Question 1

Data compression is often used in data storage and transmission. Suppose you want to use data compression in conjunction with encryption. Does it make more sense to:

Your Answer		Score	Explanation
● Compress then encrypt.	✓	1.00	Ciphertexts tend to look like random strings and therefore the only opportunity for compression is prior to encryption.
Total		1.00 / 1.00	

Question 2

Let $G:\{0,1\}^s\rightarrow\{0,1\}^n$ be a secure PRG. Which of the following is a secure PRG (there is more than one correct answer):

Your Answer		Score	Explanation
☑ $G'(k)=G(k)\oplus 1^n$	✓	0.17	a distinguisher for G' gives a distinguisher for G.
□ $G'(k)=\text{reverse}(G(k))$ where reverse(x) reverses the string x so that the first bit of x is the last bit of reverse(x), the second bit of x is the second to last bit of reverse(x), and so on.	✓	0.17	a distinguisher for G' gives a distinguisher for G.
□ $G'(k)=G(k)\|\|0$ (here $\|\|$ denotes concatenation)	✓	0.17	A distinguisher will output *not random*whenever the last bit of its input is 0.
☑ $G'(k)=G(k\oplus 1^s)$	✓	0.17	a distinguisher for G' gives a distinguisher for G.
□ $G'(k)=G(k)\|\|G(k)$ (here $\|\|$ denotes concatenation)	✓	0.17	A distinguisher will output *not random*whenever the first n bits are equal to

			the last n bits.
□ $G'(k)=G(0)$	✓	0.17	A distinguisher will output *not random*whenever its input is equal to $G(0)$.
Total		1.00 / 1.00	

Your Answer		Score	Explanation
☑ G′(k)=G(k)[0,…,n−2] (i.e., G′(k) drops the last bit of G(k))	✓	0.17	a distinguisher for G′ gives a distinguisher for G.
☑ G′(k)=G(k)⊕1n	✓	0.17	a distinguisher for G′ gives a distinguisher for G.
□ G′(k)=G(k)‖‖0 (here ‖‖denotes concatenation)	✓	0.17	A distinguisher will output not randomwhenever the last bit of its input is 0.
□ G′(k)=G(0)	✓	0.17	A distinguisher will output not randomwhenever its input is equal to G(0).
☑ G′(k1,k2)=G(k1)‖‖G(k2) (here ‖‖ denotes concatenation)	✓	0.17	a distinguisher for G′ gives a distinguisher for G.
□ G′(k)=G(k)‖‖G(k) (here ‖‖denotes concatenation)	✓	0.17	A distinguisher will output not randomwhenever the first n bits are equal to the last n bits.
Total		1.00 / 1.00	

Question 3

Let $G:K\to\{0,1\}^n$ be a secure PRG. Define $G'(k_1,k_2)=G(k_1)\land G(k_2)$ where $\land$ is the bit-wise AND function. Consider the following statistical test A on $\{0,1\}^n$:
$A(x)$ outputs $\mathrm{LSB}(x)$, the least significant bit of x.
What is $Adv_{\mathrm{PRG}}[A,G']$?   You may assume that $\mathrm{LSB}(G(k))$ is 0 for exactly half the seeds k in K.

Note: Please enter the advantage as a decimal between 0 and 1 with a leading 0. If the advantage is 3/4, you should enter it as 0.75
Answer for Question 3

Your Answer		Score	Explanation
0.25	✓	1.00	for a random string x we have $Pr[A(x)=1]=1/2$ but for a pseudorandom string $G'(k_1,k_2)$ we have $Pr_{k_1,k_2}[A(G'(k_1,k_2))=1]=1/4$.
Total		1.00 / 1.00	

Question 4

Let (E,D) be a (one-time) semantically secure cipher with key space $K=\{0,1\}^\ell$. A bank wishes to split a decryption key $k\in\{0,1\}^\ell$ into two pieces p_1 and p_2 so that both are needed for decryption. The piece p_1 can be given to one executive and p_2 to another so that both must contribute their pieces for decryption to proceed.
The bank generates random k_1 in $\{0,1\}^\ell$ and sets $k_1'\leftarrow k\oplus k_1$. Note that $k_1\oplus k_1'=k$. The bank can give k_1 to one executive and k_1' to another. Both must be present for decryption to proceed since, by itself, each piece contains no information about the secret key k (note that each piece is a one-time pad encryption of k).
Now, suppose the bank wants to split k into three pieces p_1,p_2,p_3 so that any two of the pieces enable decryption using k. This ensures that even if one executive is out sick, decryption can still succeed. To do so the bank generates two random pairs (k_1,k_1') and (k_2,k_2') as in the previous paragraph so that $k_1\oplus k_1'=k_2\oplus k_2'=k$. How should the bank assign pieces so that any two pieces enable decryption using k, but no single piece can decrypt?

Your Answer		Score	Explanation
◉ $p_1=(k_1,k_2), p_2=(k_1',k_2), p_3=(k_2')$	✓	1.00	executives 1 and 2 can decrypt using k_1,k_1', executives 1 and 3 can decrypt using k_2,k_2', and executives 2 and 3 can decrypt using k_2,k_2'. Moreover, a single executive has no information about k.
Total		1.00 / 1.00	

Question 5

Let $M=C=K=\{0,1,2,\ldots,255\}$ and consider the following cipher defined over (K,M,C):

$E(k,m)=m+k \pmod{256}; D(k,c)=c-k \pmod{256}$.

Does this cipher have perfect secrecy?

Your Answer		Score	Explanation
◉ Yes.	✓	1.00	as with the one-time pad, there is exactly one key mapping a given message m to a given ciphertext c.
Total		1.00 / 1.00	

Question 6

Let (E,D) be a (one-time) semantically secure cipher where the message and ciphertext space is $\{0,1\}^n$. Which of the following encryption schemes are (one-time) semantically secure?

Your Answer		Score	Explanation
☑ $E'(k,m)=\text{reverse}(E(k,m))$	✓	0.17	an attack on E' gives an attack on E.
□ $E'(k,m)=E(k,m) \| \text{LSB}(m)$	✓	0.17	To break semantic security, an attacker would ask for the encryption of 0^n and $0^{n-1}1$ and can distinguish EXP(0) from EXP(1).
☑	✓	0.17	an attack on E' gives an attack on E.

$E'(\ (k,k'),\ m)=E(k,m)\|\|E(k',m)$				
□	$E'(k,m)=E(k,m)\|\|k$	✓	0.17	To break semantic security, an attacker would read the secret key from the challenge ciphertext and use it to decrypt the challenge ciphertext. Basically, any ciphertext reveals the secret key.
☑	$E'(k,m)=E(k,m)\|\|E(k,m)$	✓	0.17	an attack on E' gives an attack on E.
□	$E'(k,m)=E(0^n,m)$	✓	0.17	To break semantic security, an attacker would ask for the encryption of $0^n$ and $1^n$ and can easily distinguish EXP(0) from EXP(1) because it knows the secret key, namely $0^n$.
Total			1.00 / 1.00	

Question 7

Suppose you are told that the one time pad encryption of the message "attack at dawn" is*6c73d5240a948c86981bc294814d* (the plaintext letters are encoded as 8-bit ASCII and the given ciphertext is written in hex). What would be the one time pad encryption of the message "attack at dusk" under the same OTP key?

Answer for Question 7

Your Answer		Score	Explanation
6c73d5240a948c86981bc2808548	✓	1.00	
Total		1.00 / 1.00	

Question 8

The movie industry wants to protect digital content distributed on DVD’s. We develop a variant of a method used to protect Blu-ray disks called AACS.

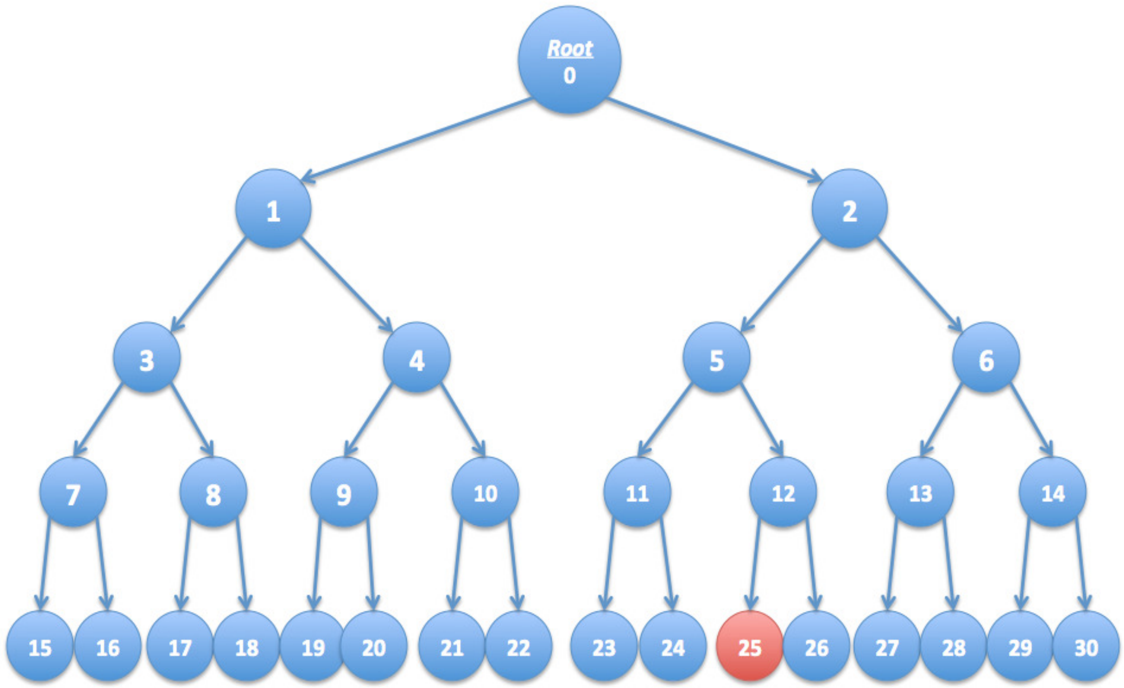
Suppose there are at most a total of n DVD players in the world (e.g. $n=2^{32}$). We view these $n$players as the leaves of a binary tree of height $\log_2 n$. Each node in this binary tree contains an AES key k_i. These keys are kept secret from consumers and are fixed for all time. At manufacturing time each DVD player is assigned a serial number $i \in [0, n-1]$. Consider the set of nodes S_i along the path from the root to leaf number i in the binary tree. The manufacturer of the DVD player embeds in player number i the keys associated with the nodes in the set S_i. A DVD movie m is encrypted as

$E(k_{\text{root}}, k) \| \| E(k, m)$

where k is a random AES key called a content-key and k_{root} is the key associated with the root of the tree. Since all DVD players have the key k_{root} all players can decrypt the movie m. We refer to $E(k_{\text{root}}, k)$ as the header and $E(k, m)$ as the body. In what follows the DVD header may contain multiple ciphertexts where each ciphertext is the encryption of the content-key k under some key k_i in the binary tree.

Suppose the keys embedded in DVD player number r are exposed by hackers and published on the Internet. In this problem we show that when the movie industry distributes a new DVD movie, they can encrypt the contents of the DVD using a slightly larger header (containing about $\log_2 n$ keys) so that all DVD players, except for player number r, can decrypt the movie. In effect, the movie industry disables player number r without affecting other players.

As shown below, consider a tree with $n=16$ leaves. Suppose the leaf node labeled 25 corresponds to an exposed DVD player key. Check the set of keys below under which to encrypt the key k so that*every player* other than player 25 can decrypt the DVD. Only four keys are needed.



Your Answer	Score	Explanation

☑ 11	✓	0.03	You cannot encrypt k under key 5, but 11's children must be able to decrypt k.
□ 10	✓	0.03	There is a better solution that does not require encrypting on the key of this node.
□ 18	✓	0.03	There is a better solution that does not require encrypting on the key of this node.
☑ 1	✓	0.03	You cannot encrypt k under the root, but 1's children must be able to decrypt k.
☑ 26	✓	0.03	You cannot encrypt k under any key on the path from the root to node 25. Therefore 26 can only decrypt if you encrypt k under key k_{26}.
□ 23	✓	0.03	There is a better solution that does not require encrypting on the key of this node.
□ 17	✓	0.03	There is a better solution that does not require encrypting on the key of this node.
☑ 6	✓	0.03	You cannot encrypt k under 2, but 6's children must be able to decrypt k.
Total		0.25 / 0.25	

Question explanation

Question 9

Continuing with the previous question, if there are n DVD players, what is the number of keys under which the content key k must be encrypted if exactly one DVD player's key needs to be revoked?

Your Answer		Score	Explanation
◉ $\log_2 n$	✓	1.00	That's right. The key will need to be encrypted under one key for each node on the path from the root to the revoked leaf. There are $\log_2 n$nodes on the path.
Total		1.00 / 1.00	

Question 10

Continuing with question 8, suppose the leaf nodes labeled 16, 18, and 25 correspond to exposed DVD player keys. Check the smallest set of keys under which to encrypt the key k so that every player other than players 16,18,25 can decrypt the DVD. Only six keys are needed.

Your Answer		Score	Explanation
☐ 30	✓	0.02	
☑ 15	✓	0.02	Yes, this will let player 15 decrypt.
☑ 26	✓	0.02	Yes, this will let player 26 decrypt.
☑ 4	✓	0.02	Yes, this will let players 19-22 decrypt.
☐ 29	✓	0.02	
☑ 17	✓	0.02	Yes, this will let player 17 decrypt.
☑ 6	✓	0.02	Yes, this will let players 27-30 decrypt.
☐ 22	✓	0.02	
☐ 7	✓	0.02	
☑ 11	✓	0.02	Yes, this will let players 23,24 decrypt.
Total		0.20 / 0.20	

Question 1

Consider the following five events:

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

Your Answer		Score	Explanation
◉ 2, 3, 4, 1, 5	✓	1.00	• The probability of event (1) is 1/2^128. • The probability of event (5) is 1/(10^6)^7 which is about 1/2^{139}. 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).
Total		1.00 / 1.00	

Question 2

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

Your Answer		Score	Explanation
◉ More than a billion (10^9) years	✓	1.00	The answer is about 540 billion years. • # machines = 4*10^12/200 = 2*10^10 • # keys processed per sec = 10^9 * (2*10^10) = 2*10^19 • # seconds = 2^128 / (2*10^19) = 1.7*10^19 This many seconds is about 540 billion years.
Total		1.00 / 1.00	

Question 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):

Your Answer		Score	Explanation
□ $F'(k, x)=k\oplus x$	✓	0.17	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.
☑ $F'((k_1,k_2), x)=F(k_1,x)$ ‖‖ $F(k_2,x)$ (here ‖‖ denotes concatenation)	✓	0.17	Correct. A distinguisher for F' gives a distinguisher for F.
□ $F'(k, x)=\{F(k,x)0_n$when $x\neq0_n$otherwise	✓	0.17	Not a PRF. A distinguisher will query at $x=0_n$ and output *not random* if the response is 0_n. This is unlikely to hold for a truly random function.
□ $F'(k,x)=F(k, x)\oplus F(k, x\oplus 1_n)$	✓	0.17	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.
☑ $F'((k_1,k_2), x)=\{F(k_1,x)k_2$when $x\neq0_n$otherwise	✓	0.17	Correct. A distinguisher for F' gives a distinguisher for F.
☑ $F'(k,x)=\text{reverse}(F(k,x))$ where reverse(y) reverses the string y so that the first bit of y is the last bit of reverse(y), the second bit of y is the second to last bit of reverse(y), and so on.	✓	0.17	Correct. A distinguisher for F' gives a distinguisher for F.
Total		1.00 / 1.00	

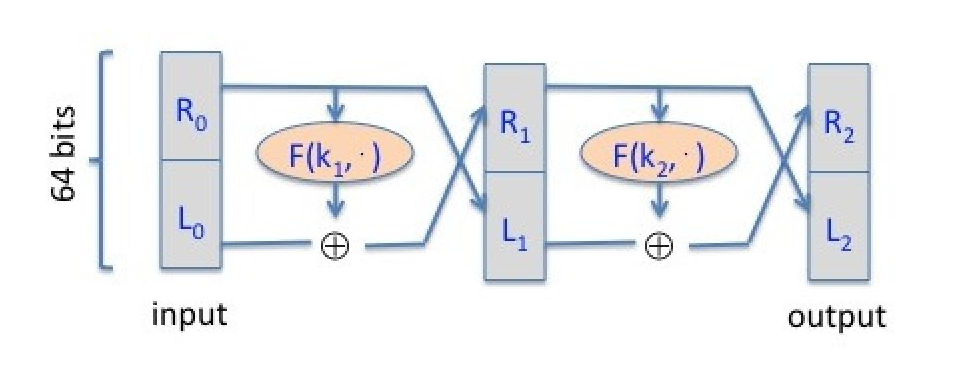
□ F′(k, x)={F(k,x)kwhen x≠0notherwise	✓	0.17	Not a PRF. A distinguisher will query at x=0n and obtain k and then query at $x=1^n$ and output not random if the response is F(k,1n). This is unlikely to hold for a truly random function.
□ F′(k,x)=F(k,x) ||| 0 (here |||denotes concatenation)	✓	0.17	Not a PRF. A distinguisher will output not random whenever the last bit of F(k,0n) is 0.
☑ F′(k,x)=reverse(F(k,x)) where reverse(y) reverses the string y so that the first bit of y is the last bit of reverse(y), the second bit of y is the second to last bit of reverse(y), and so on.	✓	0.17	Correct. A distinguisher for F′gives a distinguisher for F.
☑ F′((k1,k2), x)=F(k1,x) ||| F(k2,x) (here ||| denotes concatenation)	✓	0.17	Correct. A distinguisher for F′gives a distinguisher for F.
☑ F′((k1,k2), x)=F(k1,x)⊕F(k2,x)	✓	0.17	Correct. A distinguisher for F′gives a distinguisher for F.
☑ F′(k, x)=k⊕x	✗	0.00	Not a PRF. A distinguisher will query at x=0n and x=1n and output not random if the xor of the response is 1n. This is unlikely to hold for a truly random function.
k,x)=F(k, x)⊕F(k, x⊕1n)	✓	0.17	Not a PRF. A distinguisher will query at x=0n and x=1n and output not random whenever the two responses are equal. This is unlikely to happen for

			a truly random function.
□ F′(k, x)={F(k,x)kwhen x≠0notherwise	✓	0.17	Not a PRF. A distinguisher will query at x=0n and obtain k and then query at $x=1^n$ and output not random if the response is F(k,1n). This is unlikely to hold for a truly random function.
□ F′(k, x)={F(k,x)0nwhen x≠0notherwise	✓	0.17	Not a PRF. A distinguisher will query at x=0n and output not random if the response is 0n. This is unlikely to hold for a truly random function.
☑ F′(k,x)=F(k, x⊕1n)	✓	0.17	Correct. A distinguisher for F′ gives a distinguisher for F.
☑ F′(k,x)=F(k,x)[0,…,n−2] (i.e., F′(k,x) drops the last bit of F(k,x))	✓	0.17	Correct. A distinguisher for F′ gives a distinguisher for F.
□ F′((k1,k2), x)={F(k1,x)k2when x≠0notherwise	✗	0.00	Correct. A distinguisher for F′ gives a distinguisher for F.
F′(k, x)={F(k,x)0nwhen x≠0notherwise	✓	0.17	Not a PRF. A distinguisher will query at x=0n and output not random if the response is 0n. This is unlikely to hold for a truly random function.
☑ F′(k,x)=F(k,x) ‖‖ 0 (here ‖‖denotes concatenation)	✗	0.00	Not a PRF. A distinguisher will output not random whenever the last bit of F(k,0n) is 0.

☑ F′(k,x)=F(k,x)[0,…,n−2] (i.e., F′(k,x) drops the last bit of F(k,x))	✓	0.17	Correct. A distinguisher for F′gives a distinguisher for F.
□ F′(k, x)={F(k,x)kwhen x≠0notherwise	✓	0.17	Not a PRF. A distinguisher will query at x=0n and obtain k and then query at $x=1^n$ and output not random if the response is F(k,1n). This is unlikely to hold for a truly random function.
☑ F′((k1,k2), x)=F(k1,x)⊕F(k2,x)	✓	0.17	Correct. A distinguisher for F′gives a distinguisher for F.
☑ F′(k,x)=reverse(F(k,x)) where reverse(y) reverses the string y so that the first bit of y is the last bit of reverse(y), the second bit of y is the second to last bit of reverse(y), and so on.	✓	0.17	Correct. A distinguisher for F′gives a distinguisher for F.

Question 4

Recall that the Luby-Rackoff theorem discussed in Lecture 3.2 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. Let $F:K\times\{0,1\}32\longrightarrow\{0,1\}32$ be a secure PRF. Recall that a 2-round Feistel defines the following PRP $F2:K2\times\{0,1\}64\longrightarrow\{0,1\}64$:



Here $R0$ is the right 32 bits of the 64-bit input and $L0$ is the left 32 bits.

One of the following lines is the output of this PRP $F2$ using a random key, while the other three are the output of a truly random permutation $f:\{0,1\}^{64}\to\{0,1\}^{64}$. 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 $F_2(\cdot,1^{32}0^{32})$. Then try to detect this pattern in the given outputs.

Your Answer		Score	Explanation
● On input 0^{64} the output is "290b6e3a 39155d6f". On input $1^{32}0^{32}$ the output is "d6f491c5 b645c008".	✓	1.00	Observe that the two round Feistel has the property that the left of $F(\cdot,0^{64})\bigoplus F(\cdot,1^{32}0^{32})$ is 1^{32}. The two outputs in this answer are the only ones with this property.
Total		1.00 / 1.00	

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

Let $F:K\times\{0,1\}^{\ell}\to\{0,1\}^{\ell}$ 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 $c_1=F(k,c_0)$. Next, the attacker asks for the encryption of the one block message $m_1=c_0\oplus c_1$ 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.

Your Answer		Score	Explanation
◉ $c_1=c_0'$	✓	1.00	This follows from the definition of CBC with an encrypted nonce as defined in the question.
Total		1.00 / 1.00	

Question 6

Let m be a message consisting of ℓ AES blocks (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?

Your Answer		Score	Explanation
◉ 2	✓	1.00	Take a look at the CBC decryption circuit. Each ciphertext blocks affects only the current plaintext block and the next.
Total		1.00 / 1.00	

Question 7

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

Your Answer		Score	Explanation
● 1	✓	1.00	Take a look at the counter mode decryption circuit. Each ciphertext block affects only the current plaintext block.
Total		1.00 / 1.00	

Question 8

Recall that encryption systems do not fully hide the **length** of 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:

Your Answer		Score	Explanation
● 'In this letter I make some remarks on a general principle relevant to enciphering in general and my machine.'	✓	1.00	The length of the string is 107 bytes, which after padding becomes 112 bytes, and after prepending the IV becomes 128 bytes.
Total		1.00 / 1.00	

Question 9

Let $R:=\{0,1\}^4$ and consider the following PRF $F:R^5\times R\rightarrow R$ defined as follows:

$$F(k,x):=\left\{ t=k[0] \text{ for i=1 to 4 do} \text{if } (x[i-1]==1)t=t\oplus k[i] \text{ output } t \right.$$

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.

Answer for Question 9

Your Answer		Score	Explanation
1111	✓	1.00	
Total		1.00 / 1.00	

Question 1

Suppose a MAC system (S,V) is used to protect files in a file system by appending a MAC tag to each file. The MAC signing algorithm S is applied to the file contents and nothing else. What tampering attacks are not prevented by this system?

Your Answer		Score	Explanation
● Changing the last modification time of a file.	✓	1.00	The MAC signing algorithm is only applied to the file contents and does not protect the file meta data.
Total		1.00 / 1.00	

Question 2

Let (S,V) be a secure MAC defined over (K,M,T) where $M=\{0,1\}n$ and $T=\{0,1\}128$ (i.e. the key space is K, message space is $\{0,1\}n$, and tag space is $\{0,1\}128$). Which of the following is a secure MAC: (as usual, we use ‖‖ to denote string concatenation)

Your Answer		Score	Explanation
☑ $S'(k,m)=S(k,m)[0,\ldots,126]$ and$V'(k,m,t)=[V(k,\ m,\ t\|\|0)$ or $V(k,\ m,\ t\ \|\|1)$] (i.e., $V'(k,m,t)$ outputs ``1" if either $t\|\|0$ or $t\|\|1$ is a valid tag for m)	✓	0.17	a forger for (S',V') gives a forger for (S,V).
☐ $S'(k,m)=S(k,m)$ and $V'(k,m,t)=\{V(k,m,t)``1"\text{if } m\neq 0_n\text{otherwise}$	✓	0.17	This construction is insecure because the adversary can simply output $(0_n,0_s)$ as an existential forgery.
☐ $S'(k,m)=\{S(k,1_n)S(k,m)\text{if } m=0_n\text{otherwise}$ and $V'(k,m)=\{V(k,1_n,t)V(k,m,t)\text{if } m=0_n\text{otherwise}$	✓	0.17	This construction is insecure because an adversary can request the tag for the message $0_n$and output the result as a valid forgery for the message 1_n.
☑ $S'(k,m)=S(k,\ m\|\|m)$ and $V'(k,m,t)=V(k,\ m\|\|m,\ t)$.	✓	0.17	a forger for (S',V') gives a forger for (S,V).
☑	✓	0.17	a forger for (S',V') gives a forger

$S'((k_1,k_2), m)=(S(k_1,m),S(k_2,m))$ and $V'((k_1,k_2),m,(t_1,t_2))=[V(k_1,m,t_1)$ and $V(k_2,m,t_2)]$ (i.e., $V'((k_1,k_2),m,(t_1,t_2))$ outputs ``1" if both t_1 and t_2 are valid tags)			for (S,V).
□ $S'(k,m)=S(k,m\oplus m)$ and $V'(k,m,t)=V(k, m\oplus m, t)$	✓	0.17	This construction is insecure because an adversary can request the tag for $m=0_n$ and thereby obtain a tag for any message. This follows from the fact that $m\oplus m=0$.
Total		1.00 / 1.00	
☑ S′(k,m)=S(k,m)[0,…,126] andV′(k,m,t)=[V(k, m, t‖‖0) or V(k, m, t ‖‖1)] (i.e., V′(k,m,t) outputs ``1" if either t‖‖0 or t‖‖1 is a valid tag for m)	✓	0.17	a forger for (S′,V′) gives a forger for (S,V).
□ S′(k,m)=S(k,m⊕m) and V′(k,m,t)=V(k, m⊕m, t)	✓	0.17	This construction is insecure because an adversary can request the tag for m=0n and thereby obtain a tag for any message. This follows from the fact that m⊕m=0.

☑	$S'(k,m)=S(k, m\|\|\|\|m)$ and $V'(k,m,t)=V(k, m\|\|\|\|m, t)$.	✓	0.17	a forger for (S',V') gives a forger for (S,V).
□	$S'(k,m)=S(k, m[0,\ldots,n-2]\|\|\|\|0)$and $V'(k,m,t)=V(k, m[0,\ldots,n-2]\|\|\|\|0, t)$	✓	0.17	This construction is insecure because the tags on $m=0n$ and $m=0n-11$ are the same. Consequently, the attacker can request the tag on $m=0n$ and output an existential forgery for $m=0n-11$.
☑	$S'(k,m)=S(k,m\oplus 1n)$ and $V'(k,m,t)=V(k,m\oplus 1n,t)$.	✓	0.17	a forger for (S',V') gives a forger for (S,V).
☑	$S'(k,m)=(S(k,m),S(k,0n))$ and$V'(k,m,(t1,t2))=[V(k,m,t1)$ and $V(k,0n,t2)]$ (i.e., $V'(k,m,(t1,t2))$ outputs ``1" if both t1 and t2 are valid tags)	✗	0.00	This construction is insecure because the adversary can query for the tag of the message 1n and then obtain a valid tag for the message 0n. The adversary can then output an existential forgery for the message 0n.

Question 3

Recall that the ECBC-MAC uses a fixed IV (in the lecture we simply set the IV to 0). Suppose instead we chose a random IV for every message being signed and include the IV in the tag. In other words,$S(k,m):=(r,\ \mathrm{ECBC}_r(k,m))$ where $\mathrm{ECBC}_r(k,m)$ refers to the ECBC function using r as the IV. The verification algorithm V given key k, message m, and tag (r,t) outputs ``1'' if $t=\mathrm{ECBC}_r(k,m)$ and outputs ``0'' otherwise.

The resulting MAC system is insecure. An attacker can query for the tag of the 1-block message $m$and obtain the tag (r,t). He can then generate the following existential forgery: (we assume that the underlying block cipher operates on n-bit blocks)

Your Answer		Score	Explanation
◉ The tag $(r\oplus 1_n,\ t)$ is a valid tag for the 1-block message $m\oplus 1_n$.	✓	1.00	The CBC chain initiated with the IV $r\oplus m$ and applied to the message 0_n will produce exactly the same output as the CBC chain initiated with the IV r and applied to the message m. Therefore, the tag $(r\oplus 1_n,\ t)$ is a valid existential forgery for the message $m\oplus 1_n$.
Total		1.00 / 1.00	

Question 4

Suppose Alice is broadcasting packets to 6 recipients $B_1,\ldots,B_6$. Privacy is not important but integrity is. In other words, each of $B_1,\ldots,B_6$ should be assured that the packets he is receiving were sent by Alice.

Alice decides to use a MAC. Suppose Alice and $B_1,\ldots,B_6$ all share a secret key k. Alice computes a tag for every packet she sends using key k. Each user B_i verifies the tag when receiving the packet and drops the packet if the tag is invalid. Alice notices that this scheme is insecure because user B_1can use the key k to send packets with a valid tag to users $B_2,\ldots,B_6$ and they will all be fooled into thinking that these packets are from Alice.

Instead, Alice sets up a set of 4 secret keys $S=\{k_1,\ldots,k_4\}$. She gives each user B_i some subset $S_i\subseteq S$ of the keys. When Alice transmits a packet she appends 4 tags to it by computing the tag with each of her 4 keys. When user B_i receives a packet he accepts it as valid only if all tags corresponding to his keys in S_i are valid. For example, if user B_1 is given keys $\{k_1,k_2\}$ he will accept an incoming packet only if the first and second tags are valid. Note that B_1 cannot validate the 3rd and 4th tags because he does not have k_3 or k_4.

How should Alice assign keys to the 6 users so that no single user can forge packets on behalf of Alice and fool some other user?

Your Answer		Score	Explanation
● $S_1=\{k_2,k_4\}$, $S_2=\{k_2,k_3\}$, $S_3=\{k_3,k_4\}$, $S_4=\{k_1,k_3\}$, $S_5=\{k_1,k_2\}$, $S_6=\{k_1,k_4\}$	✓	1.00	Every user can only generate tags with the two keys he has. Since no set S_i is contained in another set S_j, no user i can fool a user j into accepting a message sent by i.
Total		1.00 / 1.00	

Question 5

Consider the encrypted CBC MAC built from AES. Suppose we compute the tag for a long message m comprising of n AES blocks. Let m' be the n-block message obtained from m by flipping the last bit of m (i.e. if the last bit of m is b then the last bit of m' is $b\oplus 1$). How many calls to AES would it take to compute the tag for m' from the tag for m and the MAC key? (in this question please ignore message padding and simply assume that the message length is always a multiple of the AES block size)

Your Answer	Score	Explanation

● 4	✓	1.00	You would decrypt the final CBC MAC encryption step done using k_2, the decrypt the last CBC MAC encryption step done using k_1, flip the last bit of the result, and re-apply the two encryptions.
Total		1.00 / 1.00	

Question 6

Let $H:M\to T$ be a collision resistant hash function. Which of the following is collision resistant: (as usual, we use ‖ to denote string concatenation)

Your Answer		Score	Explanation
□ $H'(m)=H(m)\oplus H(m\oplus 1^{\vert m\vert})$ (where $m\oplus 1^{\vert m\vert}$ is the complement of m)	✓	0.14	This construction is not collision resistant because $H(000)=H(111)$.
☑ $H'(m)=H(H(H(m)))$	✓	0.14	a collision finder for H' gives a collision finder for H.
☑ $H'(m)=H(m\|\|0)$	✓	0.14	a collision finder for H' gives a collision finder for H.
□ $H'(m)=H(\vert m\vert)$ (i.e. hash the length of m)	✓	0.14	This construction is not collision resistant because $H(000)=H(111)$.
□ $H'(m)=H(m)[0,\ldots,31]$ (i.e. output the first 32 bits of the hash)	✓	0.14	This construction is not collision resistant because an attacker can find a collision in time 2^{16} using the birthday paradox.
☑ $H'(m)=H(H(m))$	✓	0.14	a collision finder for H' gives a collision finder for H.

□ $H'(m)=H(m[0,\ldots,\|m\|-2])$ (i.e. hash m without its last bit)	✓	0.14	This construction is not collision resistant because $H(00)=H(01)$.
Total		1.00 / 1.00	
☑ H′(m)=H(m)[0,…,31] (i.e. output the first 32 bits of the hash)	✗	0.00	This construction is not collision resistant because an attacker can find a collision in time 216 using the birthday paradox.
☑ H′(m)=H(m)‖‖H(0)	✓	0.14	a collision finder for H′ gives a collision finder for H.
□ H′(m)=H(0)	✓	0.14	This construction is not collision resistant because H(0)=H(1).
□ H′(m)=H(|m|) (i.e. hash the length of m)	✓	0.14	This construction is not collision resistant because H(000)=H(111).
☑ H′(m)=H(m‖‖m)	✓	0.14	a collision finder for H′ gives a collision finder for H.
☑ H′(m)=H(H(H(m)))	✓	0.14	a collision finder for H′ gives a collision finder for H.
□ H′(m)=H(m[0,…,|m|−2]) (i.e. hash m without its last bit)	✓	0.14	This construction is not collision resistant because H(00)=H(01).

Question 7

Suppose $H1$ and $H2$ are collision resistant hash functions mapping inputs in a set M to $\{0,1\}256$. Our goal is to show that the function $H2(H1(m))$ is also collision resistant. We prove the contra-positive:

suppose $H_2(H_1(\cdot))$ is not collision resistant, that is, we are given $x \neq y$ such that $H_2(H_1(x)) = H_2(H_1(y))$. We build a collision for either H_1 or for H_2. This will prove that if $H_1$and H_2 are collision resistant then so is $H_2(H_1(\cdot))$. Which of the following must be true:

Your Answer		Score	Explanation
● Either x,y are a collision for H_1 or$H_1(x),H_1(y)$are a collision for H_2.	✓	1.00	If $H_2(H_1(x))=H_2(H_1(y))$ then either $H_1(x)=H_1(y)$ and $x \neq y$, thereby giving us a collision on H_1. Or $H_1(x) \neq H_1(y)$ but $H_2(H_1(x))=H_2(H_1(y))$ giving us a collision on H_2. Either way we obtain a collision on H_1 or H_2 as required.
Total		1.00 / 1.00	

Question 8

In this question and the next, you are asked to find collisions on two compression functions:

- $f_1(x,y) = \mathrm{AES}(y,x) \oplus y,$ and
- $f_2(x,y) = \mathrm{AES}(x,x) \oplus y,$

where $\mathrm{AES}(x,y)$ is the AES-128 encryption of y under key x.

We provide an AES function for you to play with. The function takes as input a key k and an x value and outputs $\mathrm{AES}(k,x)$ once you press the "encrypt" button. It takes as input a key k and a y value and outputs $\mathrm{AES}^{-1}(k,y)$ once you press the "decrypt" button. All three values k,x,y are assumed to be hex values (i.e. using only characters 0-9 and a-f) and the function zero-pads them as needed.

Your goal is to find four distinct pairs $(x_1,y_1),\ (x_2,y_2),\ (x_3,y_3),\ (x_4,y_4)$ such that$f_1(x_1,y_1)=f_1(x_2,y_2)$ and $f_2(x_3,y_3)=f_2(x_4,y_4)$. In other words, the first two pairs are a collision for f_1 and the last two pairs are a collision for f_2. Once you find all four pairs, please enter them below and check your answer using the "check" button.

Note for those using the NoScript browser extension: for the buttons to function correctly please allow Javascript from class.coursera.org and cloudfront.net to run in your browser. Note also that the "save answers" button does not function for this question and the next.

Answer for Question 8

Your Answer		Score	Explanation
x1 = 22222222222222222222222222222222 y1 = 11111111111111111111111111111111 x2 = 7e8bf0a7b1bf082a76c2415385ed7434 y2 = 33333333333333333333333333333333	✓	1.00	You got it !
Total		1.00 / 1.00	

Question 9

Answer for Question 9

Your Answer		Score	Explanation
x3 = 11111111111111111111111111111111 y3 = 33333333333333333333333333333333 x4 = 22222222222222222222222222222222 y4 = 90dac026e18053a5bb1c5902afedc83b	✓	1.00	Awesome!
Total		1.00 / 1.00	

Question 10

Let $H:M\to T$ be a random hash function where $|M|\gg|T|$ (i.e. the size of M is much larger than the size of T). In lecture we showed that finding a collision on H can be done with $O(|T|^{1/2})$random samples of H. How many random samples would it take until we obtain a three way collision, namely distinct strings x,y,z in M such that $H(x)=H(y)=H(z)$?

Your Answer	Score	Explanation

●	$O(\|T\|^{2/3})$	✓	1.00	An informal argument for this is as follows: suppose we collect $n$random samples. The number of triples among the n samples is $n$choose 3 which is $O(n^3)$. For a particular triple x,y,z to be a 3-way collision we need $H(x)=H(y)$ and $H(x)=H(z)$. Since each one of these two events happens with probability $1/\|T\|$ (assuming H behaves like a random function) the probability that a particular triple is a 3-way collision is $O(1/\|T\|^2)$. Using the union bound, the probability that some triple is a 3-way collision is $O(n^3/\|T\|^2)$ and since we want this probability to be close to 1, the bound on $n$follows.

Total	1.00 / 1.00

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.

Answer for Question 1

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 ∥ to denote string concatenation)

Your Answer		Score	Explanation
☑ $E'((k_1,k_2),m)=E(k_2,\ E(k_1,m))$ and $D'((k_1,k_2),\ c)=\{D(k_1,D(k_2,c)) \perp \text{if } D(k_2,c)\neq\perp \text{otherwise}$	✓	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').
☑ $E'(k,m)=(E(k,m),\ 0)$ and $D'(k,\ (c,b))=\{D(k,c) \perp \text{if } b=0 \text{otherwise}$	✓	0.25	(E',D')provides authenticated encryption because an attack on (E',D')directly gives an attack on (E,D).
□ $E'(k,m)=E(k,m)$ and $D'(k,c)=\{D(k,c) 0^n \text{if } D(k,c)\neq\perp \text{otherwise}$	✓	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.
□ $E'(k,m)=(E(k,m),\ 0)$ 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.
Total		1.00 / 1.00	
☑ E′(k,m)=E(k,m) and D′(k,c)={D(k,c)0nif D(k,c)≠⊥otherwise	✗	0.00	This system does not provide ciphertext integrity since an attacker can simply output the ciphertext 0sand win the ciphertext integrity game.
☑ E′((k1,k2),m)=E(k2, E(k1,m)) and D′((k1,k2), c)={D(k1,D(k2,c))⊥if D(k2,c)≠⊥otherwise	✓	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′).
☑ E′(k,m)=E(k,m)⊕1s and D′(k,c)=D(k,c⊕1s)	✓	0.25	(E′,D′)provides authenticated encryption because an attack on (E′,D′)directly gives an attack on (E,D).
☐ 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,0n) to obtain (c,0). It then outputs (c,1) and wins the ciphertext integrity game.

Your Answer		Score	Explanation
☑ $E'(k,m)=E(k,m\oplus 1n)$ and $D'(k,c)=\{D(k,c)\oplus 1n \perp \text{if } D(k,c)\neq\perp \text{otherwise}$	✓	0.25	(E',D') provides authenticated encryption because an attack on (E',D')directly gives an attack on (E,D).
☑ $E'(k,m)=E(k,m)\oplus 1s$ and $D'(k,c)=D(k,c\oplus 1s)$	✓	0.25	(E',D') provides authenticated encryption because an attack on (E',D')directly gives an attack on (E,D).
□ $E'(k,m)=(E(k,m),0)$ and $D'(k,(c,b))=D(k,c)$	✓	0.25	This system does not provide ciphertext integrity. The attacker queries for $E'(k,0n)$to obtain $(c,0)$. It then outputs $(c,1)$ and wins the ciphertext integrity game.
□ $E'(k,m)=(E(k,m),E(k,m))$ and $D'(k,(c1,c2))=\{D(k,c1)\perp \text{if } D(k,c1)=D(k,c2)\text{otherwise}$	✓	0.25	This system does not provide ciphertext integrity. To see why, recall that authenticated encryption (without a nonce) must be randomized to provide CPA security. Therefore, $E'(k,m)=(c1,c2)$will likely output a distinct ciphertext pair $c1\neq c2$. The attacker can then output the ciphertext $(c1,c1)$and win the ciphertext integrity game.

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
◉ use a standard implementation of one of the authenticated encryption modes GCM, CCM, EAX or OCB.	✓	1.00	
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:

$$S(k,m) := E(k,m); V(k,m,t) := \{ 10 \text{if } D(k,t) = m \text{otherwise}$$

What is the property that the encryption system (E,D) needs to satisfy for this MAC system to be secure?

Your Answer		Score	Explanation
◉ 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,

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

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?

Your Answer		Score	Explanation
● $F'(k,x) = \begin{cases} F(k,x) & \text{if } \mathrm{MSB}_{128}(k) \neq 0^{128} \\ 1^{256} & \text{otherwise} \end{cases}$	✓	1.00	$F'(k,x)$ is a secure PRF because for a uniform key $k$the probability that $\mathrm{MSB}_{128}(k) = 0^{128}$ 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.
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
● when messages are chosen at random	✓	1.00	Deterministic encryption is safe to use when the message/key pair is never used

from a large enough space so that messages are unlikely to repeat.		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
● 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.
Total		1.00 / 1.00	

Question 8

In lecture 8.5 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}?

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

Question 9

Let (E,D) be a secure tweakable block cipher. Define the following MAC (S,V):

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

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
◉ 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.
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
● 12240 ✓	1.00	Correct. Padding oracle attacks decrypt the payload one byte at a time. For each byte the attacker needs 255 guesses in the worst case. Since there are 48 bytes total, the number queries needed is $255\times48=12240$.
Total	1.00 / 1.00	

Question 1

Consider the toy key exchange protocol using an online trusted 3rd party (TTP) discussed in Lecture 9.1. Suppose Alice, Bob, and Carol are three users of this system (among many others) and each have a secret key with the TTP denoted k_a,k_b,k_c respectively. They wish to generate a group session key k_{ABC} that will be known to Alice, Bob, and Carol but unknown to an eavesdropper. How would you modify the protocol in the lecture to accomodate a group key exchange of this type? (note that all these protocols are insecure against active attacks)

Your Answer	Score	Explanation
● Alice contacts the TTP. TTP generates random k_{ABC} and sends to Alice $E(k_a,k_{ABC}),\text{ticket}_1\leftarrow E(k_b,k_{ABC}),\text{ticket}_2\leftarrow E(k_c,k_{ABC})$. Alice sends ticket_1 to Bob and ticket_2 to Carol. ✓	1.00	The protocol works because it lets Alice, Bob, and Carol obtain k_{ABC} but an eaesdropper only sees encryptions of k_{ABC}under keys he does not have.
Total	1.00 / 1.00	

Question 2

Let G be a finite cyclic group (e.g. $G=\mathbb{Z}_p^*$) with generator g. Suppose the Diffie-Hellman function $\mathrm{DH}_g(g^x,g^y)=g^{xy}$ is difficult to compute in G. Which of the following functions is also difficult to compute:

As usual, identify the f below for which the contra-positive holds: if $f(\cdot,\cdot)$ is easy to compute then so is $\mathrm{DH}_g(\cdot,\cdot)$. If you can show that then it will follow that if DH_g is hard to compute in G then so must be f.

Your Answer		Score	Explanation
□ $f(g^x,g^y)=g^{x-y}$	✓	0.25	It is easy to compute f as $f(g^x,g^y)=g^x/g^y$.
☑ $f(g^x,g^y)=\sqrt{g^{xy}}$	✓	0.25	an algorithm for calculating $f(g^x,g^y)=\pm g^{xy/2}$can easily be converted into an algorithm for calculating $\mathrm{DH}(\cdot,\cdot)$. Therefore, if f were easy to compute then so would DH, contrading the assumption.
☑ $f(g^x,g^y)=g^{xy+x+y+1}$	✓	0.25	an algorithm for calculating $f(g^x,g^y)$ can easily be converted into an algorithm for calculating $\mathrm{DH}(\cdot,\cdot)$. Therefore, if f were easy to compute then so would DH, contrading the assumption.
□ $f(g^x,g^y)=g^{x+y}$	✓	0.25	It is easy to compute f as $f(g^x,g^y)=g^x\cdot g^y$.
Total		1.00 / 1.00	

Question 3

Suppose we modify the Diffie-Hellman protocol so that Alice operates as usual, namely chooses a random a in $\{1,\ldots,p-1\}$ and sends to Bob $A\leftarrow g^a$. Bob, however, chooses a random b in $\{1,\ldots,p-1\}$ and sends to Alice $B\leftarrow g^{1/b}$. What shared secret can they generate and how would they do it?

Your Answer	Score	Explanation

● secret=$g^{a/b}$. Alice computes the secret as B^a and Bob computes $A^{1/b}$.	✓	1.00	This is correct since it is not difficult to see that both will obtain $g^{a/b}$
Total		1.00 / 1.00	

Question 4

Consider the toy key exchange protocol using public key encryption described in Lecture 9.4. Suppose that when sending his reply $c \leftarrow E(pk,x)$ to Alice, Bob appends a MAC $t:=S(x,c)$ to the ciphertext so that what is sent to Alice is the pair (c,t). Alice verifies the tag t and rejects the message from Bob if the tag does not verify. Will this additional step prevent the man in the middle attack described in the lecture?

Your Answer		Score	Explanation
● no	✓	1.00	an active attacker can still decrypt $E(pk',x)$ to recover x and then replace (c,t) by (c',t') where $c' \leftarrow E(pk,x)$ and $t \leftarrow S(x,c')$.
Total		1.00 / 1.00	

Question 5

The numbers 7 and 23 are relatively prime and therefore there must exist integers a and b such that $7a+23b=1$. Find such a pair of integers (a,b) with the smallest possible $a>0$. Given this pair, can you determine the inverse of 7 in Z_{23}?

Enter below comma separated values for a, b, and for 7^{-1} in Z_{23}.
Answer for Question 5

Your Answer		Score	Explanation
10, -3, 10	✓	1.00	

Total		1.00 / 1.00	

$7\times10+23\times(-3)=1$. Therefore $7\times10=1$ in $\mathbb{Z}_{23}$ implying that $7^{-1}=10$ in $\mathbb{Z}_{23}$.

Question 6

Solve the equation $3x+2=7$ in $\mathbb{Z}_{19}$.
Answer for Question 6

Your Answer		Score	Explanation
8	✓	1.00	
Total		1.00 / 1.00	

$x=(7-2)\times3^{-1}\in\mathbb{Z}_{19}$

Question 7

How many elements are there in $\mathbb{Z}^*_{35}$?
Answer for Question 7

Your Answer		Score	Explanation
24	✓	1.00	
Total		1.00 / 1.00	

$|\mathbb{Z}^*_{35}|=\varphi(7\times5)=(7-1)\times(5-1)$.

Question 8

How much is $2^{10001} \bmod 11$? (please do not use a calculator for this)

Hint: use Fermat's theorem.
Answer for Question 8

Your Answer		Score	Explanation
2	✓	1.00	
Total		1.00 / 1.00	

By Fermat $2^{10}=1$ in Z_{11} and therefore $1=2^{10}=2^{20}=2^{30}=2^{40}$ in Z_{11}.
Then $2^{10001}=2^{10001 \bmod 10}=2^{1}=2$ in Z_{11}.

Question 9

While we are at it, how much is $2^{245} \bmod 35$?

Hint: use Euler's theorem (you should not need a calculator)
Answer for Question 9

Your Answer		Score	Explanation
32	✓	1.00	
Total		1.00 / 1.00	

By Euler $2^{24}=1$ in Z_{35} and therefore $1=2^{24}=2^{48}=2^{72}$ in Z_{35}.
Then $2^{245}=2^{245 \bmod 24}=2^{5}=32$ in Z_{35}.

Question 10

What is the order of 2 in Z^{*}_{35}?
Answer for Question 10

Your Answer		Score	Explanation
12	✓	1.00	
Total		1.00 / 1.00	

$2^{12}=4096=1$ in $\mathbb{Z}_{35}$ and 12 is the smallest such positive integer.

Question 11

Which of the following numbers is a generator of $\mathbb{Z}^*_{13}$?

Your Answer		Score	Explanation
☑ $6, \langle 6 \rangle = \{1,6,10,8,9,2,12,7,3,5,4,11\}$	✓	0.20	correct, 6 generates the entire group $\mathbb{Z}^*_{13}$
□ $9, \langle 9 \rangle = \{1,9,3\}$	✓	0.20	No, 9 only generates three elements in $\mathbb{Z}^*_{13}$.
□ $5, \langle 5 \rangle = \{1,5,12,8\}$	✓	0.20	No, 5 only generates four elements in $\mathbb{Z}^*_{13}$.
□ $8, \langle 8 \rangle = \{1,8,12,5\}$	✓	0.20	No, 8 only generates four elements in $\mathbb{Z}^*_{13}$.
☑ $7, \langle 7 \rangle = \{1,7,10,5,9,11,12,6,3,8,4,2\}$	✓	0.20	correct, 7 generates the entire group $\mathbb{Z}^*_{13}$
Total		1.00 / 1.00	

Question 12

Solve the equation $x^2+4x+1=0$ in $\mathbb{Z}_{23}$. Use the method described in lecture 9.3 using the quadratic formula.

Answer for Question 12

Your Answer	Score	Explanation

5, 14	✓	1.00	
Total		1.00 / 1.00	

The quadratic formula gives the two roots in $\mathbb{Z}_{23}$.

Question 13

What is the 11th root of 2 in $\mathbb{Z}_{19}$? (i.e. what is $2^{1/11}$ in $\mathbb{Z}_{19}$)

Hint: observe that $11^{-1}=5$ in $\mathbb{Z}_{18}$.

Answer for Question 13

Your Answer		Score	Explanation
13	✓	1.00	
Total		1.00 / 1.00	

$2^{1/11}=2^5=32=13$ in $\mathbb{Z}_{19}$.

Question 14

What is the discete log of 5 base 2 in $\mathbb{Z}_{13}$? (i.e. what is $\mathrm{Dlog}_2(5)$)

Recall that the powers of 2 in $\mathbb{Z}_{13}$ are $\langle 2\rangle=\{1,2,4,8,3,6,12,11,9,5,10,7\}$

Answer for Question 14

Your Answer		Score	Explanation
9	✓	1.00	
Total		1.00 / 1.00	

$2^9=5$ in $\mathbb{Z}_{13}$.

Question 15

If p is a prime, how many generators are there in Z^*_p?

Your Answer		Score	Explanation
● $\varphi(p-1)$	✓	1.00	The answer is $\varphi(p-1)$. Here is why. Let g be some generator of $Z^*_p$and let $h=g^x$ for some x. It is not difficult to see that h is a generator exactly when we can write g as $g=h^y$ for some integer y (h is a generator because if $g=h^y$ then any power of g can also be written as a power of h). Since $y=x^{-1} \bmod p-1$ this y exists exactly when x is relatively prime to $p-1$. The number of such x is the size of Z_{p-1} which is precisely $\varphi(p-1)$.
Total		1.00 / 1.00	

Question 1

Recall that with symmetric ciphers it is possible to encrypt a 32-bit message and obtain a 32-bit ciphertext (e.g. with the one time pad or with a nonce-based system). Can the same be done with a public-key system?

Your Answer		Score	Explanation
● No, public-key systems with short ciphertexts can never be secure.	✓	1.00	An attacker can use the public key to build a dictionary of all 2^{32} ciphertexts of length 32 bits along with their decryption and use the dictionary to decrypt any captured ciphertext.
Total		1.00 / 1.00	
● Yes, the RSA-OAEP system can produce 32-bit ciphertexts.	✗	0.00	No, RSA-OAEP produces ciphertexts that are at least as long as the modulus.

Question 2

Let (Gen,E,D) be a semantically secure public-key encryption system. Can algorithm E be deterministic?

Your Answer	Score	Explanation
● No, semantically secure public-key encryption must be randomized. ✓	1.00	That's correct since otherwise an attacker can easily break semantic security.
Total	1.00 / 1.00	

Question 3

Let (Gen,E,D) be a chosen ciphertext secure public-key encryption system with message space $\{0,1\}^{128}$. Which of the following is also chosen ciphertext secure?

Your Answer	Score	Explanation
☑ (Gen,E',D') where $E'(\mathrm{pk},m)=E(\mathrm{pk},m\oplus 1^{128})$ and $D'(\mathrm{sk},c)=D(\mathrm{sk},c)\oplus 1^{128}$ ✓	0.25	This construction is chosen-ciphertext secure. An attack on (Gen,E',D)gives an attack on (Gen,E,D).
□ (Gen,E',D') where $E'(\mathrm{pk},m)=(E(\mathrm{pk},m),0^{128})$ and $D'(\mathrm{sk},(c_1,c_2))=D(\mathrm{sk},c_1)$. ✓	0.25	This construction is not chosen-ciphertext secure. An attacker can output two messages $m_0=0^{128}$ and $m_1=1^{128}$ and be given back a challenge ciphertext (c_1,c_2). The attacker would then ask for the decryption of $(c_1,1^{128})$ and be given in response m_0 or m_1 the

			reby letting the attacker win the game. Note that the decryption query is valid since it is different from the challenger ciphertext (c_1,c_2).
☑ (Gen,E',D') where $E'(\text{pk},m)=(E(\text{pk}, m), 0^{128})$ and $D'(\text{sk}, (c_1, c_2))=\begin{cases} D(\text{sk},c_1) & \text{if } c_2=0^{128} \\ \perp & \text{otherwise}\end{cases}$.	✓	0.25	This construction is chosen-ciphertext secure. An attack on (Gen,E',D) gives an attack on (Gen,E,D).
☐ (Gen,E',D') where $E'(\text{pk},m)=(E(\text{pk}, m), E(\text{pk}, 0^{128}))$ and $D'(\text{sk}, (c_1,c_2))=\begin{cases} D(\text{sk},c_1) & \text{if } D(\text{sk},c_2)=0^{128} \\ \perp & \text{otherwise}\end{cases}$.	✓	0.25	This construction is not chosen-ciphertext secure. An attacker can output two messages $m_0=0^{128}$ and $m_1=1^{128}$ and be given back a challenge ciphertext (c_1,c_2). He would then ask for the decryption of (c_2,c_1), which is a valid decryption query since it is different from the challenge ciphertext with high probability. The response is either 0^{128} or $\perp$ depending on the contents of the challenge ciphertext and this lets the attacker win the game.
Total		1.00 / 1.00	

☑ (Gen,E′,D′) where E′(pk,m)=(E(pk, m), 0128) andD′(sk, (c1,c2))={D(sk,c1)⊥if c2=0128 otherwise.	✓	0.25	This construction is chosen-ciphertext secure. An attack on (Gen,E′,D)gives an attack on (Gen,E,D).
☑ (Gen,E′,D′) where E′(pk,m)=(E(pk, m), E(pk, m)) and D′(sk, (c1,c2))={D(sk,c1)⊥if D(sk,c1)=D(sk,c2)otherwise.	✗	0.00	This construction is not chosen-ciphertext secure. An attacker can output two messages m0=0128and m1=1128and be given back a challenge ciphertext (c1,c2). He would then, on his own, create a new random encryption of m0, call it c3, and ask for the decryption of (c1,c3), which is a valid decryption query since it is different from the challenge ciphertext with high probability. The response is either m0 or ⊥ depending on the contents of the challenge ciphertext and this lets the attacker win the game.
□ (Gen,E′,D′) where E′(pk,m)=(E(pk, m), E(pk, 0128)) and D′(sk, (c1,c2))={D(sk,c1)⊥if D(sk,c2)=0128otherwise.	✓	0.25	This construction is not chosen-ciphertext secure. An attacker can output two messages m0=0128and m1=1128and be given back a challenge ciphertext (c1,c2). He would then ask for the decryption of (c2,c1),

			which is a valid decryption query since it is different from the challenge ciphertext with high probability. The response is either 0128 or ⊥ depending on the contents of the challenge ciphertext and this lets the attacker win the game.
☑ (Gen,E′,D′) where E′(pk,m)=[c←E(pk, m), output (c,c)] and D′(sk, (c1,c2))={D(sk, c1)⊥if c1=c2otherwise.	✓	0.25	This construction is chosen-ciphertext secure. An attack on (Gen,E′,D)gives an attack on (Gen,E,D).
☑ (Gen,E′,D′) where E′(pk,m)=[c←E(pk, m), output (c,c)] and D′(sk, (c1,c2))={D(sk, c1)⊥if c1=c2otherwise.	✓	0.25	This construction is chosen-ciphertext secure. An attack on (Gen,E′,D)gives an attack on (Gen,E,D).
☑ (Gen,E′,D′) where E′(pk,m)=(E(pk, m), 0128) and D′(sk, (c1,c2))={D(sk,c1)⊥if c2=0128 otherwise.	✓	0.25	This construction is chosen-ciphertext secure. An attack on (Gen,E′,D)gives an attack on (Gen,E,D).
□ (Gen,E′,D′) where E′(pk,m)=(E(pk, m), E(pk, 0128)) and D′(sk, (c1,c2))={D(sk,c1)⊥if D(sk,c2)=0128otherwise.	✓	0.25	This construction is not chosen-ciphertext secure. An attacker can output two messages m0=0128 and m1=1128 and be given back a challenge ciphertext (c1,c2). He

			would then ask for the decryption of (c2,c1), which is a valid decryption query since it is different from the challenge ciphertext with high probability. The response is either 0128 or ⊥depending on the contents of the challenge ciphertext and this lets the attacker win the game.
☐ (Gen,E′,D′) where E′(pk,m)=(E(pk, m), 0128) and D′(sk, (c1,c2))=D(sk,c1).	✓	0.25	This construction is not chosen-ciphertext secure. An attacker can output two messages m0=0128 and m1=1128 and be given back a challenge ciphertext (c1,c2). The attacker would then ask for the decryption of (c1,1128) and be given in response m0 or m1 thereby letting the attacker win the game. Note that the decryption query is valid since it is different from the challenger ciphertext (c1,c2).
☑ (Gen,E′,D′) where E′(pk,m)=E(pk, m⊕1128) and D′(sk,c)=D(sk,c)⊕1128	✓	0.25	This construction is chosen-ciphertext secure. An attack on (Gen,E′,D)gives an attack on (Gen,E,D).

☑ (Gen,E′,D′) where E′(pk,m)=(E(pk, m), E(pk, 0128)) and D′(sk, (c1,c2))={D(sk,c1)⊥if D(sk,c2)=0128otherwise.	✗	0.00	This construction is not chosen-ciphertext secure. An attacker can output two messages m0=0128 and m1=1128 and be given back a challenge ciphertext (c1,c2). He would then ask for the decryption of (c2,c1), which is a valid decryption query since it is different from the challenge ciphertext with high probability. The response is either 0128 or ⊥depending on the contents of the challenge ciphertext and this lets the attacker win the game.
□ (Gen,E′,D′) where E′(pk,m)=(E(pk, m), E(pk, 0128)) and D′(sk, (c1,c2))=D(sk,c1).	✓	0.25	This construction is not chosen-ciphertext secure. An attacker can output two messages m0=0128 and m1=1128 and be given back a challenge ciphertext (c1,c2). The attacker would then ask for the decryption of (c1,E(pk,1128)and be given in response m0 or m1 thereby letting the attacker win the game. Note that the decryption query is valid since it is different from the challenger ciphertext (c1,c2).

☑ (Gen,E',D') where $E'(pk,m)=[c\leftarrow E(pk,m),\ \text{output } (c,c)]$ and $D'(sk,(c_1,c_2))=\begin{cases}D(sk,c_1) & \text{if } c_1=c_2\\ \bot & \text{otherwise}\end{cases}$.	✓	0.25	This construction is chosen-ciphertext secure. An attack on (Gen,E',D)gives an attack on (Gen,E,D).

Question 4

Recall that an RSA public key consists of an RSA modulus N and an exponent e. One might be tempted to use the same RSA modulus in different public keys. For example, Alice might use $(N,3)$as her public key while Bob may use $(N,5)$ as his public key. Alice's secret key is $d_a=3^{-1} \bmod \varphi(N)$ and Bob's secret key is $d_b=5^{-1} \bmod \varphi(N)$.

In this question and the next we will show that it is insecure for Alice and Bob to use the same modulus$N$. In particular, we show that either user can use their secret key to factor N. Alice can use the factorization to compute $\varphi(N)$ and then compute Bob's secret key.

As a first step, show that Alice can use her public key $(N,3)$ and private key d_a to construct an integer multiple of $\varphi(N)$. Which of the following is an integer multiple of $\varphi(N)$?

Your Answer		Score	Explanation
● $3d_a-1$	✓	1.00	Since $d_a=3^{-1} \bmod \varphi(N)$ we know that $3d_a=1 \bmod \varphi(N)$ and therefore $3d_a-1$ is divisibly by $\varphi(N)$.
Total		1.00 / 1.00	

Question 5

Now that Alice has a multiple of $\varphi(N)$ let's see how she can factor $N=pq$. Let x be the given muliple of $\varphi(N)$. Then for any g in $\mathbb{Z}^*_N$ we have $g^x=1$ in $\mathbb{Z}_N$. Alice chooses a random g in $\mathbb{Z}^*_N$ and computes the sequence

$g^x, g^{x/2}, g^{x/4}, g^{x/8} \ldots$ in $\mathbb{Z}_N$

and stops as soon as she reaches the first element $y = g^{x/2^i}$ such that $y \neq 1$ (if she gets stuck because the exponent becomes odd, she picks a new random g and tries again). It can be shown that with probability $1/2$ this y satisfies

$$\{y = 1 \bmod p, \text{ and} y = -1 \bmod q \text{or} \{y = -1 \bmod p, \text{ and} y = 1 \bmod q$$

How can Alice use this y to factor N?

Your Answer		Score	Explanation
● compute $gcd(N, y-1)$	✓	1.00	We know that $y-1$ is divisible by p or q, but not divisible by the other. Therefore, $gcd(N, y-1)$ will output a non-trivial factor of N.
Total		1.00 / 1.00	

Question 6

In standard RSA the modulus N is a product of two distinct primes. Suppose we choose the modulus so that it is a product of three distinct primes, namely $N = pqr$. Given an exponent e relatively prime to $\varphi(N)$ we can derive the secret key as $d = e^{-1} \bmod \varphi(N)$. The public key (N, e) and secret key(N, d) work as before. What is $\varphi(N)$ when N is a product of three distinct primes?

Your Answer		Score	Explanation
● $\varphi(N) = (p-1)(q-1)(r-1)$	✓	1.00	When is a product of distinct primes then $\lvert \mathbb{Z}^*_N \rvert$ satisfies$\lvert \mathbb{Z}^*_N \rvert = \lvert \mathbb{Z}^*_p \rvert \cdot \lvert \mathbb{Z}^*_q \rvert \cdot \lvert \mathbb{Z}^*_r \rvert = (p-1)(q-1)(r-1)$.
Total		1.00 / 1.00	

Question 7

An administrator comes up with the following key management scheme: he generates an RSA modulus N and an element s in $\mathbb{Z}_N^*$. He then gives user number i the secret key $s_i = s^{r_i}$ in $\mathbb{Z}_N$where r_i is the r'th prime (i.e. 2 is the first prime, 3 is the second, and so on).

Now, the administrator encrypts a file that is accssible to users i,j and t with the key $k = s^{r_i r_j r_t}$ in $\mathbb{Z}_N$. It is easy to see that each of the three users can compute k. For example, user i computes k as $k = (s_i)^{r_j r_t}$. The administrator hopes that other than users i,j and t, no other user can compute $k$and access the file.

Unfortunately, this system is terribly insecure. Any two colluding users can combine their secret keys to recover the master secret s and then access all files on the system. Let's see how. Suppose users 1 and 2 collude. Because r_1 and r_2 are distinct primes there are integers a and b such that $a r_1 + b r_2 = 1$. Now, users 1 and 2 can compute s from the secret keys s_1 and s_2 as follows:

Your Answer		Score	Explanation
● $s = s_1^a \cdot s_2^b$ in $\mathbb{Z}_N$.	✓	1.00	$s = s_1^a \cdot s_2^b = s^{r_1 a} \cdot s^{r_2 b} = s^{r_1 a + r_2 b} = s$ in $\mathbb{Z}_N$.
Total		1.00 / 1.00	

Question 8

Let G be a finite cyclic group of order n and consider the following variant of ElGamal encryption in G:

- Gen: choose a random generator g in G and a random x in $\mathbb{Z}_n$.
  Output $\mathrm{pk} = (g, h = g^x)$ and $\mathrm{sk} = (g, x)$.
- $E(\mathrm{pk}, m \in G)$: choose a random r in $\mathbb{Z}_n$ and output $(g^r,\ m \cdot h^r)$.
- $D(\mathrm{sk}, (c_0, c_1))$: output c_1 / c_0^x.

This variant, called plain ElGamal, can be shown to be semantically secure under an appropriate assumption about G. It is however not chosen-ciphertext secure because it is easy to compute on ciphertexts. That is, let (c_0,c_1) be the output of $E(\mathrm{pk},m_0)$ and let (c_2,c_3) be the output of $E(\mathrm{pk},m_1)$. Then just given these two ciphertexts it is easy to construct the encryption of $m_0 \cdot m_1$as follows:

Your Answer		Score	Explanation
◉ $(c_0c_2,\ c_1c_3)$ is an encryption of of $m_0 \cdot m_1$.	✓	1.00	Indeed, $(c_0c_2,\ c_1c_3)=(g^{r_0+r_1},\ m_0m_1h^{r_0+r_1})$, which is a valid encryption of m_0m_1.
Total		1.00 / 1.00	

Question 9

Let G be a finite cyclic group of order n and let $\mathrm{pk}=(g,h=g^a)$ and $\mathrm{sk}=(g,a)$ be an ElGamal public/secret key pair in G as described in Segment 12.1. Suppose we want to distribute the secret key to two parties so that both parties are needed to decrypt. Moreover, during decryption the secret key is never re-constructed in a single location. A simple way to do so it to choose random numbers a_1,a_2 in $\mathbb{Z}_n$ such that $a_1+a_2=a$. One party is given a_1 and the other party is given a_2. Now, to decrypt an ElGamal ciphertext (u,c) we send u to both parties. What do the two parties return and how do we use these values to decrypt?

Your Answer		Score	Explanation
◉ party 1 returns $u_1 \leftarrow u^{a_1}$, party 2 returns $u_2 \leftarrow u^{a_2}$ and the results are combined by computing $v \leftarrow u_1 \cdot u_2$.	✓	1.00	Indeed, $v=u_1 \cdot u_2=g^{a_1+a_2}=g^a$ as needed for decryption. Note that the secret key was never re-constructed for this distributed decryption to work.
Total		1.00 / 1.00	

Question 10

Suppose Alice and Bob live in a country with 50 states. Alice is currently in state $a \in \{1,\ldots,50\}$ and Bob is currently in state $b \in \{1,\ldots,50\}$. They can communicate with one another and Alice wants to test if she is currently in the same state as Bob. If they are in the same state, Alice should learn that fact and otherwise she should learn nothing else about Bob's location. Bob should learn nothing about Alice's location.

They agree on the following scheme:

- They fix a group G of prime order p and generator g of G
- Alice chooses random x and y in $\mathbb{Z}_p$ and sends to Bob $(A_0,A_1,A_2)=(g^x,\ g^y,\ g^{xy+a})$
- Bob choose random r and s in $\mathbb{Z}_p$ and sends back to Alice $(B_1,B_2)=(A_1^r g^s,\ \ (A_2/g^b)^r A_0^s)$

What should Alice do now to test if they are in the same state (i.e. to test if $a=b$) ?

Note that Bob learns nothing from this protocol because he simply recieved a plain ElGamal encryption of g^a under the public key g^x. One can show that if $a \neq b$ then Alice learns nothing else from this protocol because she recieves the encryption of a random value.

Your Answer		Score	Explanation
◉ Alice tests if $a=b$ by checking if $B_2/B_1^x=1$.	✓	1.00	The pair (B_1,B_2) from Bob satisfies $B_1=g^{yr+s}$ and $B_2=(g^x)^{yr+s}g^{r(a-b)}$. Therefore, it is a plain ElGamal encryption of the plaintext $g^{r(a-b)}$ under the public key (g,g^x). This plaintext happens to be 1 when $a=b$. The term B_2/B_1^xcomputes the ElGamal plaintext and compares it to 1. Note that when $a \neq b$ the $r(a-b)$ term ensures that Alice learns nothing about b other than the fact that $a \neq b$. Indeed, when $a \neq b$then $r(a-b)$ is a uniform non-zero element of $\mathbb{Z}_p$.
Total		1.00 / 1.00	

Question 11

[OPTIONAL: EXTRA CREDIT] What is the bound on d for Wiener's attack when N is a product of**three** equal size distinct primes?

Your Answer		Score	Explanation
● $d<N^{1/6}/c$for some constant c.	✓	1.00	The only change to the analysis is that $N-\varphi(N)$ is now on the order of $N^{2/3}$. Everything else stays the same. Plugging in this bound gives the answer. Note that the bound is weaker in this case compared to when N is a product of two primes making the attack less effective.
Total		1.00 / 1.00	