Bài tập 3[1]

Câu hỏi 1

Giả sử một hệ MAC (S, V) được dùng để bảo vệ các file trong hệ thống bằng cách thêm một MAC tag vào mỗi file. Thuật toán ký MAC S được áp dụng trên nội dung của file và không có gì khác. Kiểu tấn công giả mạo nào dưới đây mà hệ thống này không thể chống được?

1. Hoán đổi hai file trong hệ thống file.
2. Thay thế tag và nội dung của một file bằng tag và nội dung của file đặt trên máy tính khác cũng được bảo vệ bởi cùng hệ MAC, nhưng với khóa khác.
3. Thay thế nội dung của một file với việc ghép hai file trên hệ thống.
4. Xóa byte cuối của nội dung file.

Câu hỏi 2

Xét một hệ MAC an toàn (S, V) trên (K, M, T) với $M = \{0,1\}^n$ và $T = \{0,1\}^{128}$ (cụ thể, không gian khóa là K, không gian thông điệp là $\{0,1\}^n$, và không gian tag là $\{0,1\}^{128}$). MAC nào dưới đây là MAC an toàn: (ở đây, ta dùng ký hiệu $\|$ là ghép xâu).

1. $S'(k,m) = S(k,\ m[0,\ldots,n-2]\|0)$ và $V'(k,m,t) = V(k,\ m[0,\ldots,n-2]\|0,\ t)$
2. $S'(k,m) = S(k,m)[0,\ldots,126]$ và $V'(k,m,t) = \big[V(k,\ m,\ t\|0)$ hoặc $V(k,\ m,\ t\|1)\ \big]$
   (i.e., $V'(k,m,t)$ outputs "1" nếu hoặc $t\|0$ hoặc $t\|1$ là tag hợp lệ cho m).
3. $S'(k,m) = S(k,\ m\|m)$ và $V'(k,m,t) = V(k,\ m\|m,\ t)$.
4. $S'(k,m) = S(k,m)$ và $V'(k,m,t) = \begin{cases} V(k,m,t) & \text{if } m \neq 0^n \\ \text{"1"} & \text{otherwise} \end{cases}$
5. $S'(k,m) = S(k,m)$ và
   $V'(k,m,t) = \big[V(k,\ m,\ t)$ or $V(k,\ m \oplus 1^n,\ t)\big]$
   (cụ thể, $V'(k,m,t)$ outputs "1" nếu t là một tag hợp lệ cho hoặc m hoặc $m \oplus 1^n$).
6. $S'((k_1,k_2),\ m) = \big(S(k_1,m), S(k_2,m)\big)$ và
   $V'\big((k_1,k_2), m, (t_1,t_2)\big) = \big[V(k_1,m,t_1)$ and $V(k_2,m,t_2)\big]$
   (cụ thể, $V'\big((k_1,k_2), m, (t_1,t_2)\big)$ outputs "1" nếu cả t_1 và t_2 đều là tag hợp lệ).

Câu hỏi 3

Nhắc lại rằng ECBC-MAC dùng một IV cố định (trong bài giảng chúng ta đơn giản đặt IV bằng 0). Giả sử nếu ta chọn IV ngẫu nhiên cho mỗi thông điệp bằng cách ký và kèm IV trong tag. Nói cách khác, $S(k,m) := \big(r,\ \text{ECBC}_r(k,m)\big)$ ở đó $\text{ECBC}_r(k,m)$ tham chiếu đến hàm ECBC dùng r như IV. Thuật toán kiểm tra V nhận k, thông điệp m, và tag (r,t) outputs "1" nếu $t = \text{ECBC}_r(k,m)$ và outputs "0" trong trường hợp ngược lại.

[1] https://class.coursera.org/crypto-012/

Hệ MAC xây dựng theo cách này là không an toàn. Kẻ tấn công có thể truy vấn cho thông điệp kích thước 1-block m và nhận được tag (r, t). Anh ta, sau đó, sinh ra thông điệp giả mạo như sau: (ta giả sử rằng hệ mã khối sử dụng là hệ trên khối n-bit)

1. Tag $(r,\ t \oplus r)$ là một tag hợp lệ cho thông điệp kích thước 1-block 0^n.

2. Tag $(m \oplus t,\ t)$ là một tag hợp lệ cho thông điệp kích thước 1-block 0^n.

3. Tag $(r \oplus 1^n,\ t)$ là tag hợp lệ cho thông điệp kích thước 1-block $m \oplus 1^n$.

4. Tag $(m \oplus t,\ r)$ là tag hợp lệ cho thông điệp kích thước 1-block 0^n.

Câu hỏi 4

Giả sử Alice đang phát quảng bá (broadcasting) các gói tin cho 6 người $B_1, \ldots, B_6$. Những người nhận nên đảm bảo rằng các gói tin anh ta nhận được gửi bởi Alice.

Alice quyết định sử dụng MAC. Giả sử Alice và $B_1, \ldots, B_6$ cùng chia sẻ một khóa bí mật k. Alice tính tag cho mọi gói tin cô ấy gửi sử dụng khóa k. Mỗi người dùng B_i kiểm tra tag khi nhận gói tin và loại bỏ gói tin nếu tag không hợp lệ. Alice để ý rằng sơ đồ này không an toàn bởi vì người dùng B_1 có thể dùng khóa k để gửi tag hợp lệ cho người dùng $B_1, \ldots, B_6$ và anh ta có thể lừa mọi người nghĩ rằng các gói tin này được gửi từ Alice.

Vì vậy, Alice đặt một tập gồm 4 khóa bí mật $S = \{k_1, \ldots, k_4\}$. Cô ấy đưa cho mỗi người dùng B_i một tập con $S_i \subseteq S$ khóa. Khi Alice truyền một gói tin cô ấy thêm 4 tags vào gói tin bằng cách tính tag tương ứng với 4 khóa cô ấy có. Khi người dùng B_i nhận một gói tin, anh ấy chấp nhận nó là đúng nếu và chỉ nếu mọi tag tương ứng với các khóa trong S_i của anh ấy là đúng. Ví dụ, nếu người dùng B_1 được đưa $\{k_1, k_2\}$ anh ấy sẽ chấp nhận một gói tin đến chỉ nếu cả tag đầu tiên và tag thứ hai đều hợp lệ. Để ý rằng B_1 không thể kiểm tra tính hợp lệ của tag thứ 3 hoặc thứ 4 vì anh ta không có k_3 hoặc k_4.

Alice nên gán khóa cho 6 người dùng thế nào để không có người dùng nào có thể mạo danh Alice để gửi tin cho người dùng khác?

1. $S_1 = \{k_1, k_2\},\ S_2 = \{k_1\},\ S_3 = \{k_1, k_4\},\ S_4 = \{k_2, k_3\},\ S_5 = \{k_2, k_4\},\ S_6 = \{k_3, k_4\}$

2. $S_1 = \{k_1, k_2\},\ S_2 = \{k_1, k_3\},\ S_3 = \{k_1, k_4\},\ S_4 = \{k_2, k_3, k_4\},\ S_5 = \{k_2, k_3\},\ S_6 = \{k_3, k_4\}$

3. $S_1 = \{k_1, k_2\},\ S_2 = \{k_1, k_3\},\ S_3 = \{k_1, k_4\},\ S_4 = \{k_2, k_3\},\ S_5 = \{k_2, k_4\},\ S_6 = \{k_3, k_4\}$

4. $S_1 = \{k_1, k_2\},\ S_2 = \{k_1, k_3\},\ S_3 = \{k_1, k_4\},\ S_4 = \{k_2, k_3\},\ S_5 = \{k_2, k_4\},\ S_6 = \{k_4\}$

Câu hỏi 5

Xét CBC MAC với hàm mã hóa dựa trên AES. Giả sử rằng ta tính tag cho một thông điệp dài m bao gồm n khối AES. Xét m' là thông điệp dài n-block sinh từ m bằng cách đảo bit cuối của m (cụ thể, nếu bit cuối của m là b thì bit cuối của m' là $b \oplus 1$). Cần bao nhiêu lần gọi AES để có thể tính tag cho m' từ tag của m và khóa MAC? (trong câu hỏi này, ta có thể bỏ qua việc padding mà đơn giản giả sử rằng độ dài thông điệp là chia hết cho kích thước khối của AES)

1. 4

2. 5

3. 2

4. n

Câu hỏi 6

Xét $H : M \to T$ là một hàm băm kháng xung đột. Hàm băm nào dưới đây cũng là kháng xung đột: (như thường lệ, ta dùng ký hiệu $\|$ cho phép toán ghép xâu).

1. $H'(m) = H(m\|m)$
2. $H'(m) = H(m) \oplus H(m)$
3. $H'(m) = H(|m|)$ (cụ thể, hash độ dài của m)
4. $H'(m) = H(m) \bigoplus H(m \oplus 1^{|m|})$ (với $m \oplus 1^{|m|}$ là phần bù của m)
5. $H'(m) = H(m)\|H(m)$
6. $H'(m) = H(0)$
7. $H'(m) = H(m)\|H(0)$

Câu hỏi 7

Giả sử rằng H_1 và H_2 là các hàm băm kháng xung đột ánh xạ các input trong tập M vào $\{0,1\}^{256}$. Mục đích của chúng ta là chỉ ra rằng hàm $H_2(H_1(m))$ cũng là kháng xung đột. Chúng ta chứng minh bằng phản chứng như sau: giả sử $H_2(H_1(\cdot))$ không kháng xung đột, có nghĩa rằng, ta có $x \neq y$ thỏa mãn $H_2(H_1(x)) = H_2(H_1(y))$. Ta xây dựng một xung đột hoặc cho H_1 hoặc cho H_2. Điều này chỉ ra rằng nếu H_1 và H_2 là kháng xung đột thì $H_2(H_1(\cdot))$ cũng kháng xung đột. Khẳng định nào dưới đây là đúng:

1. Hoặc $x, H_1(y)$ là xung đột cho H_2 hoặc $H_2(x), y$ là xung đột cho H_1.
2. Hoặc x, y là xung đột cho H_2 hoặc $H_1(x), H_1(y)$ là xung đột cho H_1.
3. Hoặc x, y là xung đột cho H_1 hoặc $H_1(x), H_1(y)$ là xung đột cho H_2.
4. Hoặc $H_2(x), H_2(y)$ là xung đột cho H_1 hoặc x, y là xung đột cho H_2.

Câu hỏi 8

Trong câu hỏi dưới đây, bạn được hỏi tìm xung đột cho hai hàm nén:

$$f_1(x,y) = \text{AES}(y,x) \bigoplus y \qquad \text{and}$$
$$f_2(x,y) = \text{AES}(x,x) \bigoplus y$$

với $\text{AES}(x,y)$ là mã hóa AES-128 của y dưới khóa x.

Mục đích của bạn là tìm hai cặp phân biệt $(x_1,y_1),(x_2,y_2),(x_3,y_3),(x_4,y_4)$ thỏa mãn $f_1(x_1,y_1) = f_1(x_2,y_2)$ and $f_2(x_3,y_3) = f_2(x_4,y_4)$. Nói cách khác, hai cặp đầu tiên là xung đột cho f_1 và hai cặp sau là xung đột cho f_2.

Câu hỏi 9

Xét $H : M \to T$ là một hàm băm ngẫu nhiên với $|M| \gg |T|$ (cụ thể kích thước của M lớn hơn kích thước của T). Trong bài giảng ta đã chỉ ra rằng việc tìm xung đột cho H có thể thực hiện trong

$O(|T|^{1/2})$ lần lấy mẫu ngẫu nhiên của H. Bao nhiêu mẫu ngẫu nhiên cần lấy cho đến khi ta đạt được ba xung đột, cụ thể, là ba xâu x, y, z trong M sao cho $H(x) = H(y) = H(z)$?

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