

✔ **Congratulations! You passed!**
TO PASS 80% or higher

Keep Learning

GRADE
88.88%

Week 2 - Problem Set

LATEST SUBMISSION GRADE
88.88%

1. Consider the following five events:

1 / 1 point

- Correctly guessing a random 128-bit AES key on the first try.
- Winning a lottery with 1 million contestants (the probability is $1/10^6$).
- Winning a lottery with 1 million contestants 5 times in a row (the probability is $(1/10^6)^5$).
- Winning a lottery with 1 million contestants 6 times in a row.
- Winning a lottery with 1 million contestants 7 times in a row.

What is the order of these events from most likely to least likely?

- ☒ 2, 3, 4, 1, 5
- ☐ 2, 3, 5, 4, 1
- ☐ 2, 3, 4, 5, 1
- ☐ 3, 2, 5, 4, 1

✔ **Correct**

- The probability of event (1) is $1/2^{128}$.
- The probability of event (5) is $1/(10^6)^7$ which is about $1/2^{(1139)}$. Therefore, event (5) is the least likely.
- The probability of event (4) is $1/(10^6)^6$ which is about $1/2^{(119.5)}$ which is more likely than event (1).
- The remaining events are all more likely than event (4).

2. Suppose that using commodity hardware it is possible to build a computer

1 / 1 point

for about \$200 that can brute force about 1 billion AES keys per second.

Suppose an organization wants to run an exhaustive search for a single

128-bit AES key and was willing to spend 4 trillion dollars to buy these

machines (this is more than the annual US federal budget). How long would

it take the organization to brute force this single 128-bit AES key with

these machines? Ignore additional costs such as power and maintenance.

- ☐ More than a week but less than a month
- ☐ More than a month but less than a year
- ☒ More than a billion (10^9) years
- ☐ More than a 100 years but less than a million years
- ☐ More than a year but less than 100 years

✔ **Correct**

The answer is about 540 billion years.

- # machines = $4 \times 10^{12} / 200 = 2 \times 10^{10}$
- # keys processed per sec = $10^9 \times (2 \times 10^{10}) = 2 \times 10^{19}$
- # seconds = $2^{128} / (2 \times 10^{19}) = 1.7 \times 10^{19}$

This many seconds is about 540 billion years.

3. Let $F': \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a secure PRF (i.e. a PRF where the key space, input space, and output space are all $\{0,1\}^n$) and say $n = 128$.

Which of the following is a secure PRF (there is more than one correct answer):

☒ $F'(k, x) = F(k, x)[0, \dots, n-2]$

(i.e., $F'(k, x)$ drops the last bit of $F(k, x)$)

✔ **Correct**

Correct. A distinguisher for F' gives a distinguisher for F .

☒ $F'(k, x) = k \oplus x$

! **This should not be selected**

Not a PRF. A distinguisher will query at $x = 0^n$ and $x = 1^n$ and

output not random if the xor of the response is 1^n .

This is unlikely to hold for a truly random function.

☒ $F'(k, x) = F(k, x) \oplus F(k, x \oplus 1^n)$

! **This should not be selected**

Not a PRF. A distinguisher will query at $x = 0^n$ and $x = 1^n$ and

output *not random* whenever the two responses are

equal. This is unlikely to happen for a truly random

function.

☒ $F'((k_1, k_2), x) = F(k_1, x) \parallel F(k_2, x)$ (here \parallel denotes concatenation)

✔ **Correct**

Correct. A distinguisher for F' gives a distinguisher for F .

☒ $F'(k, x) = F(k, x) \parallel 0$

(here \parallel denotes concatenation)

! **This should not be selected**

Not a PRF. A distinguisher will output *not random* whenever the last bit

of $F(k, 0^n)$ is 0.

☒ $F'((k_1, k_2), x) = \begin{cases} F(k_1, x) & \text{when } x \neq 0^n \\ k_2 & \text{otherwise} \end{cases}$

✔ **Correct**

Correct. A distinguisher for F' gives a distinguisher for F .

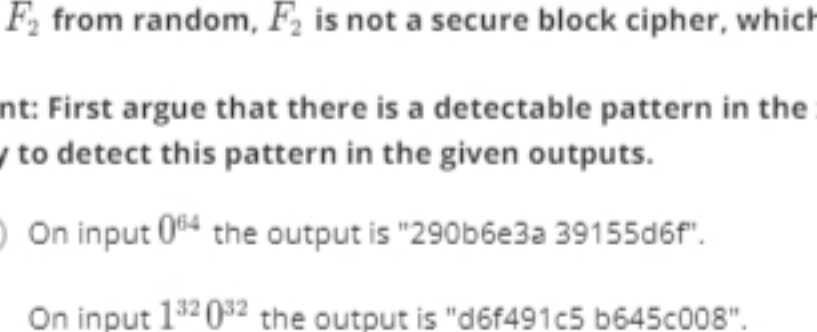
4. Recall that the Luby-Rackoff theorem discussed in [The Data Encryption Standard lecture](#) states that applying a three round Feistel network to a secure PRF gives a secure block cipher. Let's see what goes wrong if we only use a two round Feistel.

1 / 1 point

Let $F': K \times \{0,1\}^{32} \rightarrow \{0,1\}^{32}$ be a secure PRF.

Recall that a 2-round Feistel defines the following PRP

$F_2: K^2 \times \{0,1\}^{64} \rightarrow \{0,1\}^{64}$:



Here R_i is the right 32 bits of the 64-bit input and L_i is the left 32 bits.

One of the following lines is the output of this PRP F_2 using a random key, while the other three are the output of a truly random permutation $F': \{0,1\}^{64} \rightarrow \{0,1\}^{64}$. All 64-bit outputs are encoded as 16 hex characters.

Can you say which is the output of the PRP? Note that since you are able to distinguish the output of F_2 from random, F_2 is not a secure block cipher, which is what we wanted to show.

Hint: First argue that there is a detectable pattern in the xor of $F_2(\cdot, 0^{64})$ and $F_2(\cdot, 1^{62}0^{02})$. Then try to detect this pattern in the given outputs.

- ☒ On input 0^{64} the output is "290b6e3a 39155d6f".
- On input $1^{62}0^{02}$ the output is "d6f491c5 b645c008".
- ☐ On input 0^{64} the output is "5967abaf 5210722b".
- On input $1^{62}0^{02}$ the output is "b0e033c0 0bc9330e".
- ☐ On input 0^{64} the output is "9d1a4f78 cb28d863".
- On input $1^{62}0^{02}$ the output is "75a5e3ea 773ac3e0".
- ☐ On input 0^{64} the output is "7b50baad 07640c3d".
- On input $1^{62}0^{02}$ the output is "ac343a22 cea46d60".

✔ **Correct**

Observe that the two round Feistel has the property that

the left of $F(\cdot, 0^{64}) \oplus F(\cdot, 1^{62}0^{02})$ is 1^{32} .

The two outputs in this answer are the only ones with this property.

5. Nonce-based CBC. Recall that in [Lecture 4.4](#) we said that if one wants to use CBC encryption with a non-random unique nonce then the nonce must first be encrypted with an independent PRP key and the result then used as the CBC IV.

1 / 1 point

Let's see what goes wrong if one encrypts the nonce with the same PRP key as the key used for CBC encryption.

Let $F': K \times \{0,1\}^{\ell} \rightarrow \{0,1\}^{\ell}$ be a secure PRP with, say, $\ell = 128$. Let n be a nonce and suppose one encrypts a message m by first computing $IV = F'(k, n)$ and then using this IV in CBC encryption using $F'(k, \cdot)$. Note that the same key k is used for computing the IV and for CBC encryption. We show that the resulting system is not nonce-based CPA secure.

The attacker begins by asking for the encryption of the two block message $m = (0^{\ell}, 0^{\ell})$ with nonce $n = 0^{\ell}$. It receives back a two block ciphertext (c_0, c_1) . Observe that by definition of CBC we know that $c_0 = F'(k, c_0)$.

Next, the attacker asks for the encryption of the one block message $m_1 = 0^{\ell} \oplus c_1$ with nonce $n = c_0$. It receives back a one block ciphertext c'_0 .

What relation holds between c_0, c_1, c'_0 ? Note that this relation lets the adversary win the nonce-based CPA game with advantage 1.

- ☐ $c_1 = 0^{\ell}$
- ☐ $c_1 = c_0 \oplus c'_0$
- ☒ $c_1 = c'_0$
- ☐ $c_0 = c'_0$

✔ **Correct**

This follows from the definition of CBC with an encrypted nonce

as defined in the question.

6. Let m be a message consisting of ℓ AES blocks

1 / 1 point

(say $\ell = 100$). Alice encrypts m using CBC mode and transmits

the resulting ciphertext to Bob. Due to a network error,

ciphertext block number $\ell/2$ is corrupted during transmission.

All other ciphertext blocks are transmitted and received correctly.

Once Bob decrypts the received ciphertext, how many plaintext blocks

will be corrupted?

- ☒ 2
- ☐ 1
- ☐ $1 + \ell/2$
- ☐ 0
- ☐ $\ell/2$

✔ **Correct**

Take a look at the CBC decryption circuit. Each ciphertext

blocks affects only the current plaintext block and the next.

7. Let m be a message consisting of ℓ AES blocks (say $\ell = 100$). Alice encrypts m using randomized counter mode and

1 / 1 point

transmits the resulting ciphertext to Bob. Due to a network error,

ciphertext block number $\ell/2$ is corrupted during transmission.

All other ciphertext blocks are transmitted and received correctly.

Once Bob decrypts the received ciphertext, how many plaintext blocks

will be corrupted?

- ☒ 1
- ☐ 0
- ☐ $1 + \ell/2$
- ☐ 2
- ☐ $\ell/2$

✔ **Correct**

Take a look at the counter mode decryption circuit. Each

ciphertext block affects only the current plaintext block.

8. Recall that encryption systems do not fully hide the length of

1 / 1 point

transmitted messages. Leaking the length of web requests [has been used](#) to eavesdrop on encrypted HTTPS traffic to a number of

web sites, such as tax preparation sites, Google searches, and

healthcare sites.

Suppose an attacker intercepts a packet where he knows that the

packet payload is encrypted using AES in CBC mode with a random IV. The

encrypted packet payload is 128 bytes. Which of the following

messages is plausibly the decryption of the payload:

- ☒ "In this letter I make some remarks on a general principle relevant to enciphering in general and my machine."
- ☐ "If qualified opinions incline to believe in the exponential conjecture, then I think we cannot afford not to make use of it."
- ☐ "The most direct computation would be for the enemy to try all 2^{2^r} possible keys, one by one."
- ☐ "The significance of this general conjecture, assuming its truth, is easy to see. It means that it may be feasible to design ciphers that are effectively unbreakable."

✔ **Correct**

The length of the string is 107 bytes, which after padding becomes 112 bytes,

and after prepending the IV becomes 128 bytes.

9. Let $R := \{0,1\}^4$ and consider the following PRF $F: R^2 \times R \rightarrow R$ defined as follows:

1 / 1 point

$$F(k, x) := \begin{cases} t = k[0] & \text{for } i=1 \text{ to } 4 \text{ do} \\ & \text{if } x[i] = 1 \text{ then } t = t \oplus k[i] \\ & \text{output } t \end{cases}$$

That is, the key is $k = (k[0], k[1], k[2], k[3], k[4])$ in R^5 and the function at, for example, 0101 is defined as $F(k, 0101) = k[0] \oplus k[2] \oplus k[4]$.

For a random key k unknown to you, you learn that

$F(k, 0110) = 0011$ and $F(k, 0101) = 1010$ and $F(k, 1110) = 0110$.

What is the value of $F(k, 1101)$? Note that since you are able to predict the function at a new point, this PRF is insecure.

1111

✔ **Correct**