CS255: Introduction to Cryptography **Winter 2023**

Take-home Final Exam

Instructions:

- **Please answer all five questions. You have three hours.**
- You may take the exam at any time during the exam window. You have three hours from the moment you begin until the moment you submit your answers on Gradescope.
- The exam is open book, open notes, open laptops, and open Internet (e.g., to consult a static online resource). However, you are expected to do the exam on your own. **You may not interact, collaborate, or discuss the exam with another person during the exam window, or with an AI chat bot.**
- Questions: please post privately on Ed.
- To submit your answers please either (i) use the provided LaTeX template, or (ii) print out the exam and write your answers in the provided spaces, or (iii) write your answers on blank sheets of paper, but please make sure to start each question on a new page. When done, please upload your solutions to Gradescope (course code `PX6887`).
- The LaTeX template for the final is available here. Please do not share the link with others.
- Students are bound by the Stanford honor code. In particular, you are expected to do the exam on your own.

Problem 1. (25 points) Questions from all over.

a. Does counter mode encryption require a PRP, or is a PRF sufficient? Justify your answer.

Your answer:

b. Recall that the one-time pad is defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ where $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^n$ and $E(k,m) = k \oplus m$. Notice that when the key $k = 0^n$ is used, then $E(k,m) = m$ and this does not seem secure. Suppose we improve the one-time pad by setting the key space to $\mathcal{K} := \{0,1\}^n \setminus \{0^n\}$. That is, we take 0^n out of the key space so that it will never be chosen as a key. Does the resulting cipher have perfect secrecy? Justify your answer.

Your answer:

c. Let $\mathbb{G}$ be a group of prime order p with generator $g \in \mathbb{G}$. Assume that the discrete log problem is hard in $\mathbb{G}$. Consider the following PRG defined over $(\mathbb{Z}_p, \mathbb{G}^2)$: given an input $x \in \mathbb{Z}_p$, the PRG outputs $G(x) := (g^{4x}, g^{5x}) \in \mathbb{G}^2$. Is this a secure PRG? If so, explain why. If not, describe an attack.

Your answer:

d. What is the smallest possible positive value of e that can be used to define the RSA trapdoor function? Explain why a smaller value of e cannot be used.

Your answer:

e. In the ElGamal public key encryption system, is it safe to fix the group $\mathbb{G}$ and generator $g \in \mathbb{G}$, so that all users in the world use the same $(\mathbb{G}, g)$?

Your answer:

Problem 2. (21 points) Hash functions.

a. Show that if H_1 and H_2 are distinct collision resistant functions with range $\mathcal{T} := \{0,1\}^n$, then $H(x) := H_1(x) \oplus H_2(x)$ need not be collision resistant.
Hint: Let F be a collision resistant hash function with range $\mathcal{T}$. Use F to construct two collision resistant functions H_1, H_2 such that H is not collision resistant.

Your answer:

b. The UNIX `crypt` function is a hash function that only looks at the first eight bytes of the input message. For example, `crypt(helloworld)` returns the same value as `crypt(hellowor)`.

Consider the following MAC system (S, V) whose key space $\mathcal{K}$ is the set of eight character strings:

$$S(k,m) := \texttt{crypt}(m \parallel k) \qquad ; \qquad V(k,m,t) := \{\text{output } \texttt{yes} \text{ if } t = \texttt{crypt}(m \parallel k)\}$$

Here $\parallel$ denotes string concatenation. Show that this MAC is vulnerable to a chosen message attack. In particular, show that an adversary can recover k with 8×256 chosen message queries.

Your answer:

c. Let F be a secure PRF defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$ where $\mathcal{X} = \mathbb{F}_2^n$. Recall that $\mathbb{F}_2 = \{0, 1\}$ where addition and multiplication are defined modulo two (e.g., $1+1=0$ and $1 \times 1 = 1$) and $\mathbb{F}_2^n$ can be treated as the set of all n-bit strings. Let's try to construct a secure MAC for messages in $\mathbb{F}_2^\ell$, namely ℓ-bit messages, where $\ell > n$.

Let A be an $n \times \ell$ matrix over $\mathbb{F}_2$ where $\ell > n$. This matrix A is fixed and public. Here is a simple proposal for a MAC algorithm for messages m in $\mathbb{F}_2^\ell$:

$$S(k, m) := \{x \leftarrow A \cdot m \in \mathbb{F}_2^n, \quad \text{output } t \leftarrow F(k, x)\}$$
$$V(k, m, t) := \{\text{accept if } t = F(k, Am)\}$$

Someone who did not take CS255 might argue that without k, an attacker cannot compute the tag for a message m of its choice. However, show that this proposal is flawed: an attacker can carry out an existential forgery given one valid message-tag pair.

Hint: Use the fact that because $\ell > n$, the kernel of A is non-trivial, meaning that there are non-zero vectors $z \in \mathbb{F}_2^\ell$ such that $Az = 0$.

Your answer:

Problem 3. (16 points) Private broadcast encryption

a. Let (Gen, E, D) be a public key encryption scheme. Let $\mathrm{pk}_1, \mathrm{pk}_2, \ldots, \mathrm{pk}_n$ be n public keys generated using $Gen()$. Bob wants to send a long secret message M, such as a video, to all n recipients and he has a broadcast channel to do so. Think of Bob sending his message to a satellite in orbit which then beams the message to all n recipients at once.

Naively, Bob can encrypt the message M under each recipient's public key and broadcast the resulting n ciphertexts all at once. This results in a broadcast message of size $O\big(n \times \mathrm{len}(M)\big)$. Show that Bob can encrypt M so that the broadcast message size is only $O\big(n + \mathrm{len}(M)\big)$.

Hint: you may use a symmetric cipher $(E_{\mathrm{sym}}, D_{\mathrm{sym}})$ with key space $\mathcal{K}$ that provides authenticated encryption.

Your answer:

b. Next, with the same setup as in part (a), suppose that Bob wants to send the secret message M to a subset S of the n recipients, namely some $S \subseteq \{1, \ldots, n\}$. However, it is important to Bob that the set S remains secret, even from the n recipients. Of course, if recipient number i can obtain the message M from the broadcast, then it learns that i is in S. However, it should learn nothing else about the set S. For example, think of a TV-satellite system where the broadcaster does a global broadcast, but wants to keep the set of active subscribers secret. Show how to modify your construction from part (a) to satisfy this goal. Note that everyone will see the satellite broadcast, so Bob cannot solve this problem but restricting the physical broadcast to just the members of S.

To reiterate, Bob needs a way to encrypt M so that: (i) every recipient in S should receive M, (ii) recipients not in S should learn nothing about M (other than its length), and (iii) recipient i learns if i is in S, but should learn nothing else about S.
Hint: the broadcast message size is still $O\big(n + \mathrm{len}(M)\big)$ even though the number of parties who receive M may be much smaller than n.

Your answer:

Problem 4. (22 points) Variants of ElGamal. Let $\mathbb{G}$ be a group of prime order q with generator $g \in \mathbb{G}$. Here is a simplified version of the ElGamal public key encryption scheme:

$$\begin{aligned} Gen() &:= \{\alpha \xleftarrow{R} \mathbb{Z}_q, \quad \mathrm{sk} \leftarrow \alpha, \quad \mathrm{pk} \leftarrow g^{\alpha}\} \\ E(\mathrm{pk}, m) &:= \{\rho \xleftarrow{R} \mathbb{Z}_q, \quad u \leftarrow g^{\rho}, \quad v \leftarrow m \cdot \mathrm{pk}^{\rho}, \quad ct \leftarrow (u, v)\} \\ D(\mathrm{sk}, (u, v)) &:= v / u^{\mathrm{sk}} \end{aligned}$$

One can show that this scheme is semantically secure assuming a problem called Decision Diffie-Hellman (DDH) is hard in $\mathbb{G}$.

Let us consider three variants of the encryption algorithm:

$$E_1(\mathrm{pk}, m) := \{\rho \xleftarrow{R} \mathbb{Z}_q, \quad u \leftarrow g^{\rho+1}, \quad v \leftarrow m \cdot \mathrm{pk}^{\rho}, \quad ct \leftarrow (u, v)\}$$

$$E_2(\mathrm{pk}, m) := \{\rho \xleftarrow{R} \mathbb{Z}_q, \quad u \leftarrow \rho, \quad v \leftarrow m \cdot \mathrm{pk}^{\rho}, \quad ct \leftarrow (u, v)\}$$

$$E_3(\mathrm{pk}, m) := \{\rho, \psi \xleftarrow{R} \mathbb{Z}_q, \quad u \leftarrow g^{\psi}, \quad v \leftarrow m \cdot \mathrm{pk}^{\rho}, \quad ct \leftarrow (u, v)\}$$

For each variant answer two questions: (i) is there an efficient decryption algorithm using sk and if so, show the algorithm; (ii) is the encryption scheme semantically secure? If not, describe an attack, and if so, give a reduction to the security of the semantic security of the basic scheme at the top of the page (that is, show that a semantic security attacker $\mathcal{A}$ on the modified scheme can be used to construct a semantic security attacker $\mathcal{B}$ for the original scheme). Note that semantic security is well defined even if there is no decryption algorithm.

a. Is there an efficient decryption algorithm for E_1?

Your answer:

b. Is E_1 semantically secure?

Your answer:

c. Is there an efficient decryption algorithm for E_2?

Your answer:

d. Is E_2 semantically secure?

Your answer:

e. Is there an efficient decryption algorithm for E_3?

Your answer:

f. Is E_3 semantically secure?

Your answer:

Problem 5. (16 points) Challenge-response.

In class we saw a signature-based challenge-response identification protocol that is secure against active adversaries, requires no secrets on the server, and where the challenge is short (e.g., six digits) and the response is long (a digital signature).

a. Describe a challenge-response protocol with the same properties as the one we saw in class, except that the challenge is long and the response is short (e.g., six digits). The adversary's success probability should be at most $1/2^{\ell}$, where ℓ is the response length in bits.
Hint: instead of a signature scheme, construct your protocol using a public-key encryption scheme (G, E, D), where the verifier sends the encryption of a short random challenge.

Your answer:

b. In the scheme from part (a) the verifier needs to keep a secret value between the time that it sends the challenge and the time it receives the response from the prover. Suppose we require the protocol to be *public coin* meaning that the verifier cannot keep any secrets. The signature-based challenge response protocol is an example of a public coin protocol. Show that in this case, any challenge-response protocol (two rounds) where the prover sends back an ℓ-bit message can be broken in time $O(2^{\ell})$. That is, the adversary can fool the verifier with probability 1 after performing a computation that takes time $O(2^{\ell})$. You may assume the verifier runs in constant time. Hence, if ℓ is small, the protocol cannot be secure.

Your answer: