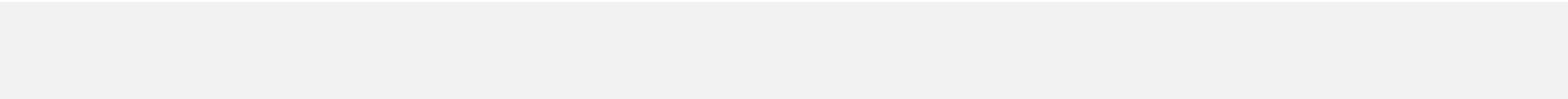


You submitted this homework on **Sat 12 Sep 2015 3:21 PM EEST**. You got a score of **10.00** out of **10.00**.



Question 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. 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.

You entered:

20814804c1767293bd9f1d9cab3bc3e7
ac1e37bfb15599e5f40eef805488281d

Your Answer	Score	Explanation
20814804c1767293bd9f1d9cab3bc3e7 ac1e37bfb15599e5f40eef805488281d	✓ 1.00	You got it!
Total	1.00 / 1.00	

Question 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 \parallel to denote string concatenation)

Your Answer	Score	Explanation
<input type="checkbox"/> $E'(k, m) = E(k, m)$ and $D'(k, c) = \begin{cases} D(k, c) & \text{if } D(k, c) \neq \perp \\ 0^n & \text{otherwise} \end{cases}$	✓ 0.25	This system does not provide ciphertext integrity since an attacker can simply output the ciphertext 0^s and win the ciphertext integrity game.
<input type="checkbox"/> $E'(k, m) = (E(k, m), 0)$ and $D'(k, (c, b)) = D(k, c)$	✓ 0.25	This system does not provide ciphertext integrity. The attacker queries for $E'(k, 0^n)$ to obtain $(c, 0)$. It then outputs $(c, 1)$ and wins the ciphertext integrity game.
<input checked="" type="checkbox"/> $E'(k, m) = E(k, m \oplus 1^n)$ and $D'(k, c) = \begin{cases} D(k, c) \oplus 1^n & \text{if } D(k, c) \neq \perp \\ \perp & \text{otherwise} \end{cases}$	✓ 0.25	(E', D') provides authenticated encryption because an attack on (E', D') directly gives an attack on (E, D) .
<input checked="" type="checkbox"/> $E'((k_1, k_2), m) = E(k_2, E(k_1, m))$ and $D'((k_1, k_2), c) = \begin{cases} D(k_1, D(k_2, c)) & \text{if } D(k_2, c) \neq \perp \\ \perp & \text{otherwise} \end{cases}$	✓ 0.25	(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') .

Total	1.00 / 1.00
-------	-------------

Question 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)

Your Answer	Score	Explanation
<input checked="" type="radio"/> use a standard implementation of one of the authenticated encryption modes GCM, CCM, EAX or OCB.	✓ 1.00	
<input type="radio"/> use a standard implementation of CBC encryption with a random IV.		
<input type="radio"/> invent your own mode of operation and implement it yourself.		
<input type="radio"/> use a standard implementation of randomized counter mode.		
Total	1.00 / 1.00	

Question 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?

Your Answer	Score	Explanation
<input type="radio"/> perfect secrecy		
<input type="radio"/> chosen ciphertext security		
<input type="radio"/> semantic security		
<input checked="" type="radio"/> authenticated encryption	✓ 1.00	Indeed, authenticated encryption implies ciphertext integrity which prevents existential forgery under a chosen message attack.
Total	1.00 / 1.00	

Question 5

In [lecture 8.1](#) 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,

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

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?

Your Answer	Score	Explanation
<input type="radio"/> $F'(k, x) = \begin{cases} F(k, x) & \text{if } \text{MSB}_{128}(k) \neq 1^{128} \\ 0^{256} & \text{otherwise} \end{cases}$		
<input type="radio"/> $F'(k, x) = F(k, x)$		
<input type="radio"/> $F'(k, x) = \begin{cases} F(k, x) & \text{if } \text{MSB}_{128}(k) \neq 1^{128} \\ 1^{256} & \text{otherwise} \end{cases}$		
<input checked="" type="radio"/> $F'(k, x) = \begin{cases} F(k, x) & \text{if } \text{MSB}_{128}(k) \neq 0^{128} \\ 1^{256} & \text{otherwise} \end{cases}$	<div>✓</div> <div>1.00</div>	<p>$F'(k, x)$ is a secure PRF because for a uniform key k the probability that $\text{MSB}_{128}(k) = 0^{128}$ is negligible. However, for the *non-uniform* key k this PRF always outputs 1 and is therefore completely insecure. This PRF cannot be used as a key derivation function for the distribution of keys described in the problem.</p>
Total	1.00 / 1.00	

Question 6

In what settings is it acceptable to use *deterministic* authenticated encryption (DAE) like SIV?

Your Answer	Score	Explanation
<input type="radio"/> to individually encrypt many packets in a voice conversation with a single key.		
<input type="radio"/> when a fixed message is repeatedly encrypted using a single key.		
<input type="radio"/> to encrypt many records in a database with a single key when the same record may repeat multiple times.		

<input checked="" type="radio"/> when messages are chosen at random from a large enough space so that messages are unlikely to repeat.	✓ 1.00	Deterministic encryption is safe to use when the message/key pair is never used more than once.
Total	1.00 / 1.00	

Question 7

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

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

Is this tweakable block cipher secure?

Your Answer	Score	Explanation
<input checked="" type="radio"/> no because for $x \neq x'$ we have $E'((k_1, k_2), 0, x) \oplus E'((k_1, k_2), 1, x) = E'((k_1, k_2), 0, x') \oplus E'((k_1, k_2), 1, x')$	✓ 1.00	since this relation holds, an attacker can make 4 queries to E' and distinguish E' from a random collection of one-to-one functions.
<input type="radio"/> no because for $t \neq t'$ we have $E'((k_1, k_2), t, 0) \oplus E'((k_1, k_2), t', 1) = E'((k_1, k_2), t', 1) \oplus E'((k_1, k_2), t, 1)$		
<input type="radio"/> yes, it is secure assuming E is a secure block cipher.		
<input type="radio"/> no because for $x \neq x'$ we have $E'((k_1, k_2), 0, x) \oplus E'((k_1, k_2), 0, x) = E'((k_1, k_2), 0, x') \oplus E'((k_1, k_2), 0, x')$		
<input type="radio"/> no because for $x \neq x'$ and $t \neq t'$ we have $E'((k_1, k_2), t, x) \oplus E'((k_1, k_2), t', x) = E'((k_1, k_2), t, x') \oplus E'((k_1, k_2), t', x')$		
Total	1.00 / 1.00	

Question 8

In [lecture 8.5](#) we discussed format preserving encryption which is a PRP on a domain $\{0, \dots, s - 1\}$ 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 $\{0, \dots, s - 1\}$.

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} ?

Your Answer	Score	Explanation
<input type="radio"/> 4		
<input checked="" type="radio"/> $2^{128}/10^{16} \approx 3.4 \times 10^{22}$	✓ 1.00	On every iteration we have a probability of $10^{16}/2^{128}$ of falling into the set $\{0, \dots, 10^{16}\}$ and therefore in expectation we will need $2^{128}/10^{16}$ iterations. This should explain why step (1) is needed.
<input type="radio"/> 10^{16}		
<input type="radio"/> 2^{128}		
Total	1.00 / 1.00	

Question 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?

Your Answer	Score	Explanation
<input checked="" type="radio"/> yes	1.00	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.
<input type="radio"/> no		
<input type="radio"/> it depends on the tweakable block cipher.		
Total	1.00 / 1.00	

Question 10

In [Lecture 7.6](#) 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.

Your Answer	Score	Explanation
<input type="radio"/> 12240		
<input checked="" type="radio"/> 12288	1.00	Correct. 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 48 bytes total, the number queries needed is $256 \times 48 = 12288$.
<input type="radio"/> 48		



65536

Total

1.00 /
1.00