is applied to the file contents and nothing else. What tampering attacks	
are not prevented by this system?	
Changing the last modification time of a file. Replacing the contents of a file with the concatenation of two files	
on the file system. Changing the first byte of the file contents.	
Replacing the tag and contents of one file with the tag and contents of a file from another computer protected by the same MAC system, but a different key.	
Incorrect The MAC tag will fail to verify if any file data is changed.	
2. Let (S,V) be a secure MAC defined over (K,M,T) where $M=\{0,1\}^n$ and $T=\{0,1\}^{128}$. That is, the key space is K , message space is $\{0,1\}^n$, and tag space is $\{0,1\}^{128}$.	0 / 1 point
Which of the following is a secure MAC: (as usual, we use to denote string concatenation)	
$S'(k,m)=S(k,m)$ and $V'(k,m,t)= \left\{ egin{array}{ll} V(k,m,t) & ext{if } m eq 0^n \\ ``1" & ext{otherwise} \end{array} ight.$	
This should not be selected $ \text{This construction is insecure because the adversary can simply output} $ $ \left(0^n,0^s\right) \text{ as an existential forgery.} $	
$S'((k_1,k_2),m)=ig(S(k_1,m),S(k_2,m)ig)$ and	
$V'\left((k_1,k_2),m,(t_1,t_2)\right) = \left[V(k_1,m,t_1) \text{ and } V(k_2,m,t_2)\right]$ (i.e., $V'\left((k_1,k_2),m,(t_1,t_2)\right)$ outputs ``1" if both t_1 and t_2 are valid tags)	
\checkmark Correct a forger for (S',V') gives a forger for (S,V) .	
$S'(k,m) = S(k,m\oplus m)$ and	
$V'(k,m,t)=V(k,\ m\oplus m,\ t)$	
This should not be selected This construction is insecure because an adversary can	
request the tag for $m=0^n$ and thereby obtain a tag $ \label{eq:formula} $ for any message. This follows from the fact that	
$m\oplus m=0.$	
$V'(k,m,t) = V(k, \ m[0,\dots,n-2] \ 0, \ t)$	
This should not be selected This construction is insecure because the tags on	
$m=0^n$ and $m=0^{n-1}1$ are the same. Consequently, $\label{eq:model}$ the attacker can request the tag on $m=0^n$ and output	
an existential forgery for $m=0^{n-1}1.$	
$V'ig(k,m,(t_1,t_2)ig) = egin{cases} V(k,m,t_1) & ext{if } t_1 = t_2 \ ext{"0"} & ext{otherwise} \end{cases}$	
(i.e., $V'ig(k,m,(t_1,t_2)ig)$ only outputs "1" if t_1 and t_2 are equal and valid)	
✓ Correct	
a forger for (S',V') gives a forger for (S,V) .	
$V'(k,m,t) = V(k,\;m \ m,\;t).$	
\checkmark Correct a forger for (S',V') gives a forger for $(S,V).$	
 Recall that the ECBC-MAC uses a fixed IV (in the lecture we simply set the IV to 0). 	1/1 point
Suppose instead we chose a random IV for every message being signed and include the IV in the tag. In other words, $S(k,m):=\left(r,\ \mathrm{ECBC}_r(k,m)\right)$	
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.	
``0" otherwise. The resulting MAC system is insecure.	
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)	
The tag $(r\oplus m,\ r)$ is a valid tag for the 1-block message 0^n . The tag $(m\oplus t,\ r)$ is a valid tag for the 1-block message 0^n . The tag $(m\oplus t,\ r)$ is a valid tag for the 1-block message 0^n .	
The tag $(m \oplus t, \ r)$ is a valid tag for the 1-block message 0^n . The tag $(r, \ t \oplus r)$ is a valid tag for the 1-block message 0^n .	
\checkmark Correct 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 $(r\oplus m,\ t)$ is a valid existential forgery for the message 0 .	
4. Suppose Alice is broadcasting packets to 6 recipients	0 / 1 point
B_1,\dots,B_6 . Privacy is not important but integrity is.	
In other words, each of B_1,\dots,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,\dots,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_1 can use the key k to send packets with a valid tag to	
use the key κ to send packets with a valid tag to users B_2,\dots,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 $S_{f i}\subseteq S$ 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?	
This should not be selected User 5 can fool user 1 into believing that a packet	
from user 5 was sent by Alice.	
$S_1 = \{k_2, k_4\}, S_2 = \{k_2, k_3\}, S_3 = \{k_3, k_4\}, S_4 = \{k_1, k_3\}, S_5 = \{k_1, k_2\}, S_6 = \{k_1, k_4\}$	
Correct 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 .	
$S_1=\{k_1,k_2\},\ S_2=\{k_1\},\ S_3=\{k_1,k_4\},\ \S_4=\{k_2,k_3\},\ S_5=\{k_2,k_4\},\ S_6=\{k_3,k_4\}$ This should not be selected	
User 1 can fool user 2 into believing that a packet from user 1 was sent by Alice.	
This should not be selected User 3 can fool user 6 into believing that a packet	
from user 3 was sent by Alice.	
5. Consider the encrypted CBC MAC built from AES. Suppose we	1/1 point
compute the tag for a long message \boldsymbol{m} comprising of \boldsymbol{n} AES blocks.	
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	
Let m^\prime be the n -block message obtained from m by flipping the	
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)	
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)	
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)	
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)	
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) \bullet 4 \bullet 6 \bullet 5 \bullet n Correct 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 ,	
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) 4 6 5 n Correct 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.	
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 $b \oplus 1$). 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) Correct 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.	0/1 point
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) \bullet 4 \bullet 6 \bullet 5 \bullet n Correct 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.	0 / 1 point
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) a 4 b 6 b 5 n correct 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.	0/1 point
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) 4 6 5 n Correct 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. 6. Let $H: M \to T$ be a collision resistant hash function. Which of the following is collision resistant: (as usual, we use \parallel to denote string concatenation) $\parallel H'(m) = H(m)$	0/1 point
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) 4 6 5 n Correct 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. 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) $H'(m) = H(m)$ (i.e. hash the length of m)	0/1 point
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) 4 6 5 n Correct 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. 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) $\ H'(m) = H(m)$ (i.e. hash the length of m)	0/1 point
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) \bigcirc 4 \bigcirc 6 \bigcirc 5 \bigcirc n Correct 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. 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) \square $M'(m) = H(m)$ (i.e. hash the length of m) 1. This should not be selected This construction is not collision resistant because $H(000) = H(111)$. \square $M'(m) = H(m) \oplus H(m \oplus 1)^{[m]}$	8/1 point
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) a 4 b 6 b 5 b 7 b 7 b 6 b 7 b 7 b 8 b 8 b 8 b 8 b 8 b 9 b 9 b 9 b 9 b 9	0/1 point
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 $b \oplus 1$). 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) 4 6 5 5 70 10 10 10 10 10 10 10 10 10 10 10 10 10	0/1 point
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) 4 6 5 7 7 Correct 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. 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) $\ H'(m) = H(m)$ (i.e. hash the length of m) 1. This should not be selected This construction is not collision resistant because $H(000) = H(111)$. $\ H'(m) = H(m) \oplus H(m \oplus 1^{ m })$ (where $m \oplus 1^{ m }$ is the complement of m) 1. This should not be selected This construction is not collision resistant because $H(000) = H(111)$. $\ H'(m) = H(m) \oplus H(m)$ 1. This should not be selected This construction is not collision resistant because $H(000) = H(111)$.	0/1 point
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 $b \oplus 1$). 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) 4 6 5 5 7 n Correct 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. 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) $W H'(m) = H(m)$ (i.e. hash the length of m) 1. This should not be selected This construction is not collision resistant because $H(000) = H(111)$. $W H'(m) = H(m) \oplus H(m \oplus 1^{ m })$ (where $m \oplus 1^{ m }$ is the complement of m) 1. This should not be selected This construction is not collision resistant because $H(000) = H(111)$. $W H'(m) = H(m) \oplus H(m \oplus 1^{ m })$ (where $M \oplus H(m) \oplus H(m)$) 1. This should not be selected This construction is not collision resistant because $H(000) = H(111)$.	0/1 point.
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) 4 6 5 n Correct 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. 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) $\ H'(m) = H(m)$ (i.e. hash the length of m) 1. This should not be selected This construction is not collision resistant because $H(000) = H(111)$. $\ H'(m) = H(m) \oplus H(m \oplus 1^{[m]})$ (where $m \oplus 1^{[m]}$ is the complement of m) 1. This should not be selected This construction is not collision resistant because $H(000) = H(111)$. $\ H'(m) = H(m) \oplus H(m)$ 1. This should not be selected This construction is not collision resistant because $H(000) = H(111)$. $\ H'(m) = H(m) \oplus H(m)$	0/1 point
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 $b \oplus 1$). 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) 4	0/1 point
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 ALS 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 ALS block size). ② 4	O/1 polint
Let m' be the n -block message obtained from m by flipping 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) 4 6 6 5 7 6 7 7 6 8 6 7 9 7 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	0/1 point
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 ALS 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 ALS block size) 4	9/1 point
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 site) 4	9/1 point
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 $b \oplus 1$). 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 site.) 4	8/1 polex
Let m' be the h-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 ⊕ 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) ② 4 ③ 6 ⑤ 5 ⑤ 7 ✓ Correct You would decrypt the final CBC MAC encryption step done using k₂, the decrypt the last CBC MAC encryption step done using k₂, the decrypt the last CBC MAC encryption step done using k₂, the decrypt the last cBC MAC encryption step done using k₂, the decrypt the last cBC mAC encryption step done using k₂, the decrypt the last cBC mAC encryption step done using k₂, the decrypt the last cBC mAC encryption step done using k₂, the decrypt the last cBC mAC encryption step done using k₂, the decrypt the last cBC mAC encryption step done using k₂, the decrypt the last cBC mAC encryption step done using k₂, the decryption is the confliction resistant in the last mile last mile step is defined as the step is done using k₂, the decryption is decreased that function is not collision resistant because H(000) = H(111). ② H'(m) = H(m) ⊕ H(m) ⊕ H(m) I This should not be selected This construction is not collision resistant because H(0) = H(1). ② H'(m) = H(m) H(m) ✓ Correct a collision finder for H' gives a collision finder for H.	2/1 point.
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 $b \oplus 1$). 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 site) 4 6 6 7 Correct You would decrypt the final CBC MAC encryption step done using k_0 , the decryption last CBC MAC encryption step done using k_0 , flip the last bit of the result, and re-apply the two encryptions. 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) $\ H'(m) = H(m)$ (i.e. hash the length of m) 1. This should not be selected This construction is not collision resistant because $H(000) = H(111)$. $\ H'(m) = H(m) \oplus H(m \oplus 1)^{m(1)}$ (where $m \oplus 1$ $\ m $ is the complement of m) 1. This should not be selected This construction is not collision resistant because $H(000) = H(111)$. $\ H'(m) = H(m) \oplus H(m)$ 1. This should not be selected This construction is not collision resistant because $H(000) = H(111)$. $\ H'(m) = H(m) \oplus H(m)$ 2. This should not be selected This construction is not collision resistant because $H(0) = H(1)$. $\ H'(m) = H(m) \oplus H(m)$ 2. This should not be selected This construction is not collision resistant because $H(0) = H(1)$. $\ H'(m) = H(m) \oplus H(m)$ 2. This should not be selected This construction is not collision resistant because $H(0) = H(1)$. $\ H'(m) = H(m) \oplus H(m)$ 2. This should not be selected This construction is not collision resistant because $H(0) = H(1)$.	171 point
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 5 then the last bit of m' is 5 ⊕ 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 site) ✓ Correct ✓ Correct You would decrypt the final CBC MAC encryption step done using k ₁ . flip the last bit of the result, and re-apply the two encryptions. 6. Let H: M → T be a collision resistant hash function. Which of the following is collision resistant: (as usual, we use to denote string concatenation) ☑ H'(m) = H(m) ☑ e. neath the length of m) ! This should not be selected This construction is not collision resistant because H(000) = H(111). ☑ H'(m) = H(m) ⊕ H(m) ! This should not be selected This construction is not collision resistant because H(000) = H(111). ☑ H'(m) = H(m) ⊕ H(m) ! This should not be selected This construction is not collision resistant because H(0) = H(1). ☑ H'(m) = H(m) ⊕ H(m) ! This should not be selected This construction is not collision resistant because H(0) = H(1). ☑ H'(m) = H(m) ⊕ H(m) ! This should not be selected This construction is not collision resistant because H(0) = H(1). ☑ H'(m) = H(m)m) ✓ Correct a collision finder for H' gives a collision finder for H. ☑ H'(m) = H(m m)	
Let m' be the m -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 (i.e. if the last bit of m is b then the last bit of m' is b the 1, how many m' after the last bit of m' is b the 1, how many m' after the last bit of m' is b the 1, how many m' after the last bit of m' and the MAC keyl (in this question please ignore message padding and simply assume that the message length is always a multiple of the AES block stee) © 4 © 5 S On Cerrect You would decrypt the final CBC MAC encryption step done using k_1 , the decryption is last CBC MAC encryption step done using k_2 , the decryption is called the contract of the following is collision resistant hash function. Which of the following is collision resistant: (as usual, we use $\ \mathbf{t} \cdot \mathbf{d} \cdot \mathbf{m} \cdot $	
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 0 the last bit of m is 0 the last bit of m is 0 the last bit of m' is 0 the last bit of 0 the result, and re-apply the two encryptions. 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) $\ H'(m) = H(m)\ $ (i.e. hash the length of 0) where m is 1^{100} is the complement of m) $\ H'(m) = H(m)\ \oplus H(m) \oplus H(m)\ $ where m is 1^{100} is the complement of m) $\ H'(m) = H(m)\ \oplus H(m)\ \oplus H(m)\ $ $\ H'(m) = H(m)\ \oplus H($	
Let m' be the n-block message obtained from m by flipping the last bit of m' is $b \otimes 1$). Now many calls to AS would it take to compute the Lag for m' from the tag for m and the MAC key? (is this question please ignare message problem, and simply assume that the message length is always a multiple of the AS block size) ① 4 ○ 6 ○ 5 ○ 7 ○ m ✓ Correct Volume decrypt the final CBC MAC encryption step done using k_1 , the decrypt the last CBC MAC encryption step done using k_1 , the decrypt the last CBC MAC encryption step done using k_1 . The problem is a decident of the result, and re-apply the two encryptions. 6. Let $H: M \to T$ be a collision resistant thath function. Which of the following is collision resistant: (as usual, we use $\ \mathbf{x} = x$	
Let m' be the n-block message obtained from m by flipping the last bit of m' is b of 1). Now many calls to AS 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 AS block state). © 4	
Let m' be the m-block message obtained from m by flipping the last bit of m' is b a 1). Now many calls to b 5 would it take to compute the tag for m' from the tag for m and the MAC key? (in this question please ignore may be also also also also also also also also	
Let m' be the n-block message obtained from m by flipping the last bit of m' is b of 1). Now many calls to b S would it take to compute the page of m' from the size to b should it take to compute the page of m' from the size for m and the MAC key? (in this quisation please ignore message padding and simply assume that the message length is always a multiple of the A S block size) © 4 © 6 S S n Verver You would decrypt the final CIC MAC encryption step done using k , the decrypt the last CBC MAC encryption step done using k , the decrypt the last CBC MAC encryption step done using k , the decrypt the last CBC MAC encryption step done using k , the decrypt the last CBC MAC encryption step done using k , the decrypt the last CBC MAC encryption step done using k , the decrypt the last CBC MAC encryption step done using k , the decrypt the last CBC MAC encryption step done using k , the decrypt the last CBC MAC encryption step done using k , the decrypt the last CBC MAC encryption step done using k , the decrypt the last CBC MAC encryption step done using k , the decrypt the last CBC MAC encryption step done using k , the decrypt the last CBC MAC encryption step done using k , the decryption is the step of the step of the following its collision resistant: (as usual, we use $\ \mathbf{u} \ = \mathbf{u} \ =$	
Let m' be the ni-block message obtained from m by flipping the last bit of m is b if the last bit of m is the b 1. How many calls to b 45 would it take to compute 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 ASS block size) © 4 © 6 S O 7 O 8 N	
Let m' be the n-block message obtained from m by flipping the last bit of m' is $b \in \mathbb{N}$. If the less bit of m' is $b \in \mathbb{N}$. If the less bit of m' is $b \in \mathbb{N}$. If the less bit of m' is $b \in \mathbb{N}$. If the less bit of m' is $b \in \mathbb{N}$. If the less bit of m' is $b \in \mathbb{N}$. If the less bit of m' is $b \in \mathbb{N}$. If the less bit of m' is $b \in \mathbb{N}$. If the less bit of $b \in \mathbb{N}$ is a complete by a grant from the less bit of $b \in \mathbb{N}$. If the less bit of $b \in \mathbb{N}$ is a complete by a grant of the less bit of $b \in \mathbb{N}$ is a complete by a grant of the less bit of $b \in \mathbb{N}$ is a complete by a grant of the less bit of $b \in \mathbb{N}$ is a complete by a grant of the less bit of the	
Let in the the in-block message obtained from mit by flupping the last bit of mit (a.e. if the last bit of mit is then the last bit of mit (b. 6). It is now many calls to ASS would it take to compute the last for mit in the off mi	
Let In the the In-block message obtained from the by hipping the last bit of mile. If the last bit of mile is then the last bit of mile is the block message and mile and call to ASS would it take to compute the better for mile mile the size of mile mile the size of mile mile mile and mile mile mile mile mile mile mile mile	
Let In' be the in-block message obtained from m' by flopping the last bit of m' Let. If the last bit of m' is it then the last size of m' is ib is 1) from many calls to ASS would it take to compact the bas for m' in the CAS would it take to compact the bas for m' in the CAS would it take to compact the bas for m' from the bas 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 ASS block stole) © 4 © 6 © 5 In Vicenest You would decopt the final CBC MAC encyption step done using k_1 , the decopt the final CBC MAC encyption step done using k_2 , the decopt the final CBC MAC encyption step done using k_3 . The decopt the final CBC MAC encyption step done using k_4 . The decopt the final CBC MAC encyption step done using k_4 . The decopt the final CBC MAC encyption step done using k_4 . The decopt the final CBC MAC encyption step done using k_4 . The decopt the final CBC MAC encyption step done using k_4 . The decopt the final CBC MAC encyption step done using k_4 . The decopt the final CBC MAC encyption step done using k_4 . The decopt the final CBC MAC encyption is set of the final CBC MAC encyption and the set of the final CBC MAC encyption and the set of the final CBC MAC encyption and the set of the final CBC MAC encyption and the set of the final CBC MAC encyption and the set of the final CBC MAC encyption and the set of the final CBC MAC encyption and the set of the final CBC MAC encyption and the set of the final CBC MAC encyption and the set of the final CBC MAC encyption and CBC MAC encypti	
Let bit of the the In-black message obtained from m by Ripping the last bit of m (a.e. if the last bit of m is then the last bit of m) in the 5-1 Neor many calls to ASS would it take to compute the tag for m from the part of m and the MAC key? (in this question please ignore message pending and simply assume that the message length is always a multiple of the ASS black (185). © 4 • 6 • 6 • 7 • 7 • Curvet View would decryst the final CBC MAC encryption step done using k_1 , the decrypt the last CBC MAC encryption step done using k_2 , the decrypt the last CBC MAC encryption step done using k_1 , flip the last to of the result, and re-apply the two encryptions. 6. Let $H: M \to T$ be a callision resistant: [As state we use $[H: M] \to H(m)$] (a.e. have the largest of m) 1. This should not be attacked The construction in an accordance string concatenation) 2. If this should not be attacked The construction in an accordance resistant because $H(m) \to H(m)$ (100). 2. If this should not be attacked The construction in an accordance resistant because $H(m) \to H(m)$ (100). 3. If this should not be attacked This construction in an accordance resistant because $H(m) \to H(m)$. 3. $H'(m) \to H(m) \oplus H(m)$ 4. $H'(m) \to H(m) \oplus H(m)$ 5. $H'(m) \to H(m) \oplus H(m)$ 7. Curvet: a collision finder for H' gives a collision finder for H . 3. $H'(m) \to H(m) \oplus H(m)$ 4. $H'(m) \to H(m) \oplus H(m)$ 5. $H'(m) \to H(m) \oplus H(m)$ 6. $H'(m) \to H(m) \oplus H(m)$ 6. $H'(m) \to H(m) \oplus H(m)$ 6. $H'(m) \to H(m) \oplus H(m)$ 7. Suppose H_1 and H_2 are collision resistant. The construction is an accordance of the state	
Let in the the in-block message obtained from m by flipping the last bit of m (in b 0.1). Now many calls to AES would it take to compute the tag for in from the tag for min due to 10. Now many calls to AES would it take to compute the tag for in from the tag for min due to the compute the tag for min of the tag for min due to the compute the tag for min of the min of	
Let In if be the Tablock message detained from it by flipping the lasts but of in it. if it is but set but on it is been the last be of inf in it. if it is but the last of in it is but on the last be of inf in it is it. if the last per inf intention to compace the rap for inf intent the last for it and the MAC kay? (it is this question please ignore many participants of the MAC kay? (it is this question please ignore many participants of the MAC kay? (it is this question please ignore many participants of the MAC kay? (it is this question please ignore many participants of the MAC kay? (it is this question please ignore many participants of the MAC kay? (it is this question please ignore many participants). **Cornect** **Cornect** **Vocational description finisis Calculated encryption stage down using kip. **Bit on a start of the result, and finishing. **Bit on Hamilton in the manufacture of the start of the calculation resistant. **Cornect** **Cornect** **In Mac doubt matter bestitions* **The cornection is not calculation resistant the best of the manufacture of the man	171 paint
Let If be the in-block message declared from in by flipping the last bit of mit do in the MAC bity? (in this question planes ignore message perioding and simply assume that the message length is always a minipide of the AES block. Bit is a second decays the final CSC MAC encryption step does using \$b_i\$, the encryption is final CSC MAC encryption step does using \$b_i\$, the encryption is final called the mit do in the mit do in the encryptions. 6. Let \$B : M — T be a collision relations that function. Which is the final federage is collision relations: (as sexual, we use [1 to demote string concatenation) [2] If the abouted mate is window. [3] This abouted mate is window. [4] This abouted mate is window. [5] This abouted mate is window. [6] This abouted mate is window. [7] This abouted mate is window. [8] This abouted mate is window. [9] This abouted mate is window. [9] This abouted mate is window. [9] This abouted mate is window. [10] This abouted mate is window. [11] This abouted mate is window. [12] This abouted mate is window. [13] This abouted mate is window. [14] This abouted mate is window. [15] This abouted mate is window. [16] This abouted mate is window. [17] This abouted mate is window. [18] This window from the abouted. [18] This abouted mate is window. [18] This window from the abouted. [18]	171 paint
Let In if he the Tablech message detailed from it by flighting the least bit of it (ii.e.) If the lets that the of in a bit here he test bit of it is the let of it is of it is by a bit of the let by the original of the let by the original of the let by the let by a bit of it is of	171 paint
Let n'' be the n -black message statistical from n by Migsping the lasts better of n (i.e. if the statistic or n is the state better of n' is the n). Normally calls to ASI was many calls to ASI was many calls to ASI was many calls to ASI was statistic to compute the targ for n' from the tag for n' from the tag for n' is and the MAC key? (in this question please ignore was provided and variety account from the state of the ASI force n' of n' o	171 paint
Let m' be the n-block message obtained from m by flipping the lets that the off of the J the close that of m' is the J beam and the set that it is not offer the J the set that is not compare that set J from the tag for m' fro	171 paint
Let M' be the n-block message standard from m by flipping the last bit of m (i.e. if the bit side of m is b has the stand in the last bit of m' is b in b . In some greits a AS was married that the termination of a in b is b . In some greit in a AS was married that the termination of a in b is a in a in a in b in a in	171 paint
Let of the other habition measures obtained from m by flipping the last better of the fact but has the fail on the base that of an in a blace in the state but of the fail face of the fail fail fail fail fail fail fail fail	171 paint
Let of the other in Black in section of in the Shown the size that the state that of the Carlo Shown are producted to the section of the Shown Shown are producted to the section of the Shown Shown are producted to the section of the Shown Shown are producted to the section of the Shown Shown are producted to the Shown and the Shown are section of the Shown Shown and the Shown are section of the Shown Shown and the Shown are section of the Shown and the Shown and the Shown are section of the Shown and the	171 paint
Lest like of the In-Mark American shaker and the first in the pilling the last like of this last like of the last like of th	171 paint
Lest 10° but the 10-bit of, massage shadared from the lyt Rigaring the last 18 of the 18 bit of 18 bit 18 bit on the 18 bit 18 bit of 18 bit 18	171 pate:
Lest the the in-block measures obstanced from in by Rigging the last last of in the 1.8 are the last that of in the 1.8 are the last that of in the 1.8 are the last last of the 1.8 are the last last of the 1.8 are the 1.8 are the 1.8 are the 1.8 are the last last that the received the last place of from the last that the measures protein sensor proteins are suggested and simply sessions that the measures though the sharpy a municiple of the 4.85 block last 1.8 are the	171 pate:
Less that we make in Judicial missages abstanced from mis by Regarding the based let of mile. If the last let of mile has the last that was a first in a bit in the mine of the last that was to compare the last from of from the last for last that was to compare the last from if from the last for last the minestage length in always a multiple of the AES block steel of the last steel of the last form of from the last form and the MAC Macay in multiple of the AES block steel of the last form of from the	171 pate:
Let on be the in Allech message absolited from mits pillipping the last that of in J_{i} is the last that of in J_{i} is the last that of the order between the stage for it is an in the stage with a self-weet for the last between the stage from an of the last between places injuried to the stage of	171 pate:
Lest that the in the collection measures decidented from mits prilipolity the last that of the object and the set of the collection measures and the set of the collection of	171 pate:
Let on be the valued measure absoluted from my to flipping the bases that of my file, a fine but that of a first file. If the was the work in the file of the table of the file of the fi	171 pate:

In lecture we showed

 $\bigcirc O(|T|^{2/3})$

 $\bigcirc O(|T|^{1/2})$

 $\bigcirc O(|T|^{1/3})$

✓ Correct

follows.

 $\bigcirc O(|T|)$

that finding a collision on H can be done with $O\big(|T|^{1/2}\big)$

in M such that $H(x)=H(y)=H(z)\mbox{?}$

random samples of H. How many random samples would it take

until we obtain a three way collision, namely distinct strings $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$

An informal argument for this is as follows: suppose we

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)

with probability $1/\lvert T \rvert$ (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 \boldsymbol{n}

and $H(x)=H(z).\,$ Since each one of these two events happens

collect \boldsymbol{n} random samples. The number of triples among the \boldsymbol{n}

Week 3 - Problem Set Graded Quiz • 20 min

! Try again once you are ready

Week 3 - Problem Set

TO PASS 80% or higher

LATEST SUBMISSION GRADE

50%

Due Nov 18, 1:29 PM IST

GRADE 50%