CS255: Cryptography and Computer Security **Winter 2018**

Final Exam

Instructions:

– **Answer all five questions.**
– The exam is open book and open notes. Laptops are allowed with the network card turned off. Connecting to a network during the exam is a serious violation of the honor code.
– Students are bound by the Stanford honor code.
– You have two and a half hours.

Problem 1. Questions from all over.

a. Explain what goes wrong if we always set the IV to 0 in randomized CBC, and use the system to encrypt two different messages $m_0 \neq m_1$ with the same key k.

b. Let $H : \mathcal{M} \to \mathcal{T}$ be a collision resistant hash where $\mathcal{M} = \{0,1\}^\ell$ and $\mathcal{T} = \{0,1\}^n$. For each of the following, explain why it is collision resistant, or describe an efficient way to find collisions:

- for a fixed $0^\ell \neq \Delta \in \mathcal{M}$ define $H_1(m) := H(m) \oplus H(m \oplus \Delta)$.
- for a fixed $0^n \neq \Delta \in \mathcal{T}$ define $H_2(m) := H(m) \oplus \Delta$.

c. TLS breaks a long message sent from one party to another into a sequence of blocks called records, and encrypts each record on its own using authenticated encryption. An attacker could block the last record from getting through, making the recipient receive a truncated message. For example, an HTTP header

```
Set-Cookie:  SID=[AuthenticationToken]; secure
```

could be truncated to

```
Set-Cookie:  SID=[AuthenticationToken]
```

by blocking the last record. Suggest a way that HTTP could defend against this.

d. Briefly explain why the Internet has widely transitioned to using Diffie-Hellman over the group of points of an elliptic curve instead of the group $\mathbb{Z}_p^*$.

e. Some authenticated key exchange protocols are secure against direct attacks, while others are forward secure. Briefly explain why is it important that key exchange protocols be forward secure.

Problem 2. Encryption-based challenge-response. In class we discussed MAC-based challenge-response identification. Here we construct a different challenge-response protocol using a symmetric encryption scheme (E, D) defined over $(\mathcal{K}, \mathcal{M}, \mathcal{T})$. The secret key given to the prover and verifier is $k \xleftarrow{R} \mathcal{K}$. The protocol begins with the verifier choosing a random $m \xleftarrow{R} \mathcal{M}$ and sending $c \leftarrow E(k, m)$ to the prover. The prover decrypts c and sends back m. The verifier accepts if the response from the prover is equal to m.

a. Show that if (E, D) is (one-time) semantically secure then the protocol is secure against direct attacks. As usual this is done by arguing the contrapositive: show that if there is an efficient adversary that breaks the protocol by a direct attack then there is an efficient adversary that breaks the semantic security of (E, D).

b. Give an example (E, D) that is (one-time) semantically secure, but results in an identification protocol that is insecure against eavesdropping.

c. Explain why this protocol is secure against active attacks if (E, D) provides authenticated encryption. Recall that in an active attack the adversary first plays the role of verifier to the real prover in multiple invocations of the protocol. Then the real prover disappears and the attacker plays the role of prover who is trying to convince the verifier. The attacker wins if the verifier is convinced.

Problem 3. Signatures with message recovery. Let (G, F, F^{-1}) be a one-way trapdoor permutation defined over $\mathcal{X} := \{0,1\}^n$. Let $\mathcal{R} := \{0,1\}^\ell$ and $\mathcal{U} := \{0,1\}^{n-\ell}$, for some $0 < \ell < n$. Let $H : \mathcal{M} \times \mathcal{U} \to \mathcal{R}$ be a hash function, and let $W : \mathcal{R} \to \mathcal{U}$ be another hash function. Consider the following signature scheme (G, S, V) for signing messages $(m_0, m_1) \in \mathcal{M} \times \mathcal{U}$:

$$S\big(\text{sk},\ (m_0, m_1)\big) := \Big\{ h \leftarrow H(m_0, m_1), \quad \sigma \leftarrow F^{-1}\big(\text{sk},\ h \,\|\, (W(h) \oplus m_1)\big), \quad \text{output } \sigma \Big\}.$$

a. Explain how the verification algorithm works.

b. One can show that this scheme is secure assuming (G, F, F^{-1}) is one-way, $1/|\mathcal{R}|$ is negligible, and H and W are modeled as random oracles. However this signature scheme has another interesting property. Show that just given the public key and (m_0, σ), where σ is a valid signature on the message (m_0, m_1), it is possible to recover m_1 efficiently.

A signature scheme that has this property is called a *signature with message recovery.* It lets the signer send shorter transmissions: the signer need only transmit (m_0, σ) and the recipient can recover m_1 by itself. This can somewhat mitigate the cost of long signatures with RSA.

Problem 4. Carter-Wegman MAC. In class we discussed CBC-MAC, PMAC and HMAC. Another important family of MACs is called Carter-Wegman MACs.

a. A one-time MAC is a MAC that is secure as long as the MAC key is only used to authenticate at most one message. Write out the security definition for a one-time MAC by suitably adapting the security definition for a (many time) MAC.

b. Let p be a prime so that $1/p$ is negligible. Here is a simple candidate one-time MAC with message space $\mathcal{M} := (\mathbb{Z}_p)^{\leq \ell}$, for some $\ell < p$, and key space $\mathcal{K} := \mathbb{Z}_p^2$:

$$S\big((k,k'),\ \mathbf{m} = (m_1,\ldots,m_n)\big) := \left\{ \text{output } t \leftarrow k' + \sum_{i=1}^{n} m_i \cdot k^i \in \mathbb{Z}_p \right\}.$$

Verification $V\big((k,k'),\mathbf{m},t\big)$ works by checking that $S\big((k,k'),\mathbf{m}\big) = t$. Show that this MAC is insecure as a one-time MAC.

Hint: use the fact that the MAC can be used to sign messages of varying lengths.

c. We can fix the problem from part (b) by defining:

$$S'\big((k,k'),\ \mathbf{m} = (m_1,\ldots,m_n)\big) := \left\{ \text{output } t \leftarrow k' + k^{n+1} + \sum_{i=1}^{n} m_i \cdot k^i \in \mathbb{Z}_p \right\}.$$

Verification V' works as before by recalculating $S'\big((k,k'),\ \mathbf{m}\big)$. This MAC can be shown to be one-time secure whenever ℓ/p is negligible, but we will leave that for another day. Instead, show that this MAC is not two-time secure.

Note: this one-time MAC is blindingly fast, requiring only one addition and one multiplication per message block.

d. We can convert (S', V') into a many-time MAC using a secure PRF. Let F be a secure PRF defined over $(\mathcal{K}, \mathbb{Z}_p, \mathbb{Z}_p)$. Define the Carter-Wegman MAC as

$$S''\big((k,k'),\ \mathbf{m}\big) := \left\{ r \xleftarrow{R} \mathbb{Z}_p,\quad t \leftarrow F(k',r) + k^{n+1} + \sum_{i=1}^{n} m_i \cdot k^i,\quad \text{output } (r,t) \right\}.$$

Note that the PRF (typically AES) is only applied to the single block r. As a result, this MAC can be faster than CBC-MAC. Explain how the verification algorithm V'' works.

e. Show that the MAC (S'', V'') becomes insecure after authenticating $O(\sqrt{p})$ messages with a single key (k, k'). Specifically, the adversary can forge message-tag pairs with probability at least $1/2$.

Problem 5. Oblivious transfer from ElGamal encryption. Alice runs an online bookstore with books $m_1, \ldots, m_n \in \mathcal{M}$. Bob wants to read book number $1 \leq i \leq n$. He pays Alice for the book and wants to download the book (assume the price is the same for all books). However, he does not want to reveal to Alice what book he is interested in. Similarly, Alice wants to make sure Bob gets exactly one book. This problem is called *oblivious transfer*. They need a protocol that reveals m_i (and nothing else) to Bob, and reveals nothing to Alice.

They decide on the following protocol that uses a group $\mathbb{G}$ of prime order q with generator g:

- Alice sends a random $v \xleftarrow{\text{R}} \mathbb{G}$ to Bob,
- Bob chooses $\alpha \xleftarrow{\text{R}} \mathbb{Z}_q$ and sends $u \leftarrow g^{\alpha} v^{-i} \in \mathbb{G}$ to Alice,
- For $j = 1, \ldots, n$ Alice encrypts book m_j using the ElGamal public-key $u_j \leftarrow uv^j$ to obtain an ElGamal ciphertext c_j. She sends all n ElGamal ciphertexts $c_1, \ldots, c_n$ to Bob.

a. Explain how Bob can recover m_i from the data it receives from Alice.
Hint: there is something special about the public-key u_i.

b. Explain why Alice learns nothing about i.

c. Let's argue that Bob learns nothing other than m_i. First, suppose there is an efficient algorithm $\mathcal{A}$ that takes as input $g^r \in \mathbb{G}$ and outputs $g^{(r^2)} \in \mathbb{G}$. Show that $\mathcal{A}$ can be used to solve the CDH problem in $\mathbb{G}$. That is, construct an efficient algorithm $\mathcal{B}$ that takes as input $(g^a, g^b) \in \mathbb{G}^2$ and outputs g^{ab}. This means that if CDH is hard in $\mathbb{G}$, then so is the squaring problem.
Hint: use the relation $(a+b)^2 = a^2 + 2ab + b^2$. Your algorithm $\mathcal{B}$ will use algorithm $\mathcal{A}$ as a subroutine.

d. Now, suppose Bob sends a maliciously crafted u so that he can later learn something about two books m_{i_1} and m_{i_2}, for some $i_1 \neq i_2$. One can show that this means that Bob has an efficient algorithm $\mathcal{A}'$ for the following problem: $\mathcal{A}'$ takes as input random $(v,\ c_1 = g^{r_1},\ c_2 = g^{r_2}) \in \mathbb{G}^3$ and outputs (u, i_1, i_2, w_1, w_2) where $w_1 = (uv^{i_1})^{r_1}$ and $w_2 = (uv^{i_2})^{r_2}$. Show that $\mathcal{A}'$ can be used to solve the squaring problem in $\mathbb{G}$. That is, construct an efficient algorithm $\mathcal{B}'$ that takes as input $g^a \in \mathbb{G}$ and outputs $g^{(a^2)}$.

You just showed that if squaring is hard in $\mathbb{G}$ (which follows from CDH by part (c)), then $\mathcal{A}'$ cannot exist and therefore Bob cannot gain information about two books.

Hint: Given random g^a as input, your algorithm $\mathcal{B}'$ sets $v \leftarrow g^a$ and sets $c_1 \leftarrow v^{s_1} = g^{as_1}$ and $c_2 \leftarrow v^{s_2} = g^{as_2}$ for random $s_1, s_2 \xleftarrow{\text{R}} \mathbb{Z}_q$. It runs $\mathcal{A}'$ on (v, c_1, c_2) and obtains (u, i_1, i_2, w_1, w_2) where $w_1 = (uv^{i_1})^{as_1}$ and $w_2 = (uv^{i_2})^{as_2}$. Use the output from $\mathcal{A}'$ to compute $g^{(a^2)}$.