Travaux Pratique de Cryptographie N° 4

MESSAGE DIGEST N°5

J. VEHENT

Novembre 2004

1.Présentation de MD5

MD5 est une amélioration de MD4, tous deux conçus par Ron Rivest (le R de RSA). MD signifie « Message Digest », qui peut être traduit par « empreinte » en français. Cet algorithme permet de créer la signature digitale d'une entrée numérique. Il produit des empreintes de 128 bits quelque soit la longueur du message en entrée.

Pour ce faire, il procéde la façon suivante:

MD5 manipule des blocs de 512 bits. Il complète la longueur du message en entrée telle que la longueur soit congrue à 448 modulo 512, en rajoutant un 1 suivi d'autant de 0 que nécessaires à la fin du message. L'opération de « padding » a toujours lieu, même si la taille du message est déjà un multiple de 448.

Ensuite, la longueur initiale (avant le padding) du message est rajoutée aux 448 bits, sous forme de 64 bits, ce qui amène à une taille multiple de 512 bits.

Chaque bloc de 512 bits est décomposé en 16 blocs de 32 bits (16 mots) et le résultat est représenté par un ensemble de 4 mots A,B,C et D.

MD5 prend 4 variables en entrée, initialisées en hexadécimal de la manière suivante :

- A : 01 23 45 67
- B : 89 ab cd ef
- C : fe dc ba 98
- D : 76 54 32 10

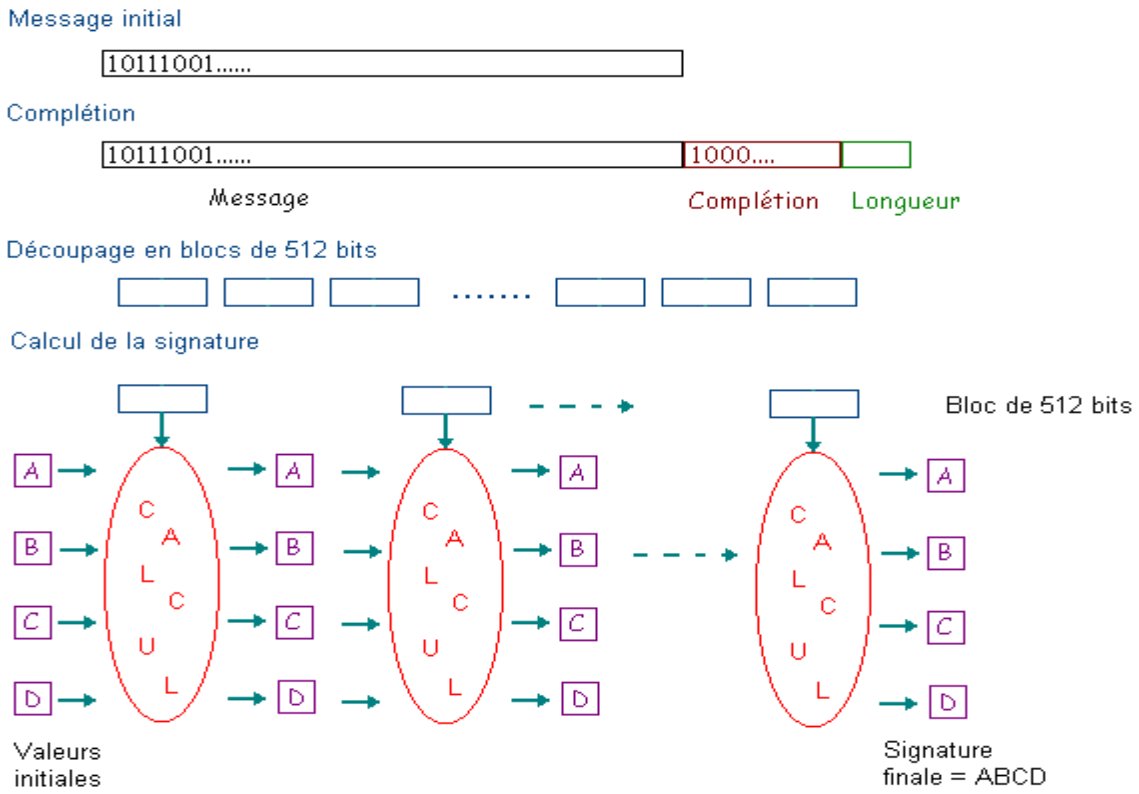
MD5 comprend 4 rondes qui exécutent chacune 16 opérations.

Les 4 fonctions non-linéaires prévues pour chaque ronde sont les suivantes :

- F(X,Y,Z) = XY or not(X) Z
- G(X,Y,Z) = XZ or Y not(Z)
- H(X,Y,Z) = X xor Y xor Z
- I(X,Y,Z) = Y xor (X or not(Z))

Le résultat, après les 4 rondes est représenté par les valeurs contenues dans les 4 variables A, B, C et D.

Voici le fonctionnement sous la forme d'un graphique:



Description du fonctionnement du MD5

La partie “CALCUL” comprend les 4 rondes qui chacunes 16 itérations que nous verrons dans le code source. Comme nous pouvons le voir sur le schéma ci-dessus, les valeurs A,B,C et D sont modifiées à chaque ronde mais pas augmentées. Ceci nous permet d'obtenir une empreinte qui est TOUJOURS de 128 bits, quelque soit la taille du message initial.

2.Le code source

Les codes sources sont disponibles dans l'archive au format HTML (afin de conserver la coloration syntaxique).

Les fichiers sont: md5c.c
md5.h
global.h

3.Création d'une empreinte

L'executable nommé “md5c” nous permet de créer une empreinte d'un fichier passé en argument. En voici un exemple avec le fichier “testempreinte.txt”

```
julien@laptop:~/Cryptographie/tp4$ ./md5c ./testempreinte.txt
MD5 (./testempreinte.txt) = 6c428ff803282c669a1e8400487acdca
```

Si l'on recalcule cette empreinte à nouveau, on obtient :

```
julien@laptop:~/Cryptographie/tp4$ ./md5c ./testempreinte.txt
MD5 (./testempreinte.txt) = 6c428ff803282c669a1e8400487acdca
```

Les resultats sont identiques, comme nous l'attendions. Celà confirme que l'algorithme de calcul de signature digitale “Message Digest n°5” n'a rien d'aléatoire.

4.Conclusion

L'algorithme MD5 fournit une bonne signature des messages. En effet, il est démontré que la difficulté d'obtenir avec deux messages la même empreinte est de l'ordre de 2^64 opérations, et que la difficulté d'obtenir avec n'importe quel message une empreinte pré-établie est de l'ordre de 2^128 opérations. On peut donc considérer que l'intégrité quasi compléte d'un message est assurée avec “Message Digest n°5”.