Spring 2020 Cryptography and Network Security

Midterm

Time: 14:20-17:20 (180 minutes) May 12, 2020

This is a 3-hour open-book exam. There are 6 questions worth of 109 points. The maximum points is 100. Please write clearly and concisely; avoid giving irrelevant information. Submit one PDF file named `exam_{student_id}.pdf` (in lowercase).

Problem 1: Crypto Questions (22 points)

1. (3 points) Please briefly explain why MAC-then-Encrypt is a bad idea.
2. (3 points) Suppose $H_1(\cdot)$ is a collision-resistant hash function, but $H_2(\cdot)$ is not. H_1 and H_2 are both constructed using the Merkle-Damgard construction. One might try patching the insecure H_2 by wrapping it with H_1: $H'(m) = H_1(H_2(m||c_2)||c_1)$, where c_1 and c_2 are constant. Is $H'(\cdot)$ collision-resistant? Why or why not?
3. (3 points) One might expect that if encrypting a message once is secure, then encrypting it twice should also be secure. This unfortunately is not always true. Let $\mathcal{E} = (E, D)$ be a cipher, and $\mathcal{E}_2 = (E_2, D_2)$ is another cipher where $E_2(k, m) = E(k, E(k, m))$. Show that there is a semantically secure cipher $\mathcal{E}$ such that $\mathcal{E}_2$ is not semantically secure. Please provide an example and briefly justify your answer.
4. We learned about several computational problems and the corresponding computational hardness assumptions in class. For example, DDH is the decisional Diffle-Hellman problem and CDH is the computational Diffle-Hellman problem. And the DDH assumption implies the CDH assumption. Please show the relationship ($\Rightarrow$, $\Leftarrow$, or $=$) between the following pairs. Briefly explain your answer.
   All of them operate in the same cyclic group G with an order q and a generator g. We define the *CDH-square problem* to be computing $g^{a^2} \in G$ given $g^a \in G$, where $a \xleftarrow{R} Z_q$. The *CDH-square assumption* says the CDH-square problem is computationally hard.
   (a) (3 points) the DDH assumption and the discrete logarithm assumption
   (b) (5 points) the CDH assumption and CDH-square assumption
5. (5 points) Bob has two cups that are idential in shape. However, since Bob suffers from red-green color blindness, Bob cannot tell whether the two cups are the same color or are red and green. Alice claims that these two cups have different colors. Please design a proof strategy for Bob to examine whether Alice's claim is true or not without relying on others or any special equipment.

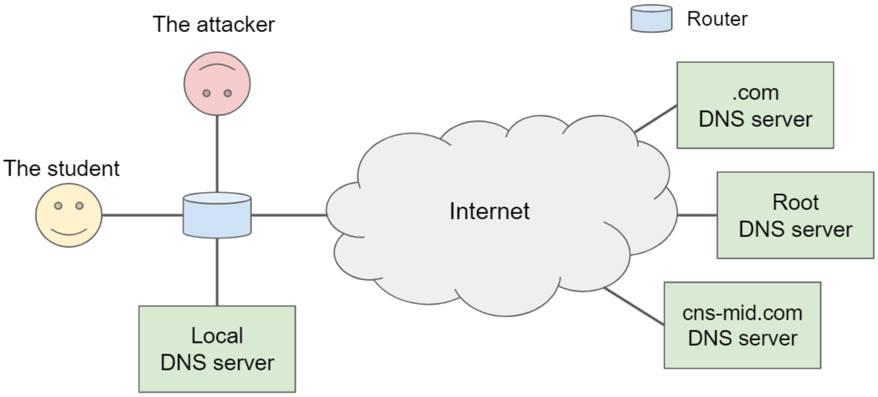


Figure 1: Overview

Problem 2: DNS (15 points)

Please answer the following questions according to Figure 1.

1. (3 points) Please describe the process of the student obtaining the IP for `www.cns-mid.com`. Assume that the student *never* connects to the server before.
2. (5 points) You are the attacker shown in Figure 1, trying to launch a *DNS Cache Poisoning attack* by sending spoofed responses to the local DNS server (also called local recursive nameserver in class). Your goal is to replace the IP for `www.cns-mid.com` in the cache of the local DNS server. In order to let the local DNS server accept your DNS responses, you need to guess the correct Query ID.
   The question is: How many guesses can you make? Your spoofed response needs to be accepted during a narrow time window—between the client sends a DNS query about `www.cns-mid.com` to the local DNS server, and the local DNS server receives the legitimate DNS response. Here are some assumptions:
   - There is no cache data in any DNS servers.
   - Each link (black lines) takes 3 ms to transmit a DNS message.
   - It takes 16 ms to transmit a DNS message through the Internet (gray area).
   - There is no packet processing delay in any router (blue component) and DNS servers.
   - The attacker can send one DNS message every ms.
3. (4 points) You realize that *the Kaminsky attack* is more reliable than just guessing the Query ID, please describe the process of this attack step by step.
4. (3 points) Please explain the reason why *the Kaminsky attack* is more powerful than the attack mentioned in Question 2.

Problem 3: Cut Mix (15 points)

A mix network is a technique designed to prevent a passive global attacker (who can observe all traffic between nodes in the mix network) from knowing the identity of a message's sender

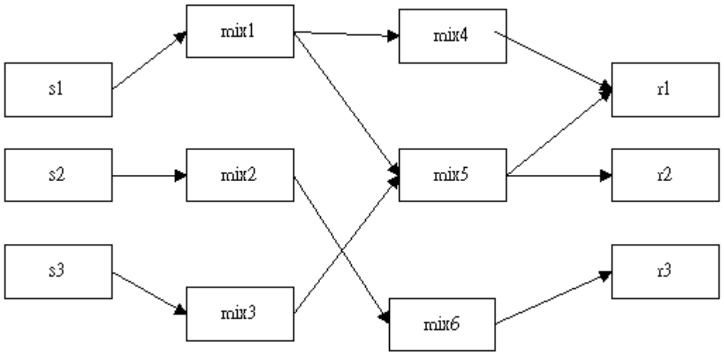


Figure 2: An example of a cut mix network

and receiver. For simplicity, we assume that **each sender sends exactly one message per round in this problem**.

Although a mix network is designed to defend against traffic analysis by a passive global attacker, in practice, the attacker may be able to derive useful information under certain conditions. For example, as we saw in Homework 2, given three senders and three receivers, and each sender only sends one message per round, the attacker observing $\{s_1, s_2, s_3\} \Rightarrow \{r_1, r_2, r_2\}$ will be able to infer that two of the senders send messages to r_2 and no one sends to r_3.

To reduce the amount of information leak, someone came up with an idea called *Cut Mix Network*. The goal of Cut Mix Network is to "confuse" the attacker such that the attacker cannot correctly determine the number of messages received by each receiver. For instance, if the attacker cannot tell whether the receiver (multi-)set is $\{r_1, r_1, r_2\}$ or $\{r_1, r_2, r_2\}$, then we say the attacker is confused.

A Cut Mix Network performs additional operations on top of a normal mix network:

1. Each message is divided and into several chunks in the mix network. All messages and chunks are padded into the same size.

2. Chucks of a message will all traverse different paths to the receiver.

3. The sender decides the number of chunks and the cutting position in the mix network.

Figure 2 shows an example of a cut mix network, in which the message of s_1 is divided into two chunks at mix1.

(a) (5 points) Can Cut Mix Network confuse the attacker from getting the correct receiver set when each receiver gets a message from only one sender? Why or why not? (For example, s_1 sends to r_1, s_2 sends to r_2 and s_3 sends to r_3)

(b) (5 points) Can Cut Mix Network confuse the attacker from getting the correct receiver set when at least one receiver gets messages from multiple senders? Why or why not? (For example, s_1 sends to r_1, s_2 sends to r_2 and s_3 sends to r_2)

(c) (5 points) What are the advantages (or disadvantages) of Cut Mix Network compared to the normal Mix Network? Please list at least two.

Problem 4: Proof of Location (22 points)

Companies worldwide are proposing new technologies to help fight the coronavirus pandemic.

PlantFlower Telecom plans to develop a *proof-of-location and proof-of-absence* service by which its customer can obtain a publicly verifiable proof showing s/he visited or did not visit a region (specifically, connected to a cell tower c) during a time interval t. A customer can then show the proof to others, for example, for proving not visiting any crowded destination during holidays.

You can assume everyone trusts PlantFlower Telecom and knows its public key PK. For each c and t, PlantFlower Telecom keeps a list of customers (denoted by $U(c,t)$) who have ever connected to the cell tower c during the time interval t. PlantFlower Telecom has a secure channel to every customer and can validate its customer's identity over the Internet.

Please answer the following questions.

1. (6 points) PlantFlower Telecom drafts a proposal using Merkle hash trees:
For each cell tower c and time interval t, PlantFlower Telecom computes a Merkle tree over all customers $u \in U(c,t)$. Specifically, each leaf node of the Merkle tree is the hash of a customer phone number, ordered by the hash values from left to right. The root of the Merkle tree is signed and published on PlantFlower Telecom's website.
Please help the company completes this proposal by describing the construction of a proof-of-location (u visited c during t) and a proof-of-absence (u did not visit c during t), respectively. Note that the proofs should be publicly verifiable.

2. (6 points) In addition to the above proposal using Merkle hash trees, two other proposals are also considered by PlantFlower Telecom:

   (a) PlantFlower Telecom provides a daily signed report for each customer u. The report contains a list of cell towers u has connected to (i.e., $\{c | u \in U(c,t) \text{ for } t \text{ in a day}\}$) in a day.

   (b) Customers request for proofs on demand. A customer u can send a request consisting a cell tower c and a time interval t, then PlantFlower Telecom will return a signed message "u was at c in t" or "u was not at c in t" depending on whether $u \in U(c,t)$ or not.

   Please compare the three proposals with respect to their performance and privacy leak to verifiers. Please clearly state additional assumptions (if any) you make while answering this.

3. (10 points) The government would like to know the traces of diagnosed patients in order to issue warnings. However, the government does not want to reveal the identity of the diagnosed patients to PlantFlower Telecom for the sake of privacy. On the other hand, PlantFlower Telecom refuses to provide the traces of all customers to the government in fear of large-scale surveillance. They came up with a solution based on ElGamal encryption, such that the government can obtain the trace of the ith customer, without revealing the value i to PlantFlower Telecom. For simplicity, let's assume there are n customers, and $i \in \{1 \cdots n\}$.
Consider a cyclic group G of order q with generator g.

1. PlantFlower: $x \overset{R}{\leftarrow} G$
2. PlantFlower → Government: x
3. Government: $a \overset{R}{\leftarrow} Z_q$, $y \leftarrow g^a x^{-i} \in G$
4. Government → PlantFlower: y
5. PlantFlower: Let $pk_j = yx^j$. For all $j \in \{1 \cdots n\}$, encrypt the jth customer's trace m_j by ElGamal encryption using pk_j as the public key. The resulting ElGamal ciphertext is c_j.
6. PlantFlower → Government: c_j for all $j \in \{1 \cdots n\}$

(a) (4 points) Explain how the government recovers the trace of the ith customer, m_i.

(b) (3 points) Explain intuitively why PlantFlower Telecom does not know the value of i. (No formal proof is needed.)

(c) (3 points) Explain intuitively why the government does not know the traces other than it of the ith customer. (No formal proof is needed.)

Problem 5: Key Exchange with KDC (15 points)

You have learned how to perform key exchange via a Key Distribution Center (KDC): The KDC distributes a shared session key to two users so that both users can use this shared session key to encrypt their messages.

You also learned about how a key exchange protocol can be compromised if the protocol is not carefully designed in Homework 2.

We say a key-exchange protocol is secure if it can ensure

1. the freshness of the shared session key,
2. the security of the shared session key (i.e., the session key cannot be derived by any computationally-bounded attackers), and
3. more importantly, entity authentication.

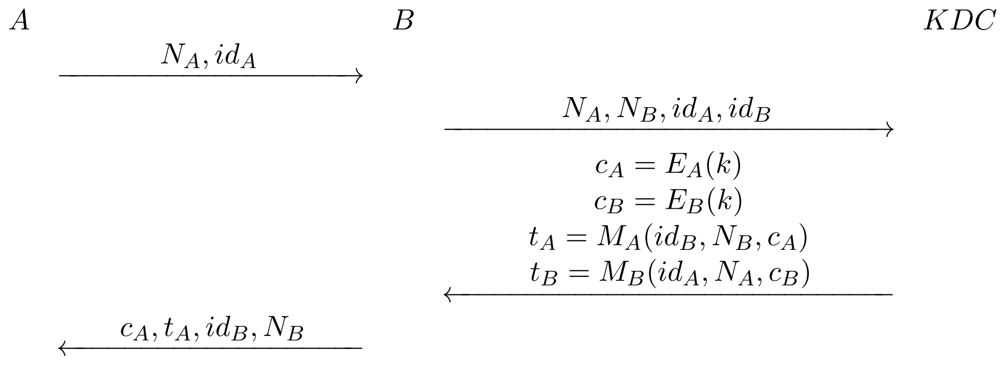
Recalling the protocol from Homework 2 Problem 7. We propose some different variations of the protocol. Can you determine whether these protocols are secure or not? You need to explain why the protocols are secure or insecure.

Notation:

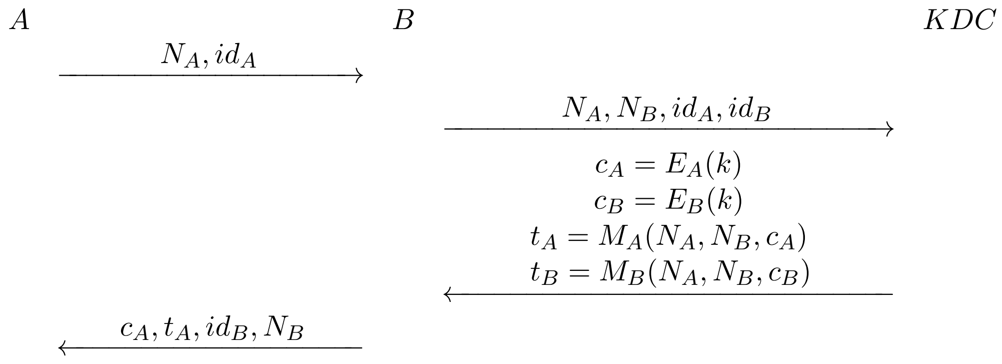
- A and B represent Alice and Bob, respectively.
- KDC represents a trusted Key Distribution Center.
- N_A and N_B denote random nonce created by A and B, respectively.
- id_A and id_B denote the identities of the users A and B, respectively.
- $E_A(m)$ and $E_B(m)$ denote encryption of message m using the shared encryption secret key between KDC and the user A and B, respectively.
- $M_A(m)$ and $M_B(m)$ denote MAC of the message m using the shared MAC secret key between KDC and the user A and B, respectively.

- k denotes the shared session key between A and B.

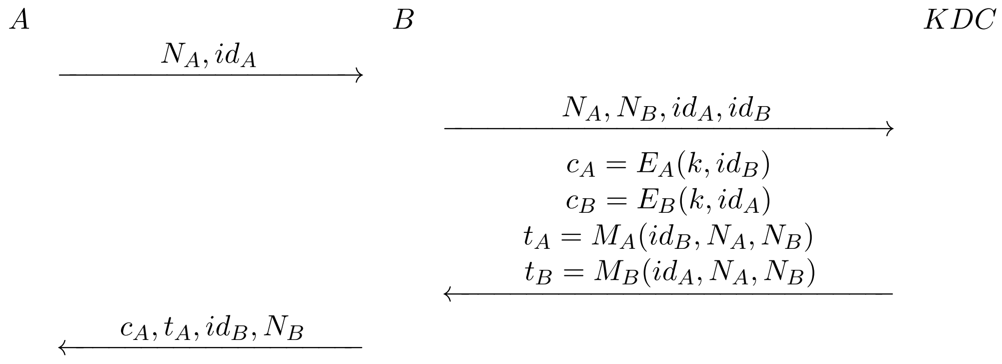
(1) (5 points)



(2) (5 points)



(3) (5 points)



Problem 6: Attack Game (20 points)

In this problem, we develop a notion of security for a cipher, called *psuedo-random ciphertext security*, which intuitively says that no efficient adversary can distinguish an encryption of a chosen message from a random ciphertext.

Let $\mathcal{E} = (E, D)$ be defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. Assume that one can efficiently generate ciphertexts from the ciphertext space $\mathcal{C}$ at random. We define an attack game between an adversary A and a challenger as follows. The adversary selects a message $m \in \mathcal{M}$ and sends m to the challenger. The challenger then computes:

$$b \overset{\mathrm{R}}{\leftarrow} \{0, 1\}, k \overset{\mathrm{R}}{\leftarrow} \mathcal{K}, c_0 \overset{\mathrm{R}}{\leftarrow} E(k, m), c_1 \overset{\mathrm{R}}{\leftarrow} \mathcal{C}, c \leftarrow c_b$$

and sends the ciphertext c to A, who then computes and outputs a bit $\hat{b}$. We define As advantage to be $|Pr[\hat{b} = b] - 1/2|$, and we say the $\mathcal{E}$ is *pseudo-random ciphertext secure* if this advantage is negligible for all efficient adversaries.

1. (10 points) Prove that if a cipher is pseudo-random ciphertext secure, then it is semantically secure.

2. (5 points) Prove that the one-time pad is pseudo-random ciphertext secure.

3. (5 points) Give an example of a cipher that is semantically secure, but not pseudo-random ciphertext secure.