HMAC : Keyed-Hash Message Authentication Code

MAHTAT Yassin

9 février 2020

1 Code d'Authentification de Message

CAM – Message Authentication Code – systéme d'authentification de message à partir d'une clé sécrète partagée. Qui ne connait pas la clé secrète, ne puisse fabriquer que de faux messages.

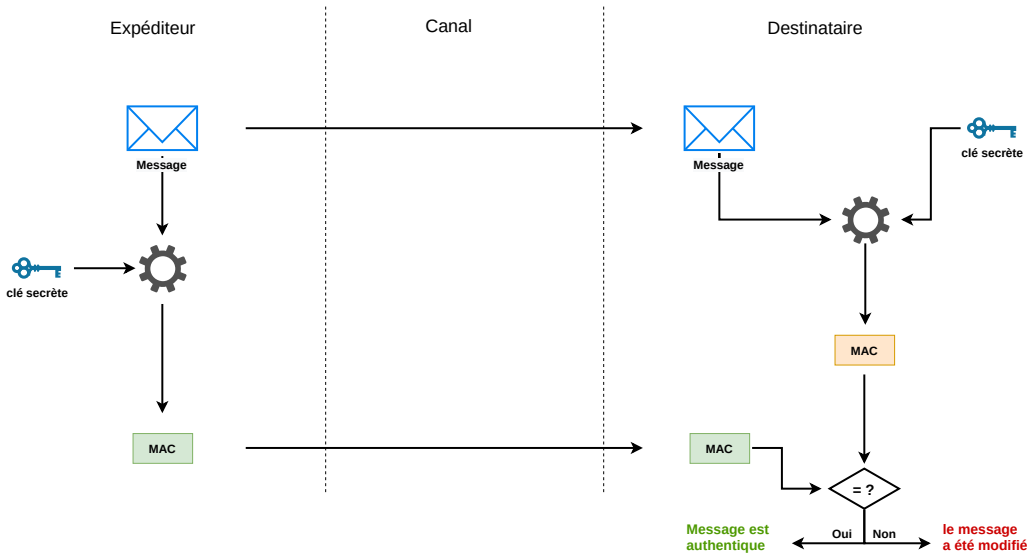

FIGURE 1 – Code d'Authentification de Message

2 Distance de Hamming

2.1 Fonction OU exclusif (XOR)

La fonction OU exclusif, est un opérateur logique de l'algèbre de Boole. À deux opérandes, qui peuvent avoir chacun la valeur VRAI ou FAUX, il associe un résultat qui a lui-même la valeur VRAI seulement si les deux opérandes ont des valeurs distinctes.

Appelons a et b les deux opérandes considérés. Convenons de représenter leur valeur ainsi :

1 = VRAI
0 = FAUX

L'opérateur XOR est défini par sa table de vérité, qui indique pour toutes les valeurs possibles de a et b la valeur du résultat $a \oplus b$:

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Soit $a, b \in \{0, 1\}$, nous avons

$$a \oplus b \ = \ \bar{a}b + a\bar{b}$$

2.2 Distance de Hamming

Soit A un alphabet et F l'ensemble des suites de longueur n à valeur dans A. La distance de Hamming entre deux éléments a et b de F est le nombre d'éléments de l'ensemble des images de a qui diffèrent de celle de b.

Formellement, si $d(\cdot, \cdot)$ désigne la distance de Hamming :

$$\forall a, b \in F \quad a = (a_i)_{i \in I} \ \text{ et } \ b = (b_i)_{i \in I} \quad \Rightarrow \quad d(a, b) = \text{card}\{i \mid a_i \neq b_i\}$$

Cas binaires, si $A = \{0, 1\}$, nous avons

$$d(a, b) \ = \ \sum_{i=0}^{n-1} (a_i \oplus b_i)$$

Exemple

Soient $a = (0\ 0\ 0\ 1\ 1\ 1\ 1)$ et $b = (1\ 1\ 0\ 1\ 0\ 1\ 1)$, alors
$d(a, b) \ = \ 1 + 1 + 0 + 0 + 1 + 0 + 0 \ = \ 3$

2.3 Propriétés

Symétrie

$\forall a, b \in F, \quad d(a, b) \ = \ d(b, a)$

Séparation

$$\forall a, b \in F, \quad d(a,b) = 0 \quad \Leftrightarrow \quad a = b$$

Inégalité triangulaire

$$\forall a, b, c \in F, \quad d(a,c) \leq d(a,b) + d(b,c)$$

3 H-MAC – keyed-hash message authentication code

3.1 construction

HMAC – Keyed-Hash Message Authentication Code – permet la génération d'un CAM à partir d'une fonction de hachage cryptographique h. Il est utilisé dnas les protocoles IpSec et SSL.

La fonction HMAC est définie comme suit :

$$\mathrm{HMAC}_K(m) \;=\; h\Big((K \oplus \mathrm{opad}) \;\Big\|\; h\Big((K \oplus \mathrm{ipad}) \,\|\, m\Big)\Big)$$

— h : Une fonction de hachage itérative.
— m : Le message à authentifier.
— K : La clé secrète.
— $\|$ désigne une concaténation et $\oplus$ un OU-exclusif.

opad = 0x 5c 5c ... 5c et ipad = 0x 36 36 ... 36 :

— Ils sont tous les deux de la taille d'un bloc.

— Si la taille de bloc de la fonction de hachage est 64-octets, opad et ipad sont 64 répétitions des octets, respectivement, 0x5c et 0x36.

— Ils ont été choisis afine d'avoire une distance de Hamming importante.

3.2 Algorithme

Algorithm 1: The HMAC Algorithm

Input: key :secret key, msg :message, hash :hash function
Output: return a hmac value

```
begin
    # calculate the key size
    key_size ← length(key)
    # calculate the block size
    block_size ← hash.block_size
    # step 1 – determine key
    if key_size > block_size :
        # hash 'key', then append zeros to create a
        # block_size-bytes string 'key'
        key ← hash(key) ‖ 0x00...0
    elif key_size < block_size :
        # Append zeros to the end of 'key', to create a
        # block_size-bytes string 'key'
        key ← key ‖ 0x00...0
    # step 2 – calculate h(K ⊕ opad)
    b1 ← hash(key ⊕ opad)
    # step 3 – calculate h((K ⊕ ipad) ‖ m)
    b2 ← hash((key ⊕ ipad) ‖ msg)
    #step 4 – calculate h((K ⊕ opad) ‖ h((K ⊕ ipad) ‖ m))
    return hash(b1 ‖ b2)
```

Références

[1] Norman Mineta, Cheryl Shavers, Under Technology, and Raymond Kammer. The keyed-hash message authentication code (hmac). 02 2001.