CS255: Introduction to Cryptography **Winter 2025**

Final Exam

Instructions:

- **Please answer all five questions. You have 2.5 hours.**
- The exam is open book, open notes, open laptops, and open Internet (to consult a static online resource such as the course textbook or Wikipedia). You are expected to do the exam on your own. A web search engine, such as Google, is not allowed. **You may not interact, collaborate, or discuss the exam with another person or an AI chat bot during the exam day. That would be a gross violation of the honor code.**
- To submit your answers please either (i) use the provided LaTeX template, or (ii) use the provided PDF with a tablet and write your answers in the provided spaces, or (iii) write your answers on a sheet of paper, starting every question on a new page. When done, please upload your solutions to Gradescope (course code `KZVDDK`). **We added fifteen minutes to the exam to give you time to upload your answers to Gradescope.**
- The **LaTeX template** for the final is available at here. Please do not share this link with others.
- Please post any clarification questions privately on Ed.
- Students are bound by the Stanford honor code. In particular, you are expected to do the exam on your own.

Problem 1. (*Questions from all over*) **(20 points)**

a. Let (E, D) be a cipher that provides authenticated encryption, where algorithms E and D take an additional nonce as input. Briefly explain how a developer should choose the nonce n when invoking $E(k, m, n)$ for some key k and message m.

Your answer:

b. Let p be a large prime and $g \in \mathbb{Z}_p^*$ of order $p-1$. Is the function $f(x) := g^x$ in $\mathbb{Z}_p$ whose domain is $\{1, \ldots, p-1\}$ a *trapdoor* one-way function? Justify your answer.

Your answer:

c. Let (N, e) be an RSA public key and let (N, d) be the corresponding RSA private key. Recall that to sign a message m using RSA-FDH we use a hash function $H : \{0,1\}^n \to \mathbb{Z}_N$ and compute the signature on m as $\sigma \leftarrow H(m)^d$ in $\mathbb{Z}_N$. Suppose the adversary can find two messages m_1, m_2 such that $H(m_1) = H(m_2) \cdot 2^e$ in $\mathbb{Z}_N$. Does this let the adversary break RSA-FDH? That is, can the adversary create an existential forgery using a chosen message attack?

Your answer:

d. When storing hashed and salted passwords in a password file, what is the purpose of using a slow hash function?

Your answer:

e. What is a harvest now decrypt later (HNDL) attack and why does it necessitate a transition to new public key encryption schemes? What are those new public key encryption schemes?

Your answer:

f. Let (S, V) be a secure MAC scheme defined over $(\mathcal{K}, \mathcal{M}, \mathcal{T})$, where the key space $\mathcal{K}$ is contained in $\mathcal{M} \cap \mathcal{T}$. Define a new MAC scheme (S', V') as

$$S'(k,m) := \begin{cases} S(k,m) & \text{if } m \neq k, \text{ and} \\ k & \text{if } m = k \end{cases} \qquad V'(k,m,t) := \begin{cases} V(k,m,t) & \text{if } m \neq k, \text{ and} \\ \text{“yes”} & \text{if } m = k \end{cases}$$

Is (S', V') a secure MAC? Briefly justify your answer.

Your answer:

Problem 2. (*Two-time secure encryption*) **(20 points)** Recall that the one-time-pad is a one-time encryption system that is secure against infinitely powerful adversaries. Our goal in this question is to design a *2-time* secure encryption against infinitely powerful adversaries. If the encryptor can be stateful then the problem is trivial — simply use two one-time pads. Here, we design a stateless 2-time secure system: every encryption is done independently of the other encryptions.

a. Give a precise definition for what it means for a symmetric encryption system to be semantically secure when a secret key is used to encrypt at most two messages. Make sure to define two experiments EXP(0) and EXP(1) as in the definition of CPA security (recall that in the CPA security definition a secret key can be used to encrypt many messages). Keep in mind that the adversary can be adaptive: its choice for a second message pair to encrypt may depend on the first ciphertext it receives from the challenger.

Your answer:

b. Show that the one time pad is insecure under your definition from part (a). Can any deterministic encryption system (without a nonce) be secure under your definition?

Your answer:

c. Let p be a 128-bit prime and consider the following encryption system: the secret key is a random pair $(x, y) \in (\mathbb{Z}_p)^2$ and to encrypt a message $m \in \mathbb{Z}_p$ do:

$$\text{choose a random } r \xleftarrow{R} \mathbb{Z}_p \text{ and output the ciphertext } (c_1, c_2) := (r,\ xr + y + m) \in \mathbb{Z}_p^2.$$

We stress that a fresh r is sampled on every invocation of the encryption algorithm. Explain how to decrypt a given ciphertext (c_1, c_2) using the secret key (x, y).

Your answer: output m **where** $m =$

d. One can show that the encryption scheme from part (c) is two-time secure (using your definition from part (a)) against infinitely powerful adversaries. Show that this scheme is not 3-time secure. That is, show an adversary that can distinguish EXP(0) from EXP(1) after making three encryption queries.

Your answer:

Problem 3. (*Strongly secure signatures*) **(20 points)** In Lecture 13 we defined security for a signature scheme by requiring that the adversary cannot forge a signature on a messages m if it did not previously ask the challenger for a signature on m. This definition does not preclude the adversary from obtaining a valid message-signature pair (m, σ) and then generating a new signature $\sigma' \neq \sigma$ for the same message m. For example, the signature scheme ECDSA is believed to be secure, but given a valid signature σ on m it is possible to find another valid signature σ' on the same m. This has led to some unexpected attacks on systems than use ECDSA.

We say that a signature scheme is **strongly secure** if the adversary cannot generate a new signature for a previously signed message. One can formulate this as a game similar to the MAC security game. In this question we will develop a general way to transform a secure signature scheme into a strongly secure one.

a. To start, let $h : \mathcal{X} \to \mathcal{Y}$ be a one-way function. Let us define another function $h' : \{0,1\} \times \mathcal{X} \to \mathcal{Y}$ defined as $h'(b, x) := h(x)$. One can show that h' is also a one-way function. Show that h' is not collision resistant by exhibiting a collision. This shows that a function can be one-way, but not collision resistant.

Your answer:

b. Consider the Lamport one-time signature scheme (Lecture 14) built from a one-way function. We showed in class that the scheme is secure as long as the adversary obtains at most one message-signature pair. Is this scheme *strongly* secure as long as the adversary obtains at most one message-signature pair?
Hint: think what happens if you use h' from part (a) as the one-way function in the Lamport signature scheme.

Your answer:

c. What additional property can we require that the one-way function $h : \mathcal{X} \to \mathcal{Y}$ satisfy to ensure that the derived Lamport scheme is strongly one-time secure?

Your answer:

d. So now we have a strongly secure one-time signature scheme $(G_{\mathsf{ots}}, S_{\mathsf{ots}}, V_{\mathsf{ots}})$. Let's use that to make any secure signature scheme strongly secure. Let (G, S, V) be a (many-time) secure signature scheme that is not strongly secure (such as ECDSA). Consider the following candidate for a strongly secure signature scheme (G, S', V') derived from (G, S, V):

$$S'(\mathsf{sk}, m) := \left\{ \begin{array}{l} (\mathsf{pk}_{\mathsf{ots}}, \mathsf{sk}_{\mathsf{ots}}) \xleftarrow{R} G_{\mathsf{ots}}(), \\ \sigma_0 \xleftarrow{R} S(\mathsf{sk}, \mathsf{pk}_{\mathsf{ots}}), \quad \sigma_1 \xleftarrow{R} S_{\mathsf{ots}}(\mathsf{sk}_{\mathsf{ots}}, m), \\ \text{output } (\sigma_0, \sigma_1, \mathsf{pk}_{\mathsf{ots}}) \end{array} \right\} \tag{1}$$

$$V'\big(\mathsf{pk}, m, (\sigma_0, \sigma_1, \mathsf{pk}_{\mathsf{ots}})\big) := \Big\{\text{accept if } V(\mathsf{pk}, \mathsf{pk}_{\mathsf{ots}}, \sigma_0) = V_{\mathsf{ots}}(\mathsf{pk}_{\mathsf{ots}}, m, \sigma_1) = \text{“yes”}\Big\}$$

One can show that this signature scheme is secure. We might hope that it is also strongly secure because the message m is signed by a (one-time) strongly secure scheme. However, that is not the case. Show that an adversary that is given a valid signature (σ_0, σ_1) on a message m, can construct a new signature (σ_0', σ_1') on the same m.
Hint: Use the fact that (G, S, V) is not strongly secure. That is, there is an algorithm $\mathcal{A}$ that is invoked as $\mathcal{A}(\mathsf{pk}, m, \sigma) \to \sigma'$ such if σ is a valid signature on m then so is σ', and $\sigma' \neq \sigma$.

Your answer:

e. Show how to enhance the construction in (1) to make the resulting (G, S', V') strongly secure. **Hint:** try to add something more for S_{ots} to sign, in addition to m, when generating σ_1.

Your answer:

$$S'(\mathsf{sk}, m) := \begin{cases} (\mathsf{pk}_{\mathsf{ots}}, \mathsf{sk}_{\mathsf{ots}}) \xleftarrow{R} G_{\mathsf{ots}}(), \\ \sigma_0 \xleftarrow{R} S(\mathsf{sk}, \mathsf{pk}_{\mathsf{ots}}) \\ \sigma_1 \xleftarrow{R} \quad \underline{\qquad\qquad\qquad\qquad} \\ \textbf{output } (\sigma_0, \sigma_1, \mathsf{pk}_{\mathsf{ots}}) \end{cases}$$

$$V'\big(\mathsf{pk}, m, (\sigma_0, \sigma_1, \mathsf{pk}_{\mathsf{ots}})\big) := \Big\{ \textbf{accept if} \quad \underline{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad} \Big.$$

f. Briefly explain why your signature scheme from part (e) is a (many-time) strongly secure signature scheme.

Your answer:

Problem 4. (*Private Information Retrieval*) **(20 points)** A phone manufacturer called Avocado wants to provide a spam filtering service. Avocado maintains a list L of all the known spamming phone numbers. When Alice receives a phone call, her phone will check Avocado's list to see if the incoming number is in L, and if so, her phone will block the call. The list L is too big to download to the phone, so it must be kept server side. Naively, the phone would send every caller's number to the Avocado server, but this violates Alice's privacy since the server learns all the phone numbers of people who call Alice. Your goal in this problem is to design a way for Alice to look up the spam status of a caller without revealing the caller's number to the server. The total communication between Alice and the server should be much less than the size of L.

To simplify the problem, let us assume that the server has a vector $L = (\ell_0, \ldots, \ell_{m-1}) \in \{0,1\}^m$ and Alice's phone has an index $i \in \{0, \ldots, m-1\}$. Our goal is to design a low communication protocol between the phone and the server so that at the end of the protocol the phone learns $\ell_i \in \{0,1\}$, while the server learns nothing about i. This problem is called *Private Information Retrieval* or PIR. The spam problem above can be reduced to this problem using a suitable data structure. Our goal is develop a PIR protocol that uses $O(\sqrt{m})$ communication between the phone and server — much less than $O(m)$.

To build a PIR protocol we will use an *additively homomorphic* encryption scheme (G, E, D, A). Algorithms G, E, D are as in a standard public key encryption scheme where the plaintext space is $\mathcal{M} := \mathbb{Z}_q$ for some prime q. The new algorithm A is invoked as $A(\mathsf{pk}, \boldsymbol{\alpha}, \boldsymbol{c}) \to c$, where pk is a public key, $\boldsymbol{\alpha} = (\alpha_0, \ldots, \alpha_{s-1})$ is a vector in $\mathbb{Z}_q^s$, and $\boldsymbol{c} = (c_0, \ldots, c_{s-1})$ is a vector of s ciphertexts. The output c is a single ciphertext. The algorithm satisfies the following property: suppose that for all $i = 0, \ldots, s-1$ we have $D(\mathsf{sk}, c_i) = m_i$, then $D(\mathsf{sk}, c) = \alpha_0 m_0 + \cdots + \alpha_{s-1} m_{s-1} \in \mathbb{Z}_q$.

a. Suppose that (G, E, D) is a semantically secure public key encryption system. Explain why the existence of algorithm A means that (G, E, D) cannot be chosen ciphertext secure. That is, describe an attack that lets an adversary win the chosen ciphertext (CCA) security game.

Your answer:

Next, suppose that the dimension of the vector L is s^2 for some integer s. The Avocado server arranges the elements of L into an $s \times s$ matrix $M \in \{0,1\}^{s \times s}$. For example, if $s = 3$ then

$$M = \begin{pmatrix} \ell_0 & \ell_1 & \ell_2 \\ \ell_3 & \ell_4 & \ell_5 \\ \ell_6 & \ell_7 & \ell_8 \end{pmatrix} \in \{0,1\}^{3 \times 3}.$$

We refer to the left most column of M as column 0 and the right most column of M as column $s-1$.

The phone and server now run the following PIR protocol:

- *step i:* The phone generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow G()$.
- *step ii:* Suppose the phone wants entry ℓ_i in L. Let $0 \le j < s$ be the number of the column in M that contains ℓ_i. The phone constructs the column vector $\boldsymbol{e}_j \in \{0,1\}^s$ that is zero everywhere except at position j where it is 1. For example, if $s = 3$ and $i = 4$, then $j = 1$ and $\boldsymbol{e}_j = (0,1,0)^\top$. The phone sends pk and the following vector of s ciphertexts to the server

$$\boldsymbol{c} \xleftarrow{R} \Big(E(\mathsf{pk}, \boldsymbol{e}_j[0]), \ldots, E(\mathsf{pk}, \boldsymbol{e}_j[s-1]) \Big).$$

- *step iii:* The server responds with a vector of s ciphertexts that is the encryption of the vector $M \cdot \boldsymbol{e}_j \in \{0,1\}^s$ obtained by multiplying the vector $\boldsymbol{e}_j$ by the matrix M. We denote this vector by $\boldsymbol{c}' = (c'_0, \ldots, c'_{s-1})$. Every element in the vector $\boldsymbol{c}'$ is a ciphertext.

b. Explain how the server computes the vector of ciphertexts $\boldsymbol{c}'$ using only M, $\boldsymbol{c}$, and pk.
Hint: use algorithm A.

Your answer:

c. Explain how the phone uses $\boldsymbol{c}'$ and its secret key sk to obtain the required ℓ_i.

Your answer:

d. Explain why the server learns nothing about i (or j) in this protocol.

Your answer:

The total communication back and forth in this protocol is only $2\sqrt{m}$ ciphertexts, where m is the size of L. It remains to construct an additively homomorphic encryption scheme. That can be easily done using ElGamal encryption, but we will leave that for another day.

Problem 5. (*Expanding the domain and range of a PRF*) **(20 points)** Let F be a secure PRF defined over $(\mathcal{K}, \mathcal{X}, \mathcal{X})$ where $\mathcal{X} = \{0,1\}^n$. Our goal is to define a secure PRF that has a larger domain or a larger range

a. Consider the PRF $F_1(k, x) := \Big(F(k, x), F(k, x \oplus c)\Big)$ where $0 \neq c \in \mathcal{X}$ is some fixed value. Then F_1 has range $\mathcal{X} \times \mathcal{X}$, which is twice the size of the range of F. Is F_1 a secure PRF? Justify your answer.

Your answer:

b. The correct way to expand the range of F is using a secure PRG $G : \mathcal{X} \to \mathcal{X}^2$ by setting

$$F_G(k, x) := G(F(k, x)). \tag{2}$$

One can show that if G is a secure PRG and F is a secure PRF, then F_G is a secure PRF whose range in $\mathcal{X}^2$. Since all we have at our disposal is F, we need to build G from F. We can do so using the idea behind deterministic counter mode encryption, where we built a PRG from a PRF. Write out the explicit construction from (2) where G is built from F. Your answer should only invoke F, possibly multiple times.

Your answer: $F_G(k, x) :=$

c. Finally, consider the following PRG $G : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ for $m > n$ and some prime q. The designer of the PRG samples a random $n \times m$ matrix $A \in \mathbb{Z}_q^{n \times m}$. The matrix A is then fixed in the PRG standard and made public. For $s \in \mathbb{Z}_q^n$ the PRG outputs the matrix-vector product

$$G(s) := A \cdot s \in \mathbb{Z}_q^m$$

This PRG expands its input from a vector of dimension n to a vector of dimension $m > n$. Is this a secure PRG? Justify your answer.
Hint: Observe that every output of G is some linear combination of the columns of A. These columns span a sub-space of $\mathbb{Z}_q^m$.

Your answer: