W3 Problem Set && Programming Assignment

Q1

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?

   - (selected) Changing the name of a file.
   - Changing the first byte of the file contents.
   - Appending data to a file.
   - Replacing the contents of a file with the concatenation of two files on the file system.

问：假设一MAC (S,V)用于保护文件系统，方式为将tag附在每个文件后，签名算法S作用域文件内容，以下哪种方式的攻击不会遭到该系统的保护？

分析：由于S只作用于文件内容，显然更改文件名会导致攻击

Q2

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 $\{0,1\}^{128}$.

Which of the following is a secure MAC: (as usual, we use $\|$ to denote string concatenation)

- [ ] $S'(k, m) = S(k,\ m[0,\ldots,n-2]\|0)$ and

  $V'(k, m, t) = V(k,\ m[0,\ldots,n-2]\|0,\ t)$

- [x] $S'(k,\ m) = [t \leftarrow S(k, m),\ \text{output } (t, t)\,]$ and

  $$V'(k, m, (t_1, t_2)) = \begin{cases} V(k, m, t_1) & \text{if } t_1 = t_2 \\ \text{"0"} & \text{otherwise} \end{cases}$$

  (i.e., $V'(k, m, (t_1, t_2))$ only outputs "1"

  if t_1 and t_2 are equal and valid)

> ✓ 正确
> a forger for (S', V') gives a forger for (S, V).

- [x] $S'(k, m) = S(k,\ m\|m)$ and

  $V'(k, m, t) = V(k,\ m\|m,\ t)$.

> ✓ 正确
> a forger for (S', V') gives a forger for (S, V).

- [ ] $S'(k, m) = S(k, m \oplus m)$ and

  $V'(k, m, t) = V(k,\ m \oplus m,\ t)$

- [x] $S'(k, m) = S(k, m \oplus 1^n)$ and

  $V'(k, m, t) = V(k, m \oplus 1^n, t)$.

> ✓ 正确
> a forger for (S', V') gives a forger for (S, V).

- [ ] $S'(k, m) = (S(k, m), S(k, 0^n))$ and

  $V'(k, m, (t_1, t_2)) = [V(k, m, t_1) \text{ and } V(k, 0^n, t_2)]$

  (i.e., $V'(k, m, (t_1, t_2))$ outputs ``1" if both t_1 and t_2 are valid tags)

问：经典看不懂题目

分析：看懂了再分析，所以答案是抄的

Q3

Q4

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_1 can 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?

- [x] $S_1 = \{k_2, k_3\},\ S_2 = \{k_2, k_4\},\ S_3 = \{k_3, k_4\},\ S_4 = \{k_1, k_2\},\ S_5 = \{k_1, k_3\},\ S_6 = \{k_1, k_4\}$

✓ 正确

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.

- [ ] $S_1 = \{k_1, k_2\},\ S_2 = \{k_1, k_3\},\ S_3 = \{k_1, k_4\},\ S_4 = \{k_2, k_3, k_4\},\ S_5 = \{k_2, k_3\},\ S_6 = \{k_3, k_4\}$
- [ ] $S_1 = \{k_1\},\ S_2 = \{k_2, k_3\},\ S_3 = \{k_3, k_4\},\ S_4 = \{k_1, k_3\},\ S_5 = \{k_1, k_2\},\ S_6 = \{k_1, k_4\}$
- [ ] $S_1 = \{k_1, k_2\},\ S_2 = \{k_1, k_3, k_4\},\ S_3 = \{k_1, k_4\},\ S_4 = \{k_2, k_3\},\ S_5 = \{k_2, k_3, k_4\},\ S_6 = \{k_3, k_4\}$

问：Alice需要向6为客户B_1~B_6广播报文，需要确保完整性但无需确保安全性（即B1~B6应当确保收到的报问确实是Alice发送的）

假设Alice使用MAC，并与B_1~B_6共享密钥k，对于Bi收到的报文，若验证tag错误则丢弃报文

Alice注意到上述模型中存在缺陷，B_1可以利用共享密钥k，将报问发送给B_2~B_6而tag验证不会出错，因此B_2~B_6会认为报文流来自于Alice

假设新方案Alice使用一密钥集合S={k_1,......k_4}，对于Bi而言，分发给其的密钥为S的子集S_i，即S_i⊆S

问下述哪种密钥分配方案能确保没有任何一个客户能欺骗其他客户

分析：第一个选项中，任意两个客户B_i，B_j之间持有的密钥的交集小于等于一个密钥，由于通过验证需要两个密钥，因此任意一个用户不能产生其他用户的更多的密钥

对于选项二，B4拥有k_2，k_3，k_4，可以欺骗用户B_5和B_6

对于选项三，同理B_4，B_5，B_6可以欺骗B_1

对于选项四，B_2可以欺骗B_3，B_6，且B_5可以欺骗B_4，B_6

Q5

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)

- (●) 4
- ( ) 2
- ( ) 3
- ( ) $n+1$

✓ **正确**

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.

问：若CBC-MAC使用AES，假设计算一长消息m的tag，该消息包含n个AES块，记m'为另一长度为n块的消息，其为消息m的最后一位取反得到，则由m的tag计算得到m'的tag需要调用多少次AES算法?

分析：基于AES的CBC-MAC使用的是PRF，因此解得最后一块消息需要调用两次，之后最后一位取反再调用两次AES，共四次

Q6

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)

- [ ] $H'(m) = H(m) \bigoplus H(m \oplus 1^{|m|})$

  (where $m \oplus 1^{|m|}$ is the complement of m)

- [ ] $H'(m) = H(m[0, \dots, |m| - 2])$

  (i.e. hash m without its last bit)

- [x] $H'(m) = H(m) \| H(m)$

  ✓ **正确**
  a collision finder for H' gives a collision finder for H.

- [ ] $H'(m) = H(m)[0, \dots, 31]$

  (i.e. output the first 32 bits of the hash)

- [ ] $H'(m) = H(0)$

- [x] $H'(m) = H(m \| 0)$

  ✓ **正确**
  a collision finder for H' gives a collision finder for H.

- [x] $H'(m) = H(m \| m)$

  ✓ **正确**
  a collision finder for H' gives a collision finder for H.

问：若H : M → T为一抗碰撞hash函数，则下列哪些函数仍为抗碰撞？

分析：

1. 若m取000，则H'(000)=H(000)⊕H(111)，若m取111，则H'(111)=H(111)⊕H(000)，即H'(000)=H'(111)，碰撞
2. 截断消息最后一位不抗碰撞，有H'(00)=H'(01)
3. 显然H抗碰撞则H'也是
4. 同2
5. 显然不抗碰撞，因为有H'(0)=H'(1)
6. 显然H抗碰撞则H'也是
7. 显然H抗碰撞则H'也是

Q7

7. Suppose H_1 and H_2 are collision resistant

hash functions mapping inputs in a set M to $\{0,1\}^{256}$.

Our goal is to show that the function $H_2(H_1(m))$ 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:

- ( ) Either x, y are a collision for H_2 or

  $H_1(x), H_1(y)$ are a collision for H_1.

- ( ) Either x, y are a collision for H_1 or

  x, y are a collision for H_2.

- (•) Either x, y are a collision for H_1 or

  $H_1(x), H_1(y)$ are a collision for H_2.

- ( ) Either $H_2(x), H_2(y)$ are a collision for H_1 or

  x, y are a collision for H_2.

Q8

Q9

9. Repeat the previous question, but now to find a collision for the compression function $f_2(x,y) = \mathrm{AES}(x,x) \bigoplus y$.

Which of the following methods finds the required (x_1, y_1) and (x_2, y_2)?

- Choose x_1, x_2, y_1 arbitrarily (with $x_1 \neq x_2$) and set

  $y_2 = AES(x_1, x_1) \oplus AES(x_2, x_2)$
- Choose x_1, x_2, y_1 arbitrarily (with $x_1 \neq x_2$) and set

  $y_2 = y_1 \oplus x_1 \oplus AES(x_2, x_2)$
- Choose x_1, x_2, y_1 arbitrarily (with $x_1 \neq x_2$) and set

  $y_2 = y_1 \oplus AES(x_1, x_1) \oplus AES(x_2, x_2)$
- Choose x_1, x_2, y_1 arbitrarily (with $x_1 \neq x_2$) and set

  $y_2 = y_1 \oplus AES(x_1, x_1)$

✓ 正确

Awesome!

问：没看懂

答：蒙对的

Q10

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 $H(x) = H(y) = H(z)$?

- $O(|T|^{2/3})$
- $O(|T|^{1/2})$
- $O(|T|)$
- $O(|T|^{1/3})$

问：记H : M → T为一随机hash函数，｜M｜>>｜T｜，找到H的碰撞的期望为O(｜T｜$^{1/2}$)，若希望找到三个碰撞，即找到不同的x，y，z，使得H(x)=H(y)=H(z)，期望为多少

分析：首先对于给定的集合，包含n个元素，n个任意选择3个为C n-3，即期望为O(n^3)，对于每组特定的元素，需要求H(x)=H(y)=H(z)

而随机hash函数，产生碰撞的概率为1/｜T｜，则产生上述三路碰撞的概率为1/｜T｜^{2}（需要满足H(x)=H(y)且H(x)=H(z)）

因此期望为O(n^3/｜T｜^{2})