NOM: ABIO
POSTNOM: BAMONGOYO
PROMOTION: L2 INFORMATIQUE DE GESTION

LABORATOIRE INFORMATIQUE

Sujet “Fonctionnalité de l’algorithme MD5 ”

Ce qu’il faut savoir sur l’algorithme MD5 (Message Digest 5) est une fonction de hachage cryptographique qui calcule, à partir d'un fichier numérique, son empreinte numérique (en l'occurrence une séquence de 128 bits ou 32 caractères en notation hexadécimale) avec une probabilité très forte que deux fichiers différents donnent deux empreintes différentes. Elle n'est plus considérée comme sûre pour un usage en cryptographie car durant l'été 2004, des chercheurs chinois ont montré comment réaliser des collisions avec MD5, c'est-à-dire comment produire deux messages ayant la même empreinte (le même résumé) en appliquant l'algorithme MD5. Cela dit, le MD5 est encore utilisé pour des usages non sensibles, par exemple pour vérifier l'intégrité d'un fichier téléchargé. De plus, le MD5 est une sorte de "mythe" pour certains geeks.

Voyons maintenant comment fonctionne le MD5.

Etape 1 : Complétion

Le message est constitué de b bits m1...mb. On complète le message par un 1, et suffisamment de 0 pour que le message étendu ait une longueur congruente à 448, modulo 512. Puis on ajoute à ce message la valeur de b, codée en binaire sur 64 bits (on a donc b qui peut valoir jusque 264... ce qui est énorme). On obtient donc un message dont la longueur totale est un multiple de 512 bits. On va travailler itérativement sur chacun des blocs de 512 bits.

Etape 2 : Initialisation

On définit 4 buffers de 32 bits A,B,C et D, initialisés ainsi (les chiffres sont hexadécimaux, ie a=10, b=11...).

A=01234567

B=89abcdef

C=fedcba98

D=76543210

On définit aussi 4 fonctions F,G,H et I, qui prennent des arguments codés sur 32 bits, et renvoie une valeur sur 32 bits, les opérations se faisant bit à bit.

F(X,Y,Z) = (X AND Y) OR (not(X) AND Z)

G(X,Y,Z) = (X AND Z) OR (Y AND not(Z))

H(X,Y,Z) = X xor Y xor Z

I(X,Y,Z) = Y xor (X OR not(Z))

Ce qu'il y a d'important avec ces 4 fonctions et que si les bits de leurs arguments X, Y et Z sont indépendants, les bits du résultat le sont aussi.

Etape 3 : Calcul itératif

Pour chaque bloc de 512 bits du texte, on fait les opérations suivantes : on sauvegarde les valeurs des registres dans AA, BB, CC, DD.

on calcule de nouvelles valeurs pour A,B,C,D à partir de leurs anciennes valeurs, à partir des bits du bloc qu'on étudie, et à partir des 4 fonctions F, G, H, I.

on fait A=AA+A, B=BB+B, C=CC+C, D=DD+D.

Le détail des calculs se trouve en annexe.

Etape 4 : Ecriture du résumé

Le résumé sur 128 bits est obtenu en mettant bout à bout les 4 buffers A, B, C, D de 32 bits.

NOM: MICHAEL
POSTNOM: KYLE
PROMOTION: L2 RTEL

LABORATOIRE INFORMATIQUE

Sujet “Fonctionnalité de l’algorithme MD5 ”

Ce qu’il faut savoir sur l’algorithme MD5 (Message Digest 5) est une fonction de hachage cryptographique qui calcule, à partir d'un fichier numérique, son empreinte numérique (en l'occurrence une séquence de 128 bits ou 32 caractères en notation hexadécimale) avec une probabilité très forte que deux fichiers différents donnent deux empreintes différentes.

Voyons maintenant comment fonctionne le MD5.

Etape 1 : Complétion

Le message est constitué de b bits m1...mb. On complète le message par un 1, et suffisamment de 0 pour que le message étendu ait une longueur congruente à 448, modulo 512. Puis on ajoute à ce message la valeur de b, codée en binaire sur 64 bits (on a donc b qui peut valoir jusque 264... ce qui est énorme). On obtient donc un message dont la longueur totale est un multiple de 512 bits. On va travailler itérativement sur chacun des blocs de 512 bits.

Etape 2 : Initialisation

On définit 4 buffers de 32 bits A,B,C et D, initialisés ainsi (les chiffres sont hexadécimaux, ie a=10, b=11...).

A=01234567

B=89abcdef

C=fedcba98

D=76543210

Ce qu'il y a d'important avec ces 4 fonctions et que si les bits de leurs arguments X, Y et Z sont indépendants, les bits du résultat le sont aussi.

Etape 3 : Calcul itératif

Pour chaque bloc de 512 bits du texte, on fait les opérations suivantes : on sauvegarde les valeurs des registres dans AA, BB, CC, DD.

on calcule de nouvelles valeurs pour A,B,C,D à partir de leurs anciennes valeurs, à partir des bits du bloc qu'on étudie, et à partir des 4 fonctions F, G, H, I.

on fait A=AA+A, B=BB+B, C=CC+C, D=DD+D.

Le détail des calculs se trouve en annexe.

Etape 4 : Ecriture du résumé

Le résumé sur 128 bits est obtenu en mettant bout à bout les 4 buffers A, B, C, D de 32 bits.