Fonction de Hachage SHA1

Cryptographie asymétrique

Partie: 3

Prof:Mme F.Omary

2018-2019

Introduction&Historique

- Les **fonctions de hachage SHA** sont un ensemble de fonctions de hachage cryptographique conçu par la National Security Agency (NSA).

- Les algorithmes SHA sont structurées différemment et se distinguent comme *SHA-0, SHA-1 , SHA-2* ...

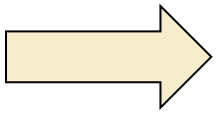
- SHA-1 est la mieux établie des fonctions de hachage actuelles SHA, et est employée dans plusieurs applications de sécurité et les protocoles.

Introduction (suite)

- **SHA-0** (*Secure Hash Algorithm*) ,
  - CommeMD5 basé sur MD4.
  - Fonctionne également à partir de blocs de 512 bits
  - Produit par contre des condensés de 160 bits en sortie.
- Le **SHA-0** Pour des raisons de sécurité insuffisante. était légitimement soupçonné de contenir des failles qui permettraient d'aboutir rapidement à des collisions
- le **SHA-0** s'est vu modifié peu après sa sortie (1993) et complexifié pour obtenir le **SHA-1** (1995).

Introduction(suite)

- **SHA-1** semble offrir une plus grande résistance aux attaques, le soutien à la NSA affirme que le changement a augmenté la sécurité.

- C’ est une fonction de hachage crée par NSA en 1995 afin de remédier à la vulnérabilité de son prédécesseur le SHA-0
- Attaque théorique (2^ 63) 

Successeurs: SHA-224,SHA-256, SHA-384,SHA-512

Description de SHA-1

- **Les caractéristiques de SHA-1:**

- taille du message : 2^64 bits maximum
- taille des blocs : 512 bits
- taille des mots : 32 bits
- taille du condensé : 160 bits

Description (suite)

- **SHA-1** utilise une succession de fonctions logiques utilisées lors du calcul des valeurs de hachage:

$$f_t(x,y,z) = \begin{cases} Ch(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z), & \text{si } 0 \leq t \leq 19 \\ Parity(x,y,z) = x \oplus y \oplus z, & \text{si } 20 \leq t \leq 39 \\ Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z), & \text{si } 40 \leq t \leq 59 \\ Parity(x,y,z) = x \oplus y \oplus z, & \text{si } 60 \leq t \leq 79 \end{cases}$$

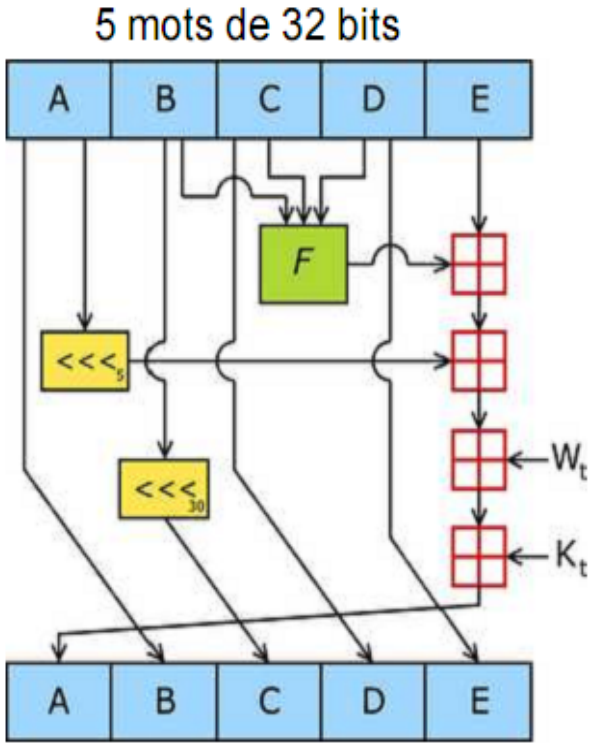
- **SHA-1** utilise quatre valeurs réparties dans les 80 constantes **:**

$$K_t = \begin{cases} \text{0x5a827999}, & \text{si } 0 \leq t \leq 19 \\ \text{0x6ed9eba1}, & \text{si } 20 \leq t \leq 39 \\ \text{0x8f1bbcdc}, & \text{si } 40 \leq t \leq 59 \\ \text{0xca62c1d6}, & \text{si } 60 \leq t \leq 79 \end{cases}$$

Schéma d'un tour de SHA-1

SHA-1 :

- Composé de la répétition de 80 opérations regroupées en 4 fois 20 opérations
  - K_t = constante
  - W_t = valeur dépendant des blocs M_i du message



Une opération

$$f_t(x,y,z) = \begin{cases} Ch(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z), & \text{si } 0 \leq t \leq 19 \\ Parity(x,y,z) = x \oplus y \oplus z, & \text{si } 20 \leq t \leq 39 \\ Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z), & \text{si } 40 \leq t \leq 59 \\ Parity(x,y,z) = x \oplus y \oplus z, & \text{si } 60 \leq t \leq 79 \end{cases}$$

- Concernant Wt voir plus loin l'algorithme SHA1

- L'algorithme de SHA-1 peut être découpé en deux phases : le prétraitement et le calcul du condensé.

Prétraitement

- **Le prétraitement**

Cette opération se déroule en trois étapes :

- Complétion(padding) du message à hacher M pour que sa longueur soit multiple de 512:
  1. Le symbole 1 est ajouté à la fin de M
  2. Un nombre minimal de 0 sont ajoutés à la fin de M afin que :IMI=n*512-64
  3. Longueur de M est écrite en base 2 comme

Prétraitement (suite)

un nombre de 64 bits, collé à la fin du mot obtenu précédemment

- Découpage du résultat obtenu en blocs de 512bits
- Initialisation des valeurs de hachage.
- H0=67452301, H1=EFCDAB89, H2=98BADCFE, H3=10325476,H4=C3D2E1F0

Algorithme SHA-1

- On exécute la procédure suivante, pour i=1,2,…,n

  1. Ecrire **Mi** comme une suite $\mathbf{M_i=W_0W_1…W_{15}}$ de 16 mots de 32 bits.

  2. Pour t=16,17,…,79 calculer:

  $\mathbf{W_t= S^1(W_{t-3}\oplus W_{t-8}\oplus W_{t-14}\oplus W_{t-16})}$

  3. Poser $\mathbf{A=H_0}$, $\mathbf{B=H_1}$, $\mathbf{C=H_2}$,$\mathbf{D=H_3}$, et $\mathbf{E=H_4}$,

  4.Pour t=0,1,…,79 calculer

  $\mathbf{T=S^5(A)+f_t(B,C,D)+E+W_t+K_t}$, E=D, D=C, $\mathbf{C=S^{36}(B)}$, B=A, A=T.

Algorithme SHA-1(suite)

5. calculer

$H_0=H_0+A$, $H_1=H_1+B$, $H_2=H_2+C$, $H_3=H_3+D$, $H_4=H_4+E$.

- la valeur hachée est

  $SHA\text{-}1(x)=H_0H_1H_2H_3H_4$

- **Note**: $S^k(W)$ désigne un décalage à gauche de k bits, circulaire d'une chaîne w

  \+ désigne l'addition modulo 2^{16}

Sécurité

- Attaque des anniversaires:
  - On calcule autant de valeurs hachées que le temps et l'espace le permettent
  - Ces valeurs sont stockées avec leur images inverses et triées afin de chercher une collision.
  - Nous pouvons analyser cette procédure en utilisant le paradoxe des anniversaires:
- Les valeurs hachées sont les anniversaires

Sécurité(suite)

- On suppose que les chaînes de caractères peuvent être choisies dans Σ^* de façon que la distribution correspondante, sur les valeurs hachées, soit uniforme.

– On a montré le résultat suivant:

- Si k chaînes de caractères sont choisies dans Σ^* avec:

$$k \geq \frac{(1+\sqrt{1+(8\times \ln(2)\times |\sum|^n)})}{2}$$

Sécurité(suite)

- La probabilité que deux valeurs hachées soient égales est supérieure à ½
- Pour simplifier on suppose que Σ={0,1}, **alors:**

$$k \geq \frac{(1+\sqrt{1+(8\times \ln(2)\times 2^{n})})}{2}$$

- Donc en calculant un peu plus que $\mathbf{2^{n/2}}$valeurs hachées , l'attaque des anniversaires trouve une collision avec une probabilité >1/2
- Pour empêcher de telles attaques : n doit être

Sécurité (suite)

- n doit être choisie de façon que le calcul de $\mathbf{2^{n/2}}$ valeurs hachées soit infaisable
- Actuellement il est recommandé de prendre n>=160.

Tableau