Popa & Wagner Spring 2016 — CS 161 Computer Security — Midterm 2

PRINT your name: _____, _____
(last) (first)

I am aware of the Berkeley Campus Code of Student Conduct and acknowledge that academic misconduct will be reported to the Center for Student Conduct.

SIGN your name: _____

PRINT your class account login: `cs161-`_____ and SID: _____

Name of the person sitting to your left: _____ Name of the person sitting to your right: _____

You may consult two sheets of paper (double-sided) of notes. You may not consult other notes, textbooks, etc. Calculators, computers, and other electronic devices are not permitted. If you think a question is ambiguous, choose the most reasonable assumption and document your assumption clearly.

You have 90 minutes. There are 9 questions, of varying credit (100 points total). The questions are of varying difficulty, so avoid spending too long on any one question.

Do not turn this page until your instructor tells you to do so.

Problem 1 ***True or False*** **(10 points)**

Circle True or False. Do not justify your answer.

(a) TRUE or FALSE: It is safe (IND-CPA-secure) to encrypt multiple messages in CBC mode with a constant IV of 0, using the same encryption key each time.

(b) TRUE or FALSE: It is safe (IND-CPA-secure) to encrypt multiple messages in CBC mode with a constant IV of 0, as long as the encryption key is different for each message sent.

(c) TRUE or FALSE: Encrypting a message with CBC mode protects the integrity of the message.

(d) TRUE or FALSE: It's ok for multiple people using El Gamal public key encryption to use the same modulus p.

(e) TRUE or FALSE: It's ok for multiple people using RSA signatures to use the same modulus n.

Problem 2 ***More True or False*** **(8 points)**

In this question, H refers to a secure cryptographic hash function and $\text{len}(x)$ is a 128-bit int storing the length of x. You can assume that x and y are at most one million bytes long. Circle True or False. Do not justify your answer.

(a) TRUE or FALSE: Let $F(x, y) = H(x||y)$. Given x, y, and $F(x, y)$, it is easy for an attacker to find x', y' such that $F(x', y') = F(x, y)$ and $x \neq x'$.

(b) TRUE or FALSE: Let $F(x, y) = H(\text{len}(x)||x||y)$. Given x, y, and $F(x, y)$, it is easy for an attacker to find x', y' such that $F(x', y') = F(x, y)$ and $x \neq x'$.

(c) TRUE or FALSE: Let $F(x, y) = H(\text{len}(y)||x||y)$. Given x, y, and $F(x, y)$, it is easy for an attacker to find x', y' such that $F(x', y') = F(x, y)$ and $x \neq x'$.

(d) TRUE or FALSE: Let $F(x, y) = H(x||\text{len}(x)||y)$. Given x, y, and $F(x, y)$, it is easy for an attacker to find x', y' such that $F(x', y') = F(x, y)$ and $x \neq x'$.

Problem 3 ***Multiple choice*** **(6 points)**

Circle all the options that apply.

(a) When using the CBC block chaining mode, if the IV is modified by an attacker during transmission (so the correct IV was used during encryption, but the receiver receives the modified value), the recipient can still successfully decrypt:

1. the entire ciphertext
2. none of the ciphertext
3. only the first block
4. all but the first block
5. other set of blocks than above

(b) Which of the following are properties of CTR mode?

1. encryption can be parallelized
2. decryption can be parallelized
3. the nonce does not have to be random, as long as it never repeats
4. it turns a block cipher into a stream cipher
5. it is more secure than CBC mode
6. it provides integrity and authentication for the message

Problem 4 ***TLS*** **(16 points)**

An attacker is trying to attack the company Wahoo and its users. Assume that users always visit Wahoo's website with an HTTPS connection, using RSA and AES encryption (no Diffie-Hellman). (You may assume that Wahoo does not use certificate pinning—if you don't know what that is, you can ignore it.) For each of the following attack scenarios, circle all of the options that an attacker could achieve in that attack scenario.

(a) If the attacker obtains a copy of Wahoo's certificate, the attacker could:

1. impersonate the Wahoo web server to a user
2. discover some of the plaintext of data sent during a past connection between a user and Wahoo's website
3. discover all of the plaintext of data sent during a past connection between a user and Wahoo's website
4. replay data that a user previously sent to the Wahoo server over a prior HTTPS connection
5. none of the above

(b) If the attacker obtains the private key of a certificate authority trusted by users of Wahoo, the attacker could:

1. impersonate the Wahoo web server to a user
2. discover some of the plaintext of data sent during a past connection between a user and Wahoo's website
3. discover all of the plaintext of data sent during a past connection between a user and Wahoo's website
4. replay data that a user previously sent to the Wahoo server over a prior HTTPS connection
5. none of the above

(c) If the attacker is a man in the middle on a HTTPS connection between a user and Wahoo's website, the attacker could:

1. impersonate the Wahoo web server to this user
2. discover some of the plaintext of data sent during *this* connection
3. discover all of the plaintext of data sent during *this* connection
4. discover all of the plaintext of data sent during a *past* connection between a user and Wahoo's website
5. replay data that a user previously sent to the Wahoo server over a prior HTTPS connection
6. none of the above

(d) Suppose the attacker obtains the private key that was used by Wahoo's server during a past connection between a victim and Wahoo's server, but not the current private key. Also, assume that the certificate corresponding to the old private key has been revoked and is no longer valid. This attacker could:

1. impersonate the Wahoo web server to this user
2. discover all of the plaintext of data sent during a *current* connection (one where the current private key is used) between a user and Wahoo's website
3. discover all of the plaintext of data sent during a *past* connection (one where the old private key was used) between a user and Wahoo's website
4. none of the above

Problem 5 ***Iliad Identification Integrity scheme*** **(15 points)**

Jeff is hired by the big computer company, Iliad, to do a security review of their devices. All of Iliad's devices are assigned a unique *identification number*. Iliad's devices use the Iliad Identification Integrity (I^3) scheme to protect the integrity of identification numbers. Each of Iliad's devices has a unique key K embedded in the hardware that can only be used for verification with the I^3 scheme and can't be extracted by any means. In the I^3 scheme, when the device is manufactured, it does the following:

1. Generate a random 16-byte identification number N. Output N via a direct physical link to the factory machine.
2. Generate a random 16-byte IV.
3. Define the plaintext P to be N followed by 16 zero-bytes. Encrypt P with AES-CBC using the IV and device's embedded key K, resulting in ciphertext C.
4. Store N, IV, and C on the device's flash storage.

When Iliad's products are powered up, they will obtain N as follows:

1. Read IV and C from flash storage.
2. Decrypt C using IV and embedded key K, to get the plaintext P'.
3. If the last 16 bytes of P' are 0 and the first 16 bytes of P' match the N stored in flash, return N. Otherwise, signal an error.

The flash storage has no other protections, and an attacker could potentially tamper with the data stored on flash.

Answer the following questions.

(a) Jeff has a hunch that the scheme does not actually protect the integrity of N. Briefly, what's the reason for Jeff's hunch? One sentence should be enough.

(b) Upon further investigation, Jeff determines that the I^3 scheme is in fact insecure. Describe how you can modify N, IV and C to have the verification process accept at least one other N.

Modified value of $N =$

Modified value of $IV =$

Modified value of $C =$

(c) Jeff now needs to recommend a change to the scheme to make it secure, so that Iliad won't lose face. Describe a change to the I^3 scheme so that it will actually protect the integrity of N.

Problem 6 ***Name confidentiality in Project 2*** **(12 points)**

Jamie is working on Project 2 (Part 1) and had several ideas for how to keep the filename secret. Jamie will store the file at id i, where i is computed from the filename n in some way. For each approach listed below, circle either "secure" or "broken" according to whether it would meet the requirements for filename confidentiality from Project 2 or not.

You can assume CBC mode encryption uses a symmetric key that is generated and stored securely and not known to the attacker. H represents the SHA256 cryptographic hash function. Don't justify your answer.

(a) SECURE or BROKEN: $i = n$.

(b) SECURE or BROKEN: $i = H(n)$.

(c) SECURE or BROKEN: i is a RSA signature on n, using Jamie's private RSA key.

(d) SECURE or BROKEN: Encrypt n using AES in CBC mode, with all-zeros IV, and use the resulting ciphertext (excluding the IV) as i.

(e) SECURE or BROKEN: Encrypt n using AES in CBC mode, with all-zeros IV, and use the last 16 bytes of the resulting ciphertext as i.

(f) SECURE or BROKEN: Encrypt n using AES in CBC mode, with all-zeros IV, and let j denote the resulting ciphertext. Use $i = H(j)$.

Problem 7 ***Computing on encrypted data*** **(10 points)**

A cool property of some encryption schemes is that they allow you to compute on encrypted data! Let $\mathsf{Enc}(M)$ denote the encryption of message M. Given C_1 and C_2 (the ciphertexts for messages M_1 and M_2), anyone (even without the decryption keys) can compute a ciphertext C_3 that will decrypt to the product $M_1 \times M_2$.

The idea is that we want a server in the cloud to do some work for us, but we don't want the cloud to see our data. So we give encrypted data to the cloud (without giving the decryption key to the cloud), and the cloud gives us back the encrypted computation result. Let's figure out how the cloud can perform this computation.

(a) Recall the El Gamal scheme: The El Gamal public key is (p, g, h), where x is the private key and $h = g^x \bmod p$. The encryption of a message M is $\mathsf{Enc}(M) = (g^r \bmod p, M \times h^r \bmod p)$, for a random r.

Given only $C_1 = \mathsf{Enc}(M_1) = (s_1, t_1)$ and $C_2 = \mathsf{Enc}(M_2) = (s_2, t_2)$ and the El Gamal public key, show how the cloud can compute a ciphertext $C_3 = \mathsf{Enc}(M_1 \times M_2 \bmod p)$. In other words, show how the cloud can compute a ciphertext C_3 that will decrypt to $M_1 \times M_2 \bmod p$. Show an equation that the cloud can use to compute C_3:

$C_3 =$

(b) Suppose the cloud has two plaintext values u_1, u_2, each a 1024-bit number. Suppose the cloud also has two ciphertexts $C_1 = \mathsf{Enc}(x_1)$ and $C_2 = \mathsf{Enc}(x_2)$ that were computed using RSA encryption, and the cloud knows the RSA public key, but the cloud doesn't know x_1 or x_2 or the RSA private key. How can the cloud compute a ciphertext $C_3 = \mathsf{Enc}(x_1^{u_1} \times x_2^{u_2} \bmod n)$ efficiently? You don't need to justify your answer or explain why your solution works.

Problem 8 ***Protocol analysis*** **(15 points)**

Alice and Bob want to simulate flipping a fair coin. Ideally, each of them would like to be sure that the other can't "cheat" and force the coin to be heads or tails.

For each of the following schemes, determine whether the scheme is secure or not and then circle "Secure" or "Broken". If you circle secure, you don't need to justify your answer. If you circle broken, give the attack that either Alice or Bob would mount to give them better than 50% chance of obtaining heads.

In the following, let p be a 2048-bit prime number, g a generator modulo p, and H a secure cryptographic hash function; these are fixed in advance and known to everyone. You can assume that both parties complete the protocol. That is, neither party will refuse to finish the protocol.

(a) Alice randomly picks a such that $0 < a < p$ and sends $g^a \bmod p$ to Bob. Then, Bob randomly picks b such that $0 < b < p$ and sends $g^b \bmod p$ to Alice. Then, both compute $g^{ab} \bmod p$. If $g^{ab} \bmod p$ is even, the coin flip is heads; if odd, the coin flip is tails.

SECURE or BROKEN

Attack (if you circled "Broken"):

(b) Alice randomly picks a such that $0 < a < p$ and sends $g^a \bmod p$ to Bob. Then, Bob randomly picks b such that $0 < b < p$ and sends b to Alice. Then, Alice sends a to Bob and Bob checks that it matches what Alice sent earlier. If $H(a||b)$ is even, the coin flip is heads; if odd, the coin flip is tails.

SECURE or BROKEN

Attack (if you circled "Broken"):

(c) Alice randomly picks a to be either 0 or 1 and sends $H(a)$ to Bob. Then, Bob randomly picks b to be either 0 or 1 and sends b to Alice. Then, Alice reveals a to Bob and Bob checks that it matches what Alice sent earlier. If both picked 0 or both picked 1, the coin flip is heads. If one picked 0 and the other picked 1, the coin flip is tails.

SECURE or BROKEN

Attack (if you circled "Broken"):

(d) Alice randomly picks a such that $0 < a < 2^{128}$ and sends $H(a)$ to Bob. Then, Bob randomly picks "even" or "odd" and sends that to Alice. Then, Alice reveals a to Bob and Bob checks that it matches what Alice sent earlier. If Bob's guess about a was right (e.g., Bob picked "even" and a is even, or Bob picked "odd" and a is odd), the coin flip outcome is heads, otherwise it is tails.

SECURE or BROKEN

Attack (if you circled "Broken"):

(e) Alice randomly picks a such that $0 < a < 2^{128}$ and randomly picks an AES key k. Alice computes $c = E_k(a)$ [the AES-CBC encryption of a, under a random IV; c includes the IV] and sends c to Bob. Then, Bob randomly picks "even" or "odd" and sends that to Alice. Then, Alice reveals a and k to Bob and Bob checks that it matches what Alice sent earlier. If Bob's guess about a was right (e.g., Bob picked "even" and a is even, or Bob picked "odd" and a is odd), the coin flip outcome is heads, otherwise it is tails.

SECURE or BROKEN

Attack (if you circled "Broken"):

Problem 9 ***RSA keypairs*** **(8 points)**

You want to generate a RSA keypair and store the private key on two different servers, one in New York and the other in California.

You have a high-security key storage device that can store up to 150 bits of secret key material, and will be highly resistant to tampering or reverse engineering. Corporate policy says that secret key material must not leave those two servers, with the sole exception that it can be copied between a server and the key storage device via USB. You are not allowed to transport or communicate any secret key material except on this key storage device. (For instance, storing a copy of your private key on your laptop's hard drive is prohibited. Sending your private key over the Internet is prohibited, even if it is encrypted before transmission.) The key storage device must never leave your personal possession. You have only one key storage device and can only afford to take one trip from California to New York.

Describe how you can securely generate such a keypair and install it on both servers, with only one trip from California to New York.