Lab #2 - September 13, 2018

Due Dec 31 at 11:59pm **Points** 9 **Questions** 9
Available Sep 13 at 9:15am - Dec 31 at 11:59pm 4 months **Time Limit** None
Allowed Attempts Unlimited

Instructions

[1] Good morning and welcome to the second lab session!

Lab sessions provide the opportunity for recitation and more in-depth understanding of the materials covered in class, as well as preparation for upcoming homework assignments.

Your attendance only of a given lab session (and, thus, your participation in the assignments and/or discussions) gives you full credit. You are expected to stay in the lab for the entire session, or until the TAs release the class possibly earlier than scheduled, and to actively participate in the discussions (e.g., asking questions, answering to questions, etc.).

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 2	2 minutes	6 out of 9
LATEST	Attempt 2	2 minutes	6 out of 9
	Attempt 1	32 minutes	5 out of 9

Submitted Oct 4 at 1:59pm

Question 1

1 / 1 pts

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(	72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29	)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[	123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	|
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D	]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

By now we know that using OTP with the same key generally leaks the XOR of plaintext messages. This is a not-so-benign leakage when English text is transmitted - for example, when Alice and Bob chat, say, using only letters and spaces. To see why, take a look at the ASCII table above (or visit its source page [here (http://web.alfredstate.edu/faculty/weimandn/miscellaneous/ascii/ascii_index.html)](http://web.alfredstate.edu/faculty/weimandn/miscellaneous/ascii/ascii_index.html) .)

Prefixed by 000, column one contains 32 control (non-printable) characters. Prefixed by 001, column two contains 32 special and arithmetic characters, whereas columns three and four include all alphabetic characters, respectively capitals prefixed by 010, and lower cases prefixed by 011.

What happens when two alphabetic characters are XORed?

- ○ They can be XORed to any of the 128 characters in the table.

Correct!

- ◉ They are necessarily XORed to a character of columns one or two.

  Yes, the prefix of the resulting character is 01* XOR 01* = 00*, thus is contained in column one or two.

- ○ They are necessarily XORed to a control character.

The prefix of the resulting character is 01* XOR 01* = 00*, thus it will necessarily be contained in column one or two.

Question 2

1 / 1 pts

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(	72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29	)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[	123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	|
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D	]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

What happens when an alphabetic character is XORed with the space character SP?

Correct!

- (●) The result is the alphabetic character in "flipped" case.

Indeed, the prefix of the resulting character is 010 XOR 01b = 00b', where b' is the complement of b, thus the XORed character changes from column three to column four or vice versa. Importantly, the suffix of the character remains the same, thus the character does not change but only its case is flipped.

- ( ) The result can be any of the 128 characters in the table.

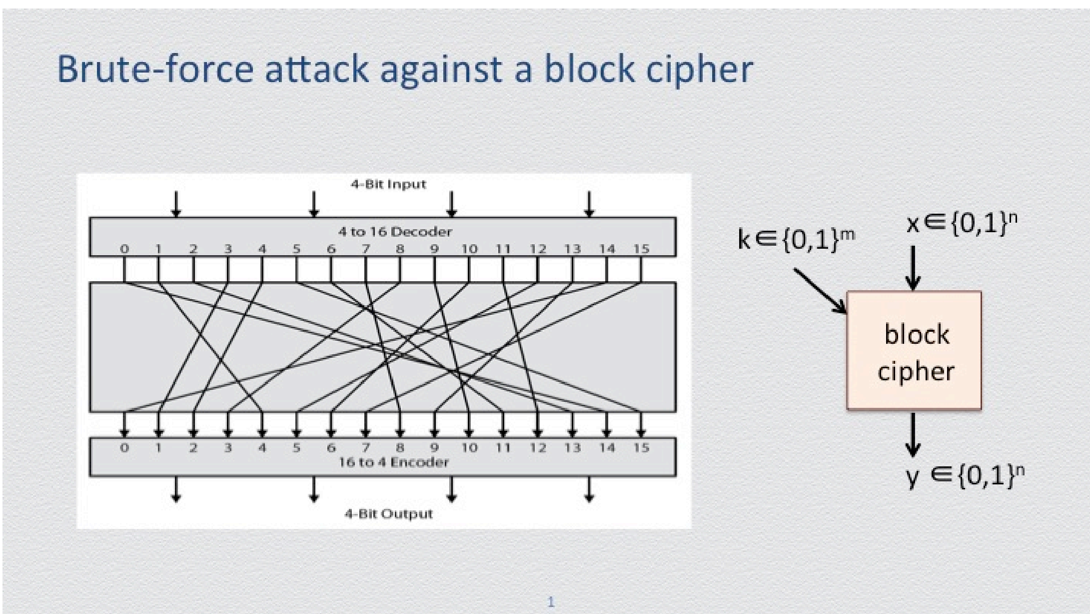
The result is a control character.

The result is the alphabetic character in "flipped" case. E.g., 't' will become 'T' or 'h' will become 'H'.

Indeed, the prefix of the resulting character is 010 XOR 01b = 00b', where b' is the complement of b, thus the XORed character changes from column three to column four or vice versa. Importantly, the suffix of the character remains the same, thus the character does not change but only its case is flipped.

Question 3

1 / 1 pts



A block cipher of block size n and key size m as above (e.g., DES or AES) is a cipher that approximates a random permutation mapping n-bit inputs (x) to n-bit outputs (y). The role of the secret key (k) is to specify the specific permutation that is implemented by the block cipher.

Assume that an attacker has holds a valid plaintext-ciphertext pair (M,C) of a block cipher (Enc, Dec) as the above. That is, $Enc_k(M) = C$. What constitutes a brute-force attack against this cipher?

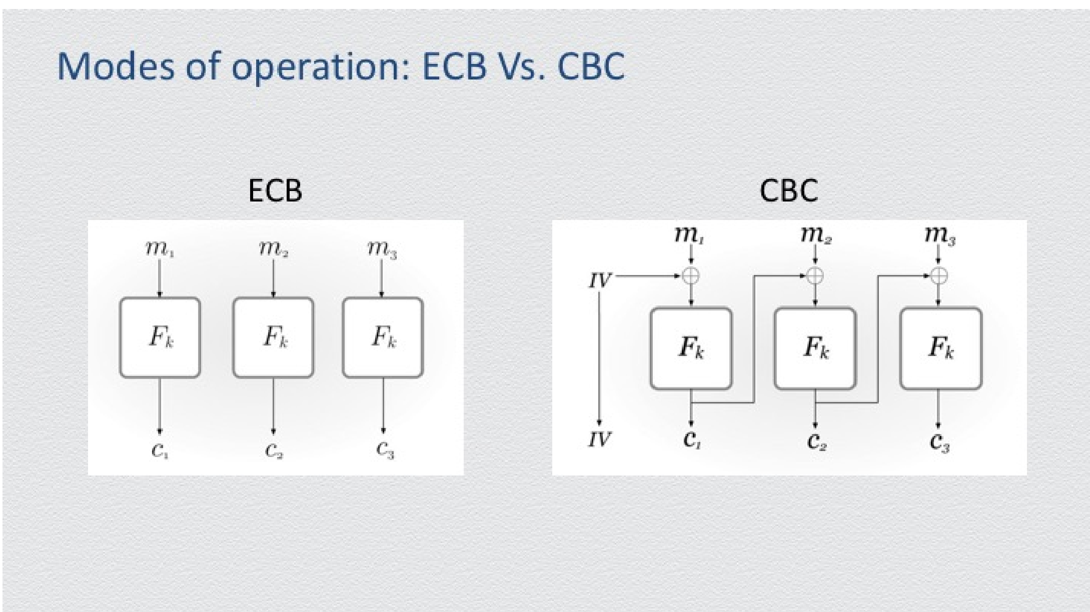
Correct!

- (selected) Going through all possible m-bit keys k and checking whether $Enc_k(M) = C$.
- The cipher is broken already, as the attacker can inspect the block cipher "box" and discover the key.
- Going through all $(2^n)!$ permutations and checking which ones map M to C.

A brute-force attack in this case constitutes going through all possible m-bit keys k and checking whether Enck(M) = C.

Question 4

1 / 1 pts



For any block cipher, the mode of its operation describes the way by which the block cipher encrypts or decrypts a sequence of message blocks (that is, messages that are longer than the block size, say, have size that is a multiple of the block size).

In terms of efficiency and tolerance against possible network-connectivity problems, which mode of operation between ECB and CBC is preferable to use when block ciphers are used for protecting the confidentiality of messages sent over an insecure channel?

Correct!

- (selected) The ECB mode, because it can run faster without being affected by possible dropped packets.

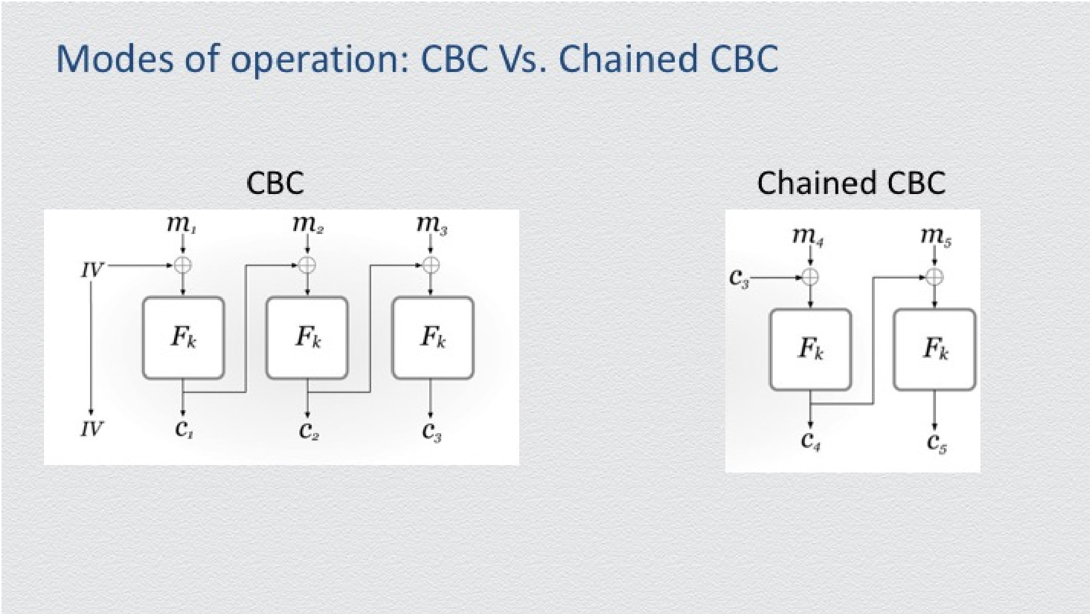
  Indeed, as blocks are processed independently of each other, encryption and decryption can be highly parallelized and network errors over a transmitted encrypted block do not disrupt any other blocks.

- The CBC mode, because the chaining pratically adds no extra overhead and also serves as an error-correcting checksum allowing to tolerate transmission errors.

- They are equally preferable.

Question 5

0 / 1 pts



Suppose that Alice wish to send message m to Bob, followed by another message m'.

In CBC mode, any such message is sent in the ciphertext form (IV, block ciphertext 1, block ciphertext 2, ...) for a initial vector that is selected uniformly at random . That is, messages m = (m_1, m_2, m_3) and m' = (m_1, m_2) are transmitted as as c = (IV, c_1, c_2, c_3) and c' = (IV', c_1, c_2), respectively (for some uniformly random initial vectors IV and IV').

In chained-CBC mode, any message that is subsequent of another (i.e., it is not the very first to be sent) is transmitted in the ciphertext form (block ciphertext 1, block ciphertext 2, ...) computed using as initial vector the ciphertext c^* of the last block that was transmitted. That is, messages m = (m_1, m_2, m_3) and m' = (m_4, m_5) are transmitted as as c = (IV, c_1, c_2, c_3) and c' = (c_4, c_5), respectively, where c_4 is computed over input c_3 XOR c_4.

What useful property does chained-CBC mode achieve?

- ○ Message compression, because the ciphertext is shorter than the plaintext.

'ou Answered

- ◉ Stronger security, because less initial vectors are leaked to the attacker, thus its search space is smaller.

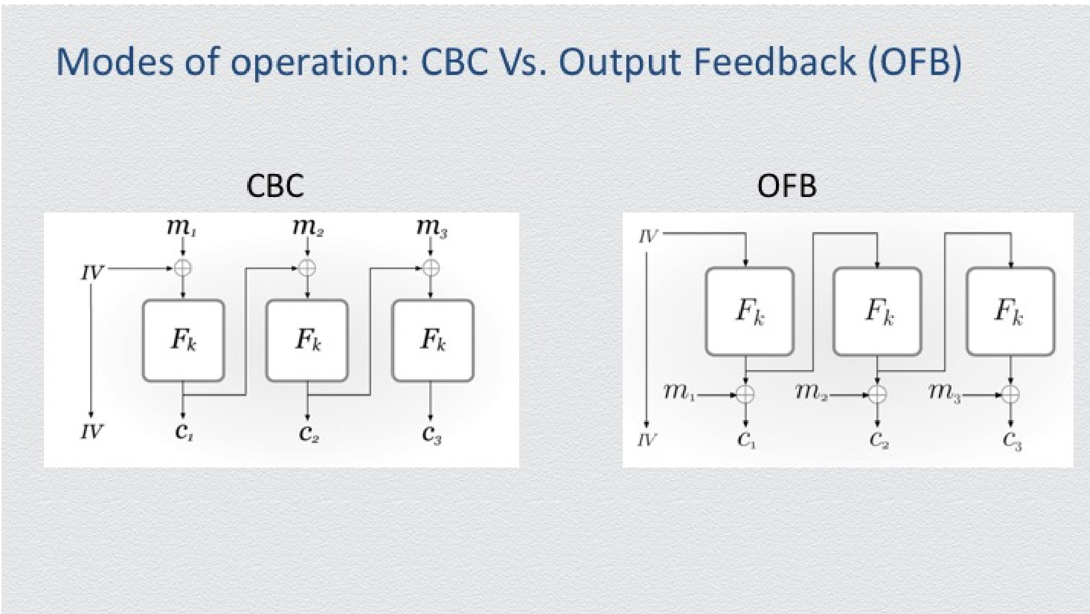
No. What is required for CBC to be secure is that IV is uniformly random, but not secret. Thus, the security of chained CBC is not strengthened (in fact, it is weakened!).

orrect Answer

Bandwidth efficiency, because fewer initial vectors are transmitted.

Question 6

0 / 1 pts



The OFB mode of operation resembles the CBC mode but now the chaining occurs at the output layer of the block-cipher transformations (IV is still a uniformly random string). Compared to CBC, what efficiency advantage does OFB provide?

Message encryption/decryption can be parallelized.

'ou Answered

Message encryption/decryption is faster if preprocessing is allowed.

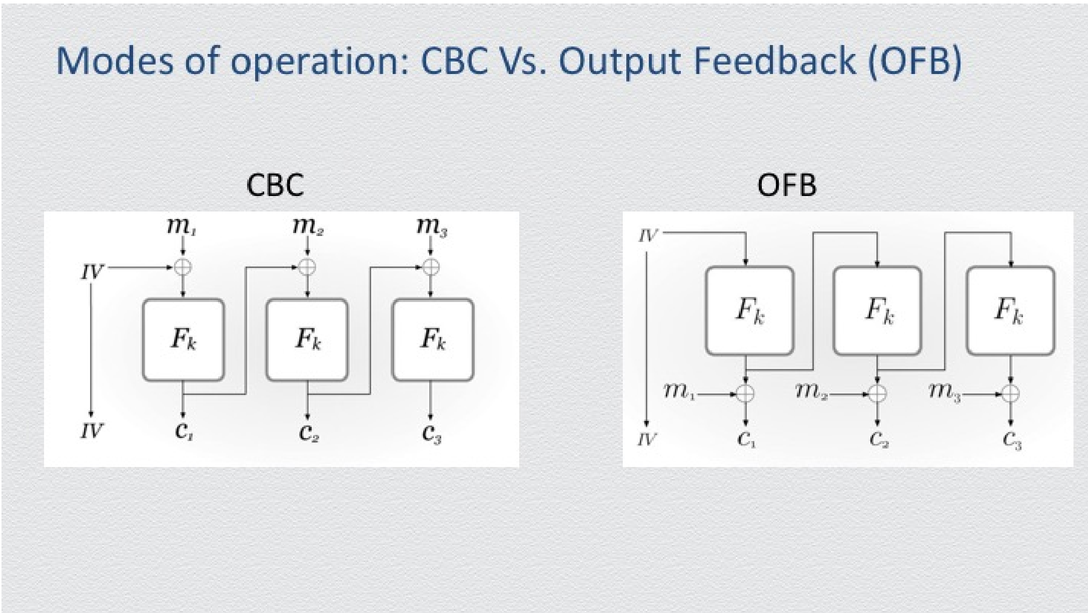
No, no information can be precomputed.

orrect Answer

The encrypted message requires no preprocessing.

Question 7

1 / 1 pts



Which of the two encryption modes is closer to the one-time pad cipher?

Both modes as they both employ the XOR function.

None of these modes as they both involve key reuse.

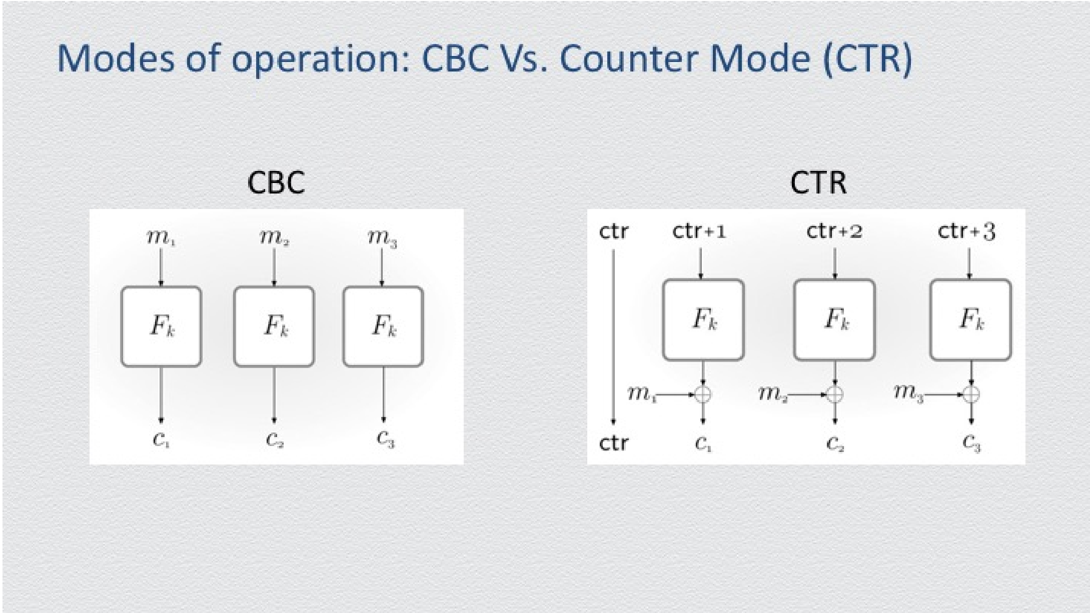
Correct!

The OFB mode as it implements XOR-type of message masking.

Indeed, this is essentially one-time pad encryption under an ephemeral long pseudorandom key that is computed by repeated applications of a block cipher on a fresh random initial block.

Question 8

0 / 1 pts



Finally, the CTR (counter) mode of operation resembles the OFB mode where no chaining is employed at the input layer of the block-cipher transformations. But it also resembles the ECB mode as encryption involves the independent invocation of a bock cipher (in the case of the CTR mode, while maintaining some state, a counter ctr).

How does CTR compare to ECB?

'ou Answered

- ECB is faster for encryption whereas CTR is faster for decryption.

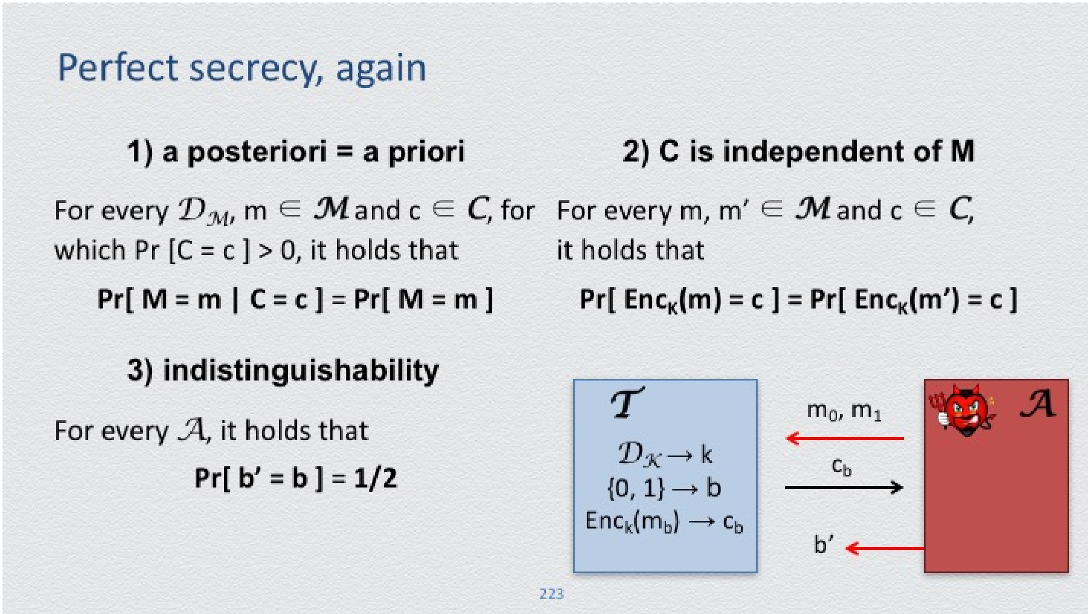
Not really.

orrect Answer

- CTR is more secure than ECB, as it uses a counter.
- ECB is more secure than CTR, as it does not leak and internal state.

Question 9

1 / 1 pts



It turns out that there is a third way to define perfect secrecy for symmetric-key encryption through "indistinguishability." This concept captures security through a game played between an attacker A and a trusted party T. The attacker provides T with two messages, then T randomly chooses a secret key k and a flip a coin to decide which of the two messages to encrypt; then, T provides A with the produced ciphertext and A wins the game, if A succeeds in distinguishing the two chosen messages by simply observing the provided ciphertext, i.e., if A finds which bit was flipped by T. With such an interpretation of this third new security definition, it can be shown that all the above three security definitions above are equivalent.

Intuitively, how does the new definition capture secrecy in a perfect manner?

Correct!

By requiring that the best winning strategy of A is to simply guess the bit that T chose and by providing A with the choice of selecting which messages to be challenged against.

Indeed, these two properties together capture perfect security, since A is completely unable to distinguish any two messages (even of its choice).

By providing A with the choice of selecting which messages to be challenged against.

By requiring that the best winning strategy of A is to simply guess the bit that T chose.