Homework 3

Q1 True or False: Cryptography

7 Points

Q1.1

1 Point

If the daily lottery numbers are truly random, then they can be used as the entropy for a one-time-pad since a one-time-pad needs to be random.

- ○ True
- ◉ False

EXPLANATION

False, since the information is public.

✔ **Correct**

Save Answer Last saved on **Feb 23 at 5:09 PM**

Q1.2

1 Point

Suppose there is a transmission error in a block B of ciphertext using CBC mode. This error propagates to every subsequent block in decryption, which means that the block B and every block after B cannot be decrypted correctly.

- ○ True
- ◉ False

EXPLANATION

False. Only B and the block after B are decrypted incorrectly.

✔ Correct

Save Answer Last saved on **Feb 23 at 5:09 PM**

Q1.3

1 Point

The IV for CBC mode must be kept secret.

- ○ True
- ◉ False

EXPLANATION

False. It can be public. For instance, it is normally sent in the clear along with the ciphertext, so any eavesdropper can see the IV---this does not cause any security problems.

✔ Correct

Save Answer Last saved on **Feb 23 at 5:10 PM**

Q1.4

1 Point

Alice and Bob share a symmetric key k. Alice sends
Bob a message encrypted with k stating, "I owe you $100", using AES-CBC encryption. Assuming AES is secure, we can be confident that an active attacker cannot tamper with this message; its integrity is protected.

- ○ True
- ◉ False

EXPLANATION

False. An attacker can still modify the ciphertext sent, and there is no way for Bob to tell if the message has been modified.

✔ Correct

Save Answer Last saved on **Feb 23 at 5:10 PM**

Q1.5

1 Point

Alice and Bob share a secret symmetric key k which they use for calculating MACs. Alice sends the message $M =$ "I, Alice, owe you, Bob, $100" to Bob along with its message authentication code $\mathrm{MAC}_k(M)$. Bob can present $(M, \mathrm{MAC}_k(M))$ to a judge as proof that Alice owes him $100 since a MAC provides integrity.

- ○ True
- ◉ False

EXPLANATION

False. A MAC provides integrity, but not does not prove that Alice generated the MAC. Bob can create MACs himself and so that does not prove that Alice wrote the message.

✔ Correct

Save Answer Last saved on **Feb 23 at 5:10 PM**

Q1.6

1 Point

The random number r in El Gamal can be made public.

- ○ True
- ◉ False

EXPLANATION

False. If it is public, an attacker can calculate $B^{-r}c_2$ (since B is public).

✔ Correct

Save Answer Last saved on **Feb 23 at 5:10 PM**

Q1.7
1 Point

It is okay if multiple people use the same modulus p for their El Gamal public key.

- (●) True
- ( ) False

EXPLANATION

True, since p is public and known.

✔ Correct

Save Answer Last saved on **Feb 23 at 5:10 PM**

Q2 Hashing Functions
4 Points

Recall the definition of "one-way functions" and "collision-resistance" from lecture.

- We say a function f is one-way if given $f(x)$ it is hard to find x' such that $f(x') = f(x)$.
- We say a function f is “collision-resistant” if it is hard to find two inputs x, y such that $f(x) = f(y)$ but $x \neq y$.

For each of the given functions H below, determine if it is one-way or not, and if it is collision-resistant or not.

Q2.1
1 Point

Select if $H(x) = x$ is:

- ( ) One way
- (●) Collision resistant
- ( ) Both
- ( ) None

EXPLANATION

This function is collision-resistant because given two different inputs, $x' \neq x$, the hashes $H(x) = x$ and $H(x') = x'$ are always different.

This function is not one-way because given $H(x)$, we can use it directly as the input to the hash function to get $H(H(x)) = H(x)$.

✔ Correct

Save Answer Last saved on **Feb 23 at 5:10 PM**

Q2.2

1 Point

Select if $H(x) = x \bmod 2$ is:

- ○ One way
- ○ Collision resistant
- ○ Both
- ◉ None

EXPLANATION

This function is not collision-resistant. Consider $H(0) = H(2) = 0$.

This function is not one-way because given $H(x) = 0$, we know any even value of x will satisfy $H(x) = 0$.

✔ Correct

Save Answer Last saved on **Feb 23 at 5:11 PM**

Q2.3

1 Point

Let E_k be an ideally secure block cipher with a known and published key k.

Select if $H(x) = E_k(x)$ is:

- ○ One way
- ◉ Collision resistant
- ○ Both
- ○ None

EXPLANATION

This function is collision-resistant. The output of an ideally secure block cipher is indistinguishable from a random *permutation* of bits, so two different inputs will always result in different output.

This function is not one-way because the key is known and published, so given $H(x)$, we can calculate $D_k(H(x)) = D_k(E_k(x)) = x$, and use this as the input to the hash to get $E_k(x) = H(x)$.

✔ **Correct**

Save Answer Last saved on **Feb 23 at 5:11 PM**

Q2.4

1 Point

Select if $H(x) = 0$ is:

- ○ One way
- ○ Collision resistant
- ○ Both
- ◉ None

EXPLANATION

This function is not collision-resistant. Consider $H(0) = H(1) = 0$.

This function is not one-way because given $H(x) = 0$, we know any value of x will satisfy $H(x) = 0$.

✔ **Correct**

Save Answer Last saved on **Feb 23 at 5:13 PM**

Q3 El Gamal Encryption

3 Points

Recall the definition of El Gamal encryption from lecture:

- Everyone knows a large prime p, and an integer g.
- Bob chooses a private key b, and computes a public key $B = g^b \pmod{p}$.
- To encrypt a message m, Alice generates a random r, and creates the ciphertext $(c_1, c_2) = (g^r,\ m \cdot B^r) \pmod{p}$.
- To decrypt the ciphertext, Bob calculates $c_1^{-b} c_2 \equiv m \pmod{p}$.

(Note: Since everything is $\bmod p$, we need $2 \leq g \leq p-2,\ 0 \leq b \leq p-2$, and $0 \leq r \leq p-2$.)

As mentioned in the textbook, this simplified El Gamal scheme is actually not IND-CPA secure. In this question, we'll explore some attacks on this scheme.

Q3.1

1 Point

Alice encrypts m and sends the ciphertext (c_1, c_2) to Bob. Construct a ciphertext (c_1', c_2') which is the encryption of $2m$. All computations are mod p.

$c_1' =$

- (●) c_1
- ( ) $2 + c_1$
- ( ) $2 \oplus c_1$
- ( ) $2c_1$
- ( ) c_1^2

$c_2' =$

- ○ c_2
- ○ $2 + c_2$
- ○ $2 \oplus c_2$
- ◉ $2c_2$
- ○ c_2^2

EXPLANATION

Intuitively, given ciphertext $(g^r,\ m \cdot B^r)$, if we change the ciphertext to $(g^r,\ 2m \cdot B^r)$, it will decrypt to $2m$. To do this, we should leave c_1 unchanged and multiply c_2 by 2.

Formally, construct the ciphertext $(c_1', c_2') = (c_1, 2c_2)$. The decryption of this is $c_1'^{-b} c_2' = c_1^{-b}(2c_2) = 2(c_1^{-b_1} c_2) = 2m$.

✔ Correct

Save Answer Last saved on **Feb 23 at 5:17 PM**

Q3.2

1 Point

Suppose you intercept two ciphertexts $(g^{r_1}, m_1 B^{r_1})$ and $(g^{r_2}, m_2 B^{r_2})$ that Alice has encrypted for Bob. Assume they are encryptions of some unknown messages m_1 and m_2.

Construct a ciphertext (c_1, c_2) which is a valid El Gamal encryption of the message
$m_1 m_2$. All computations are mod p.

$c_1 =$

- ○ $g^{r_1} B^{r_1}$
- ○ $g^{r_1} m_1$
- ◉ $g^{r_1 + r_2}$
- ○ $g^{r_1 * r_2}$

$c_2 =$

- $m_1 m_2$
- $B^{r_1+r_2}$
- $g^{r_1+r_2} B^{r_1+r_2}$
- $m_1 m_2 B^{r_1+r_2}$

EXPLANATION

The approach from the previous part doesn't quite work here, because we only know $m_1 B^{r_1}$ and $m_2 B^{r_2}$, not m_1 or m_2. Since we want the encryption of $m_1 m_2$, intuitively we might try multiplying the two values that we know involve m_1 and m_2. This gives $c_2 = (m_1 B^{r_1})(m_2 B^{r_2}) = m_1 m_2 B^{r_1+r_2}$.

Intuitively, in this new ciphertext, the exponent of B is $r = r_1 + r_2$, so c_1 should be $g^r = g^{r_1+r_2}$.

Formally, the decryption of the new ciphertext will be $c_1^{-b} c_2 = c_1^{-b}\ m_1 m_2 B^{r_1+r_2}$. We want this to be $m_1 m_2$, so we can write the equation $c_1^{-b}\ m_1 m_2 B^{r_1+r_2} = m_1 m_2$.

Divide both sides by $m_1 m_2$:
$c_1^{-b}\ B^{r_1+r_2} = 1$

Multiply both sides by c_1^b:
$c_1^b = B^{r_1+r_2}$

Use $B = g^b$ from definition of El Gamal:
$c_1^b = g^{b(r_1+r_2)}$

$c_1 = g^{r_1+r_2}$

✔ Correct

Save Answer Last saved on **Feb 23 at 5:17 PM**

Q3.3

1 Point

Consider a new scheme where the value r is not generated randomly every time. Instead, Alice randomly generates an initial value r_0, and then

increments r_0 by 1 every time she needs to encrypt another message. Is this encryption scheme IND-CPA secure?

- ○ Yes
- ◉ No

Suppose Alice encrypts m_0 and then encrypts m_1 immediately after, both using the scheme above. Which of the following values can an adversary obtain, given knowledge of these two ciphertexts?

- ○ m_0
- ○ m_1
- ○ r_0
- ○ $m_0 m_1$
- ○ $m_0 + m_1$
- ○ $m_0 - m_1$
- ◉ m_0 / m_1
- ○ None of the above

EXPLANATION

First part

This scheme is not IND-CPA. Intuitively, this variant is weaker than plain El Gamal from lecture, since it remove a source of randomness. Since plain El Gamal is not IND-CPA, this weaker variant is also not IND-CPA.

Formally, you could play the IND-CPA game as a challenger as follows: ask the oracle to encrypt m_0. Suppose the encryption is $(c_1, c_2) = (g^r, m_0 B^r)$. Now send two messages m_0 and m_1 and ask the oracle to encrypt one. If the oracle encrypted m_0, the ciphertext would be $(g^{r+1}, m_0 B^{r+1}) = (gc_1, Bc_2)$. You can calculate this value since you know g and B. If this matches what the oracle sent, you know that $M = m_0$. Otherwise, $M = m_1$. Therefore you can distinguish and you win the IND-CPA game.

Second part

The encryption of m_0 is $(g^r, m_0 B^r)$, and the encryption of m_1 is $(g^{r+1}, m_1 B^{r+1})$. Intuitively, the adversary wants to make the B^r and B^{r+1} terms cancel out to learn some relationship between m_0 and m_1. Division (formally, multiplying by the inverse $\bmod p$) causes those terms to cancel out:

$$(m_0 B^r)/(m_1 B^{r+1}) = m_0/m_1 B$$

Multiplying this result by B (which is publicly known) gives m_0/m_1.

✔ Correct

Save Answer Last saved on **Feb 25 at 3:18 AM**

Q4 Length Extension

4 Points

One subtle crypto fail you've heard about in lecture is SHA-2's susceptibility to *length extension attacks*. In this question, we'll walk you through a simplified version of the SHA-2 algorithm and show how that algorithm is fundamentally vulnerable.

Q4.1

1 Point

Hashes are functions that map a string of arbitrary length to a string of constant length. However, it turns out the SHA-2 algorithm actually only works if the input is a multiple of 64 bytes. So the first step of the SHA-2 algorithm is to pad the input by appending `0` repeatedly to the input until its length is a multiple of 64.

In problem 4, we reasoned that padding with zeros isn't a valid padding to use for encryption. Why is it acceptable for hashing?

- ○ We always know the input's length before running the hashing algorithm
- ◉ There is no need to recover a hash function's input
- ○ Cryptographic hash functions map zeros to random bits

EXPLANATION

The whole point of cryptographic hash functions is that we can apply them to arbitrary inputs and produce outputs which are impossible to reverse. Adding zeros does nothing to mask *or* reveal the actual input, so it's a perfectly acceptable choice for our padding here.

In practice, the padding is a little more complicated to avoid the problem where `message` and `message0` have the same hash (since they'd be the same after padding in our simplified scheme).

✔ **Correct**

Save Answer Last saved on **Feb 25 at 3:20 AM**

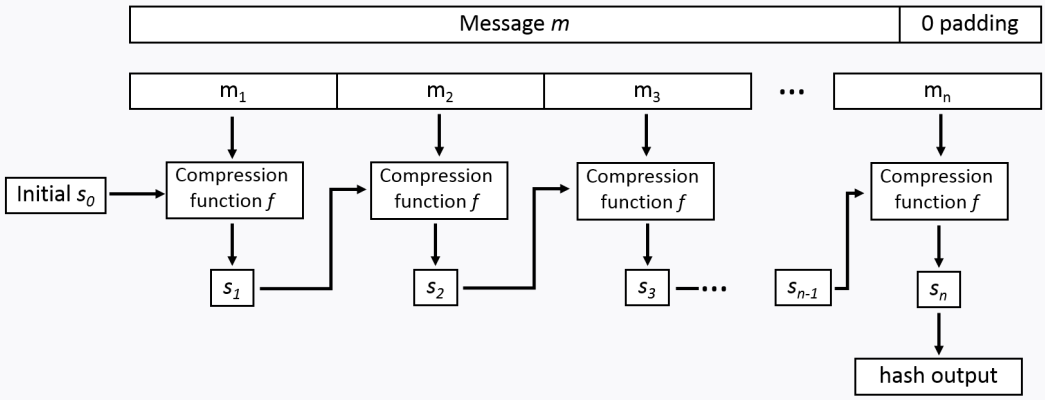
Q4.2

1 Point

Given a padded input (i.e. the input length is a multiple of 64), the SHA-2 algorithm first divides the message into 64-byte blocks. Then, it initializes its *internal state* to a constant, publicly known value s_0.

For each block, SHA-2 updates the internal state by calculating $s_i = f(s_{i-1}, m_i)$, where s_i is the new internal state after processing the current block, s_{i-1} is the previous internal state, m_i is the current block, and f is some complicated *one-way compression function*.

The final hash output is the internal state after all n blocks have been processed.



Suppose that an attacker observes an internal state s_i before the algorithm completes $(i < n)$. Can they compute the hash output $\text{SHA-2}(m)$ without knowing m?

- Yes
- No

EXPLANATION

Because the attacker doesn't know the final chunks of m, they cannot simulate the rest of the SHA-2 algorithm.

✔ **Correct**

Save Answer Last saved on **Feb 25 at 3:22 AM**

Q4.3

1 Point

Again, suppose the attacker observes an internal state s_i. Let m' be an arbitrary one-block-long message of the attacker's choosing.

Can the attacker compute $\text{SHA-2}(m_1 \,||\, m_2 \,||\, \ldots \,||\, m_i \,||\, m')$?

- Yes
- No

EXPLANATION

Since the attacker knows the internal state of the algorithm after processing block m_i, they can simply continue running the algorithm with their modified message m'. Formally, the attacker can calculate $f(s_i, m')$ to learn the internal state after processing m', and output this since m' is the last block of the message.

Notice that if $i = n$, the internal state s_n is the hash output! This means that if the attacker knows the hash output $\mathrm{SHA2}(m)$, they can also calculate $\mathrm{SHA2}(m||m')$, for any arbitrary m'.

✔ Correct

Save Answer Last saved on **Feb 25 at 3:22 AM**

Q4.4

1 Point

Suppose that SHA-2 used 8-byte blocks instead of 64-byte blocks. Given SHA-2(`EvanBot`), which of the following could an attacker calculate?

- [ ] SHA-2(EvanBot-is-real)
- [x] SHA-2(EvanBot001)
- [ ] SHA-2(Evan-is-a-Bot)
- [ ] None of the above

EXPLANATION

Given SHA-2(`EvanBot`), the attacker knows the internal state of the hash after processing the block `EvanBot0` (since the message has to be 0-padded until it's 8 bytes long). This means the attacker can add an arbitrary message after `EvanBot0` (option 2). However, the attacker cannot add an arbitrary message after just `EvanBot` (option 1) or add messages in between `Evan` and `Bot` (option 3).

Further reading: Stack Overflow, Wikipedia

✔ Correct

Save Answer Last saved on **Feb 25 at 3:23 AM**

Q5 Feedback

1 Point

What's something we could do to make the class better? Or, what did you find most difficult or confusing from lectures or the rest of class, and what would you like to see explained better?

If you have feedback, submit your comments on **this form**. Your name will not be connected to any feedback you provide, and anything you submit here will not affect your grade.

Leaving feedback is completely optional! To encourage submissions, please grab the magic word from the feedback form (available in the first page) and enter it here.

SHA-256

✔ **Correct**

Save Answer Last saved on **Feb 25 at 3:25 AM**

Save All Answers

Submit & View Submission ›