Week 4 - Problem Set

4/10 points earned (40%)

You haven't passed yet. You need at least 80% to pass. Review the material and try again! You have 3 attempts every 8 hours.

Review Related Lesson (/learn/crypto/home/week/4)

---

0 / 1 points

1.
An attacker intercepts the following ciphertext (hex encoded):

20814804c1767293b99f1d9cab3bc3e7
ac1e37bfb15599e5f40eef805488281d

He knows that the plaintext is the ASCII encoding of the message "Pay Bob 100$" (excluding the quotes). He also knows that the cipher used is CBC encryption with a random IV using AES as the underlying block cipher.

Page Saved

Add comment...

Show that the attacker can change the ciphertext so that it will decrypt to "Pay Bob 500$". What is the resulting ciphertext (hex encoded)?

This shows that CBC provides no integrity.

Enter answer here

Incorrect Response

---

0 / 1 points

2.

Let (E, D) be an encryption system with key space K, message space $\{0,1\}^n$ and ciphertext space $\{0,1\}^s$. Suppose (E, D) provides authenticated encryption. Which of the following systems provide authenticated encryption: (as usual, we use $\|$ to denote string concatenation)

☐ $E'(k,m) = \left[c \leftarrow E(k,m),\ \text{output}\ (c,c)\right]$ and

$$D'(k,\ (c_1,c_2)\ ) = \begin{cases} D(k,c_1) & \text{if } c_1 = c_2 \\ \perp & \text{otherwise} \end{cases}$$

Correct Response

(E', D') provides authenticated encryption because an attack on (E', D')

directly gives an attack on (E, D).

☐ $E'(k,m) = \big(E(k,m),\ E(k,m)\big)$ and

$$D'(k,\ (c_1,c_2)\ ) = D(k,c_1)$$

Correct Response

This system does not provide ciphertext integrity. The attacker

can query for $E'(k,0^n)$ to obtain (c_1, c_2). It then outputs

$(c_1, 0^s)$ and wins the ciphertext integrity game.

☐ $E'(k,m) = \big(E(k,m),\ H(m)\ \big)$ and

$$D'(k,\ (c,h)\ ) = \begin{cases} D(k,c) & \text{if } H(D(k,c)) = h \\ \perp & \text{otherwise} \end{cases}$$

(here H is some collision resistant hash function)

Incorrect Response

This system is not CPA secure because $H(m)$ leaks information about

the message in the ciphertext.

☐ $E'\left((k_1, k_2), m\right) = E(k_2,\ E(k_1, m))$ and

$$D'\left((k_1, k_2),\ c\right) = \begin{cases} D(k_1,\ D(k_2, c)) & \text{if } D(k_2, c) \neq \perp \\ \perp & \text{otherwise} \end{cases}$$

Correct Response

(E', D') provides authenticated encryption because an attack on (E', D')

gives an attack on (E, D). It's an interesting exercise to work out

the ciphertext integrity attack on (E, D) given a ciphertext integrity

attacker on (E', D').

✔ 1 / 1 points

3.
If you need to build an application that needs to encrypt multiple

messages using a single key, what encryption

method should you use? (for now, we ignore the question of key generation

and management)

○ invent your own mode of operation and implement it yourself.

○ use a standard implementation of one of the authenticated encryption modes GCM, CCM, EAX or OCB.

Correct Response

○ implement MAC-then-Encrypt yourself

○ implement Encrypt-and-MAC yourself

0 / 1 points

4.

Let (E, D) be a symmetric encryption system with message space M (think

of M as only consisting for short messages, say 32 bytes).

Define the following MAC (S, V) for messages in M:

$$S(k, m) := E(k, m) \quad ; \quad V(k, m, t) := \begin{cases} 1 & \text{if } D(k, t) = m \\ 0 & \text{otherwise} \end{cases}$$

What is the property that the encryption system (E, D) needs to satisfy

for this MAC system to be secure?

- ciphertext integrity
- perfect secrecy
- semantic security

Incorrect Response

the one time pad, for example, would not give a secure MAC.

- semantic security under a chosen plaintext attack

0 / 1 points

5.

In Key Derivation (https://www-origin.coursera.org/learn/crypto/lecture/A1ETP/key-derivation) we discussed how to derive session keys

from a shared secret. The problem is what to do when the shared

secret is non-uniform. In this question we show that using a PRF with

a *non-uniform* key may result in non-uniform values. This shows that

session keys cannot be derived by directly using a *non-uniform*

secret as a key in a PRF. Instead, one has to use a key derivation

function like HKDF.

Suppose k is a *non-uniform* secret key sampled from the key space $\{0,1\}^{256}$.

In particular, k is sampled uniformly from the set of all keys whose most significant

128 bits are all 0. In other words, k is chosen uniformly from a small subset of the key space. More precisely,

$$\text{for all } c \in \{0,1\}^{256}: \quad \Pr[k=c] = \begin{cases} 1/2^{128} & \text{if } \mathrm{MSB}_{128}(c) = 0^{128} \\ 0 & \text{otherwise} \end{cases}$$

Let $F(k,x)$ be a secure PRF with input space $\{0,1\}^{256}$. Which

of the following is a secure PRF when the key k is uniform in the

key space $\{0,1\}^{256}$, but is insecure when the key is sampled from the *non-uniform*

distribution described above?

- $F'(k,x) = \begin{cases} F(k,x) & \text{if } \mathrm{MSB}_{128}(k) \neq 0^{128} \\ 0^{256} & \text{otherwise} \end{cases}$
- $F'(k,x) = \begin{cases} F(k,x) & \text{if } \mathrm{MSB}_{128}(k) = 0^{128} \\ 0^{256} & \text{otherwise} \end{cases}$
- $F'(k,x) = \begin{cases} F(k,x) & \text{if } \mathrm{MSB}_{128}(k) \neq 1^{128} \\ 0^{256} & \text{otherwise} \end{cases}$
- $F'(k,x) = \begin{cases} F(k,x) & \text{if } \mathrm{MSB}_{128}(k) = 0^{128} \\ 1^{256} & \text{otherwise} \end{cases}$

Incorrect Response

This F' is trivially insecure as a PRF.

---

0 / 1 points

6.

In what settings is it acceptable to use *deterministic* authenticated

encryption (DAE) like SIV?

- when the encryption key is used to encrypt only one message.
- when a fixed message is repeatedly encrypted using a single key.
- to individually encrypt many packets in a voice conversation with a single key.

Incorrect Response

This would be insecure because an attacker would be able to tell

when two transmitted packets are equal.

- to encrypt many records in a database with a single key when the same record may repeat multiple times.

---

1 / 1 points

7.

Let $E(k, x)$ be a secure block cipher. Consider the following

tweakable block cipher:

$$E'\left((k_1, k_2), t, x\right) = E(k_1, x) \bigoplus E(k_2, t).$$

Is this tweakable block cipher secure?

- no because for $x \neq x'$ we have

$$E'((k_1, k_2), 0, x) \bigoplus E'((k_1, k_2), 1, x) = E'((k_1, k_2), 0, x') \bigoplus E'((k_1, k_2), 1, x')$$

Correct Response

since this relation holds, an attacker can make 4 queries to E'

and distinguish E' from a random collection of one-to-one functions.

- no because for $x \neq x'$ and $t \neq t'$ we have

  $$E'((k_1, k_2), t, x) \bigoplus E'((k_1, k_2), t', x) = E'((k_1, k_2), t, x') \bigoplus E'((k_1, k_2), t', x)$$

- no because for $t \neq t'$ we have

  $$E'((k_1, k_2), t, 0) \bigoplus E'((k_1, k_2), t', 1) = E'((k_1, k_2), t', 1) \bigoplus E'((k_1, k_2), t', 0)$$

- yes, it is secure assuming E is a secure block cipher.

- no because for $x \neq x'$ we have

  $$E'((k_1, k_2), 0, x) \bigoplus E'((k_1, k_2), 0, x) = E'((k_1, k_2), 0, x') \bigoplus E'((k_1, k_2), 0, x')$$

✔ 1 / 1 points

8.

In Format Preserving Encryption (https://www-origin.coursera.org/learn/crypto/lecture/aFRSZ/format-preserving-encryption) we discussed format preserving encryption

which is a PRP on a domain $\{0, \ldots, s-1\}$ for some pre-specified

value of s.

Recall that the construction we presented worked in two steps,

where the second step worked by iterating the PRP until the output

fell into the set $\{0, \ldots, s-1\}$.

Suppose we try to build a format preserving credit card encryption

system from AES using *only* the second step. That is, we start with

a PRP with domain $\{0,1\}^{128}$ from which we want to build a PRP

with domain 10^{16}. If we only used step (2), how many iterations of

AES would be needed in expectation for each evaluation of the PRP

with domain 10^{16}?

- 2^{128}
- $2^{128}/10^{16} \approx 3.4 \times 10^{22}$

Correct Response

On every iteration we have a probability of $10^{16}/2^{128}$ of falling

into the set $\{0, \ldots, 10^{16}\}$ and therefore in expectation we

will need $2^{128}/10^{16}$ iterations. This should explain why

step (1) is needed.

- 4
- $10^{16}/2^{128}$

---

1 / 1 points

9.

Let (E, D) be a secure tweakable block cipher.

Define the following MAC (S, V):

$$S(k, m) := E(k, m, 0) \quad ; \quad V(k, m, \text{tag}) := \begin{cases} 1 & \text{if } E(k, m, 0) = \text{tag} \\ 0 & \text{otherwise} \end{cases}$$

In other words, the message m is used as the tweak and the plaintext given to E is always set to 0.

Is this MAC secure?

- it depends on the tweakable block cipher.
- yes

Correct Response

A tweakable block cipher is indistinguishable from a

collection of random permutations. The chosen message attack on the

MAC gives the attacker the image of 0 under a number of the

permutations in the family. But that tells the attacker nothing about

the image of 0 under some other member of the family.

- no

---

0 / 1 points

10.

In CBC Padding Attacks (https://www-origin.coursera.org/learn/crypto/lecture/8s23o/cbc-padding-attacks) we discussed padding oracle attacks. These chosen-ciphertext attacks can break poor implementations of MAC-then-encrypt.

Consider a system that implements MAC-then-encrypt where encryption is done using CBC with a random IV using AES as the block cipher. Suppose the system is vulnerable to a padding oracle attack. An attacker intercepts a 64-byte ciphertext c (the first 16 bytes of c are the IV and the remaining 48 bytes are the encrypted payload). How many chosen ciphertext queries would the attacker need *in the worst case* in order to decrypt the entire 48 byte payload? Recall that padding oracle attacks decrypt the payload one byte at a time.

- [x] 16384

Incorrect Response
Padding oracle attacks decrypt the payload one byte at a time. For each byte the attacker needs no more than 256 guesses in the worst case. Since there are only 48 bytes of encrypted payload, the attacker need fewer than 16386 queries.

- [ ] 256
- [ ] 1024
- [ ] 48
- [ ] 12288