure MAC with key space $\mathcal{K}_1$, message space $\mathcal{M}_1$ and signature space $\mathcal{T}$. Let $F : \mathcal{K}_2 \times \mathcal{X} \to \mathcal{T}$ be a secure PRF. Consider the MAC scheme $\mathcal{I} = (\mathsf{Sign}, \mathsf{Verify})$ with key space $\mathcal{K}_1 \times \mathcal{K}_2$, message space $\mathcal{M}$ and signature space $\mathcal{X} \times \mathcal{T}$:

- $\mathsf{Sign}((k_1, k_2), m)$: sample $r \leftarrow \mathcal{X}$, $\sigma_1 \leftarrow \mathsf{Sign}_1(k_1, m)$, output $(r, F(k_2, r) \oplus \sigma_1)$.
- $\mathsf{Verify}((k_1, k_2), m, (r, \sigma))$: output $\mathsf{Verify}_1(k_1, m, \sigma \oplus F(k_2, r))$

Show that $\mathcal{I}$ is a secure MAC scheme, assuming $\mathcal{I}_1$ is a secure one-time MAC scheme and F is a secure PRF.

4. (8 marks) **CCA Security vs Authenticated Encryption**

An authenticated encryption scheme is one that satisfies both CPA security and ciphertext integrity. In class, we discussed security against chosen ciphertext attacks (CCA security), which captures many active attacks such as padding oracle attacks, malleability attacks. We also saw that authenticated encryption implies CCA security. CCA security trivially implies CPA security, however CCA security does not necessarily imply ciphertext integrity. As pointed out by one of the students in class, CCA security does not capture attacks where an adversary can convert an arbitrary ciphertext into an encryption of a fixed message. Such attacks are called *plaintext integrity attacks*, defined formally via the security game below.

note that by such a forgery, he still cannot break CCA but CTINT breaks

PTXT-INT

- (**Setup Phase**) Challenger picks a key k and initializes $\mathcal{S} = \emptyset$.
- (**Query Phase**) Adversary is allowed polynomially many encryption queries. For each query m_i, the challenger sends $\mathsf{ct}_i \leftarrow \mathsf{Enc}(k, m_i)$ and sets $\mathcal{S} = \mathcal{S} \cup \{m_i\}$.
- (**Plaintext Integrity Attack**) Adversary sends a ciphertext ct^*, and wins if $\mathsf{Dec}(k, \mathsf{ct}^*) \neq \perp$, and $\mathsf{Dec}(k, \mathsf{ct}^*) \notin \mathcal{S}$.

Figure 1: Security game capturing plaintext integrity attacks. Here, the adversary is allowed to query for encryptions of any messages of its choice, and must produce a valid encryption of a new message.

Note that if a scheme is secure against ciphertext integrity attacks, then it is also secure against plaintext integrity attacks. Hence, an authenticated encryption scheme is secure against both chosen ciphertext attacks as well as plaintext integrity attacks. The converse also holds!

(a) Let $\mathcal{E}$ be an encryption scheme that satisfies security against chosen ciphertext attacks and plaintext integrity attacks. Show that $\mathcal{E}$ is a secure authenticated encryption scheme. Clearly, if it satisfies CCA security, then it also satisfies CPA security. Therefore, you only need to show that it is secure against ciphertext integrity attacks.

(b) Construct an encryption scheme $\mathcal{E}$ that satisfies CCA security, but is not secure against plaintext integrity attacks. You can use any cryptographic primitive for building $\mathcal{E}$.

Here we need to show that CCA+PTINT => CTINT. Intuitively, this is true because if an adversary breaks CTINT, he produces a ciphertext of (1) a previously queried message or (2) a new message.
If (1) happens then CCA breaks and if (2) happens then PTINT breaks.

5. (10 marks) **Modular Arithmetic and Basic Group Theory**

This problem will introduce some of the preliminaries that we will require in the second half of this course. The starting point is Euclid's algorithm for computing the *greatest common divisor* (g.c.d.) of two numbers. Euclid's algorithm, in fact, can be extended to solve the following problem:

Given two positive numbers a, b, find integers x, y such that $a \cdot x + b \cdot y = \gcd(a, b)$.

The running time of Extended Euclid's Algorithm is polynomial in the input description (that is, $\log(a) + \log(b)$).

(a) (1 mark) Let p be a prime, and $a \in \mathbb{Z}_p^*$. Use the Extended Euclid's algorithm to find an integer $b \in \mathbb{Z}_p$ such that $a \cdot b = 1 \bmod p$. The running time of your algorithm should be polynomial in $\log(p)$. Prove that this b must be unique (that is, there cannot be two different numbers $b, b' \in \mathbb{Z}_p$ such that $a \cdot b = 1 \bmod p$ and $a \cdot b' = 1 \bmod p$.

This simple result has interesting consequences, and this holds only for prime moduli. For instance, this fact is used to prove the following theorem (which we used in the analysis of our information-theoretically secure UHF):

Let $h(y) = \sum_{i=0}^{d} a_i \cdot y^i$ be a polynomial, where each $a_i \in \mathbb{Z}_p$. If h is a non-trivial polynomial (that is, it is not identically equal to the zero polynomial), then

$$|\{y \in \mathbb{Z}_p \ : \ h(y) = 0 \bmod p\}| \leq d.$$

(b) (1 mark) Give an example of a modulus n which is a product of two different primes, and a degree 2 polynomial h such that

$$|\{y \in \mathbb{Z}_n \ : \ h(y) = 0 \bmod n\}| > 2.$$

Let p be a prime. The set $\mathbb{Z}_p^*$, together with the operation 'multiplication modulo p' forms a *group*. A group consists of a set G together with an operation $\otimes$ defined on the set elements. A group must satisfy the following properties:

- there exists an identity element in the group. In the case of $\mathbb{Z}_p^*$, the identity element is 1.
- for all $a, b \in G$, $a \otimes b$ also belongs to the group. In the case of $\mathbb{Z}_p^*$, for any $a, b \in \mathbb{Z}_p^*$, $a \cdot b \bmod p$ is also present in $\mathbb{Z}_p^*$.
- for all $a, b, c \in G$, $a \otimes (b \otimes c) = (a \otimes b) \otimes c$. This property holds for $\mathbb{Z}_p^*$.
- for every element $a \in G$, there exists a unique element $b \in G$ such that $a \otimes G$ is equal to the identity element of G. We proved this in Part 5a.

For any element $a \in \mathbb{Z}_p^*$, the order of a, denoted by $\mathsf{ord}(a)$, is the smallest non-negative number t such that $a^t = 1 \bmod p$. Clearly, $\mathsf{ord}(a)$ can be at most $p-1$. This is because, if we consider the list of elements

$$a^1 \bmod p, \quad a^2 \bmod p, \quad a^3 \bmod p, \quad \ldots, \quad a^{p-1} \bmod p, \quad a^p \bmod p$$

then this list has p elements, and each element is a number in $\mathbb{Z}_p^*$. Either this list contains 1 (in which case, we look at the first position where 1 appears), or there exist two indices $1 \leq k < \ell \leq p-1$ such that $a^k \bmod p = a^\ell \bmod p$. In this case, note that $a^{\ell-k} = 1 \bmod p$.

(c) (2 marks) Let p be a prime, and $a \in \mathbb{Z}_p^*$. Prove that $\mathsf{ord}(a)$ divides $(p-1)$.

Let q be a prime such that $p = 2 \cdot q + 1$ is also a prime. Let $a \in \mathbb{Z}_p^*$. From Part 5c, it follows that $\mathsf{ord}(a) \in \{1, 2, q, p-1\}$. If $\mathsf{ord}(a) = p-1$ and $b = a \cdot a \bmod p$, then $\mathsf{ord}(b) = q$. Moreover, most elements in $\mathbb{Z}_p^*$ have order either q or $p-1$ (only two elements in $\mathbb{Z}_p^*$ have order 2; why?). This suggests that it is easy to sample an element in $\mathbb{Z}_p^*$ that has order q. Let $a \in \mathbb{Z}_p^*$ be such that $\mathsf{ord}(a) = q$, and let

$$\langle a \rangle_p = \left\{ a^0 \bmod p, a^1 \bmod p, a^2 \bmod p, \ldots, a^{q-1} \bmod p \right\}.$$

This set, together with the operation *multiplication modulo* p forms a multiplicative group. Moreover, the size of this group is q, which is itself a prime. Such prime-order groups are extremely useful in cryptography because certain computational problems are believed to be hard on such groups.

From this point onwards, we will talk about prime-order groups $(G, \otimes)$ abstractly. In such groups, the elements of G can be represented efficiently, and the group operation $\otimes$ can be performed efficiently. Below, we discuss two problems that are believed to be computationally hard for certain prime-order groups. Let $q = |G|$.

- *Discrete Log Problem:* Given q, a uniformly random group element $g \in G$ and $h = g^a$ for uniformly random $a \in \mathbb{Z}_q$, compute a. Note that a is uniquely defined given q, g, h.
- *Decisional Diffie-Hellman (DDH) Problem:* The following two distributions are computationally indistinguishable: [1]

$$\left\{ (g, g^a, g^b, g^{a \cdot b}) \ : \ g \leftarrow G, a, b \leftarrow \mathbb{Z}_q \right\}$$
$$\left\{ (g, g^a, g^b, g^c) \ : \ g \leftarrow G, a, b, c \leftarrow \mathbb{Z}_q \right\}$$

  This problem was used by Whitfield Diffie and Martin Hellman for designing the first key-exchange protocol.

[1] Two efficiently-sampleable distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ are said to be computationally indistinguishable if no p.p.t. adversary can distinguish between a sample from $\mathcal{D}_0$ and a sample from $\mathcal{D}_1$.

(d) (2 marks) **A DDH variant**

Let G be a prime order group of size $q = \Theta(2^n)$. Consider the following variant of Decision Diffie-Hellman, which we call $\mathsf{powersDDH}$. For any p.p.t. adversary, the following two distributions are computationally indistinguishable:

$$\mathcal{D}_0 = \left\{ \left(g, g^a, g^{a^2}, g^{a^3}\right) \ : \ g \leftarrow G, a \leftarrow \mathbb{Z}_q \right\}$$
$$\mathcal{D}_1 = \left\{ \left(g, g^a, g^b, g^c\right) \ : \ g \leftarrow G, a, b, c \leftarrow \mathbb{Z}_q \right\}$$

Show that if $\mathsf{powersDDH}$ is hard for G, then so is the DDH problem. More formally, show that if there exists an adversary that solves the DDH problem, then there exists a reduction algorithm that solves $\mathsf{powersDDH}$. You don't need to provide the analysis, just describe how the reduction would work.

(e) (4 marks) Let q be an n-bit prime, and G be a q-size group such that the DDH problem is hard on G. Let $t = \mathsf{poly}(n)$. For any matrix $\mathbf{M} \in \mathbb{Z}_q^{t \times t}$ and $g \in G$, let $g^{\mathbf{M}}$ denote the following matrix:

$$g^{\mathbf{M}} = \begin{bmatrix} g^{\mathbf{M}[1,1]} & g^{\mathbf{M}[1,2]} & \dots & g^{\mathbf{M}[1,t]} \\ g^{\mathbf{M}[2,1]} & g^{\mathbf{M}[2,2]} & \dots & g^{\mathbf{M}[2,t]} \\ \vdots & \vdots & \ddots & \vdots \\ g^{\mathbf{M}[t,1]} & g^{\mathbf{M}[t,2]} & \dots & g^{\mathbf{M}[t,t]} \end{bmatrix}$$

Let $\mathsf{Rank}_1[t, q] \subset \mathbb{Z}_q^{t \times t}$ denote the set of all $t \times t$ rank one matrices.[2] Show that the following two distributions are computationally indistinguishable.

$$\mathcal{D}_0 = \left\{ \left(g, g^{\mathbf{M}}\right) \ : \ g \leftarrow G, \mathbf{M} \leftarrow \mathsf{Rank}_1[t, q] \right\}$$
$$\mathcal{D}_1 = \left\{ \left(g, g^{\mathbf{M}}\right) \ : \ g \leftarrow G, \mathbf{M} \leftarrow \mathbb{Z}_q^{t \times t} \right\}$$

The proof will involve a sequence of hybrid-experiments. World-0 will correspond to sampling from $\mathcal{D}_0$, World-1 will correspond to sampling from $\mathcal{D}_1$, and the intermediate hybrids will be used to gradually transition from World-0 to World-1. Each of these transitions must only rely on the assumption that DDH is hard for group G.

---

[2] A matrix $\mathbf{M}$ is said to be rank one if there exists a vector $\mathbf{v}$ and scalars $\lambda_1, \dots, \lambda_t$ such that the i^{th} row of the matrix is $\lambda_i \cdot \mathbf{v}$ for all i.