TP 3 - Definitions and requirements

jeudi 18 janvier 2024 18:54

1 Second preimage resistance $\overset{?}{\Rightarrow}$ collision resistance $\not\Leftarrow$

$\Rightarrow$: Let as imagine an adversary that can find a second preimage of $y = hash(x)$

↳ he can find $x' \neq x$ s.t. $h(x') = y = h(x)$

By definition of the second preimage, he is given x. Thus he can produce a collision since $x' \neq x$ and $h(x) = h(x')$ □

Let's assume that a hash function is collision resistant. If an adversary could find a second preimage, he would be able to find a collision, which was assumed to be infeasible

↳ it is also infeasible to find a second preimage

$\Rightarrow$ collision resistance $\Rightarrow$ second preimage resistance

If a hash function is second preimage resistant, a collision could nevertheless be found by other means like second preimage attack.

Intuitively, a collision attack does not require a specific image y.

So a collision attack in general does not help finding a preimage for the given image y.

$\Rightarrow$ collision resistance $\not\Rightarrow$ second preimage resistent

2 XOF h of length n bytes

1. Start with $n=1$ $\Rightarrow$ $l=8$ bits after how many iteration do we have a collision
2. Increase n until this starts to take too much time, which law gives $t(n)$?

$t(n) = 2^{l/2} = 2^{4n}$ Birthday Paradox

3 $H = hash(m)$ flip first bit of $m = m'$ $H' = h(m)$

→ No, there would be no similarities between H and H'

→ Hamming Distance $(H, H') = \frac{l}{2}$ Birthday Paradox

4 $l = 320$ bits of output ++ block 288 bits CV 480 bits

iterated functions → cuts input in blocks of 192 bits, process sequentially and $CV_i = 224$ bits

$\Rightarrow$ adversary can either find output or internal collision

⇒ adversary can either find output or internal collision

↳ $\sqrt{2^{320}} = 2^{160}$ attempts to find a collision (birthday paradox) → 2^{112} attempts for collision resist

→ $\sqrt{2^{224}} = 2^{112}$ attempts

H → 2^{160} or 2^{240} ⇒ 2^{160} attempts

5 output_block = AES_{key}(input_block)

$f_1(x,y) = AES_x(y) \oplus x$

$f_2(x,y) = AES_x(x) \oplus y$

collision for f_1 and f_2

find a_1, b_1 and a_2, b_2 s.t. $f_1(a_1, b_1) = f_1(a_2, b_2)$

and c_1, d_1 c_2, d_2 s.t. $f_2(a_1, d_1) = f_2(c_2, d_2)$

→ for f_1: x, y, x', x' s.t. $AES_x(y') \oplus x' = AES_x(y) \oplus x$

⇒ $y' = AES^{-1}_{x'}(AES_x(y) \oplus x \oplus x')$ and x, x', y random

→ for f_2: $x\ y\ x'\ y'$ s.t. $AES_x(x) \oplus y = AES_{x'}(x') \oplus y'$

x, x' random ⇒ $y = AES_{x'}(x')$ ✓ $y' = AES_x(x)$

6 SHAES : compression function from AES-256 in the Davies Meyer construction

1. Size of SHAES : as Davies-Meyer is a block cipher of fixed size (key size) ⇒ 256 bits
2. Chaining value : equal to the block size of AES-256 → 128 bits
3. SHAES a valid option ? 128 bits of CV → 2^{64} attempts is kinda weak

7 $x = (x_0, x_1, \dots, x_{n-1}) \in \mathbb{Z}_2^n$ mapped to the integer $X = \sum_i x_i 2^i = \text{integer}(x)$

$x = \text{vector}(X)$

1. $f(x,y) = \text{vector}(\text{integer}(x) + \text{integer}(y))$ Modular addition $x+y$ is non-linear in $\mathbb{Z}_2^n$

Let $n=2$ $f(0,0) = 0$ ⇒ $f(x \oplus x', y \oplus y') \neq f(x,y) \oplus f(x',y')$

Let $n=2$ $f(0,0)=0$ $\Rightarrow f(x\oplus x', y\oplus y') \neq f(x,y)\oplus f(x',y')$

if $(x,y)=(1,0)$ and $(x',y')=(0,1) \Rightarrow f(1,1)=2$ and $f(1,0)\oplus f(0,1) = 1\oplus 1 = 0$

2. $F(x,y)$ = integer (vector(X) $\oplus$ vector(Y)) is not linear

$f(x+x', y+y') \neq f(x,y) + f(x',y')$

$(X,Y)=(1,0)$ $(X',Y')=(0,1) \Rightarrow f(1,1)=0$ $\quad f(0,1)+f(1,0)=1+1=2$

8 RO = random oracle → outputs an infinite sequence of uniformly and ll distributed bit for each input $x \in \{0,1\}^*$

RO(x, l) if truncate outputs to l first bits

1. Preimage attack, given $y \in \{0,1\}^l \Rightarrow$ find x? s.t. RO(x, l) = y

→ Proba = ? Assume $t \ll 2^l$

$\Rightarrow$ Proba $\approx \frac{t}{2^l}$ $= \frac{\#\text{tries}}{\#\text{possibilities}}$

2. Multi-target preimage attack: given m distinct values $\{y_1, \ldots, y_m\}$ with $y_i \in \{0,1\}^l$

$\Rightarrow$ Proba $\approx \frac{t \cdot m}{2^l}$ $= \frac{\#\text{tries} \cdot \#y}{\#\text{possibilities}}$

b(x) = XOF b(x, l) = truncated to l known attack → $t/2^{c/2}$ ← security

3. l and c to have 128 bits preimage resistance

Proba $= \frac{t}{2^l} + \frac{t}{2^{c/2}} = t\left(2^{-l} + 2^{-c/2}\right) \approx t\,2^{\max(-l, -\frac{c}{2})} \approx t \cdot 2^{-\min(l, \frac{c}{2})}$

$\Rightarrow 128 : t = 2^{128}$ $\quad l \geq 128$ $\quad c/2 \geq 128$ $\quad c \geq 256$

4. Multi preimage: $m = 2^{32}$

Proba $= \frac{mt}{2^l} + \frac{t}{2^{c/2}} = t\,2^{\max(-l+\log_2(m), -c/2)} = t\,2^{-\min(l-\log_2(m), c/2)}$

128 bit → $t = 2^{128}$ $\quad l - 32 \geq 128$ $\quad c/2 \geq 128$

128 bit → $t = 2^{128}$ $\quad l - 32 \geq 128$ $\quad c/2 \geq 128$

$l \geq 160$ $\quad c \geq 256$

9 π = permutation $\quad b = 1600 \rightarrow [\pi] \rightarrow$ b-bit string

$c = 0$ $\quad l$ output → $n = 256$ bits

1. How to obtain a collision

With $c=0$ → no security

→ resetting the state to its initial value $0^b \Rightarrow$ tozero $= \pi^{-1}(0^b)$

→ After absorbing tozero, the state will be 0^b

$\Rightarrow$ for all s $\quad l(\text{tozero}|s) = l(s)$ □

2. Given two prefixes x and y of b bits each, have a collision $s_1 = x||\ldots$ and $s_2 = y||\ldots$

→ $\Pi(x) \oplus x' = \Pi(y) \oplus y' \iff x' \oplus y' = \Pi(x) \oplus \Pi(y)$ is suitable

(x', y') and $(\Pi(x), \Pi(y)) \Rightarrow l(x|x') = l(y|y')$

3. Given an output z of n bits → how to find a preimage?

$\Rightarrow l(x) = z \Rightarrow$ extend z (z') → 1600−256 bits and consider this as the last step of the last function $s_1 = z||z'$

$s_0 = \pi^{-1}(s_1) \Rightarrow$ unpad s_0 by removing the last bits 0 until a 1 is found and remove

↳ remove padding

$\Rightarrow$ works only if $s_0 \neq 0^b$, else try with another z'

4. Hashes ASCII input text, most significant bit ALWAYS TO 0.

↳ input string m s.t. 8th bit is 0.

How to find a collision, what is the complexity of our attack.

$\Rightarrow \frac{b}{8} = 200$ bits restricted

$\Rightarrow$ as a sponge with $c = 200$

$\Rightarrow$ as a sponge with $c = 200$

$\Rightarrow$ To get a collision, finding x and y s.t. after being absorbed, the values of the c elements is identical.

$\Rightarrow$ $\pi(x)$ and $\pi(y)$ is thus the state

$\Rightarrow$ Find x', y' s.t. cancel the $\neq$ between $\pi(x)$ and $\pi(y)$

$\Rightarrow$ $h(x|x') = h(y|y')$

The time consuming is to find the c elem. $\Rightarrow$ $2^{\frac{200}{2}} = 2^{100}$ attempts

5. Given z of n bits $\rightarrow$ find a preimage

as 3. but $z_0 = \pi'(z|z')$ must have a c part $= 0$ for it to satisfy the restriction

$\Rightarrow 2^{200}$.

III K is a k-bit secret-key

1. $MAC_K(\text{message}) = \text{hash}(\text{message}) \oplus K$ EO-CMA

Recover the key $K = MAC_K(m) \oplus h(m)$

$\rightarrow$ adversary query tag $= MAC_K(m)$ and recovers K as explained above

$\rightarrow$ " " $tag' = MAC_K(m')$ with $m \neq m'$

$\rightarrow$ adversary wins a pair (tag', m') not queried before □

2. $MAC_K(m) = h(K_1 \| m) \oplus K$ $K = K_1 \| K_2$ $|K_1| = |K|_2 = k/2$

$\rightarrow$ security is $k/2$ $\Rightarrow 2^{k/2}$

$\rightarrow$ get a known (m, tag) by querying

↳ For each value possible of K_1^* for first $k/2$ bits of K

- compute $K^* = \text{hash}(K_1^* \| m) \oplus tag$
- if first $k/2$ bits of K^* do not match K_1^*, this cannot be a correct guess

↳ else: double check by sending another (m, tag) pair.