Fonctions de hachage

Dr. Noureddine Chikouche
Université de M'sila
https://sites.google.com/view/chikouchenoureddine

Plan du cours

- Concepts et définitions
- Fonction MD-5
- Catégories des fonctions de hachage
- Exemples d'utilisations
- Recommandations

- Comment assurer l’intégrité du message?
- Comment assurer l’authentification d’un entité?

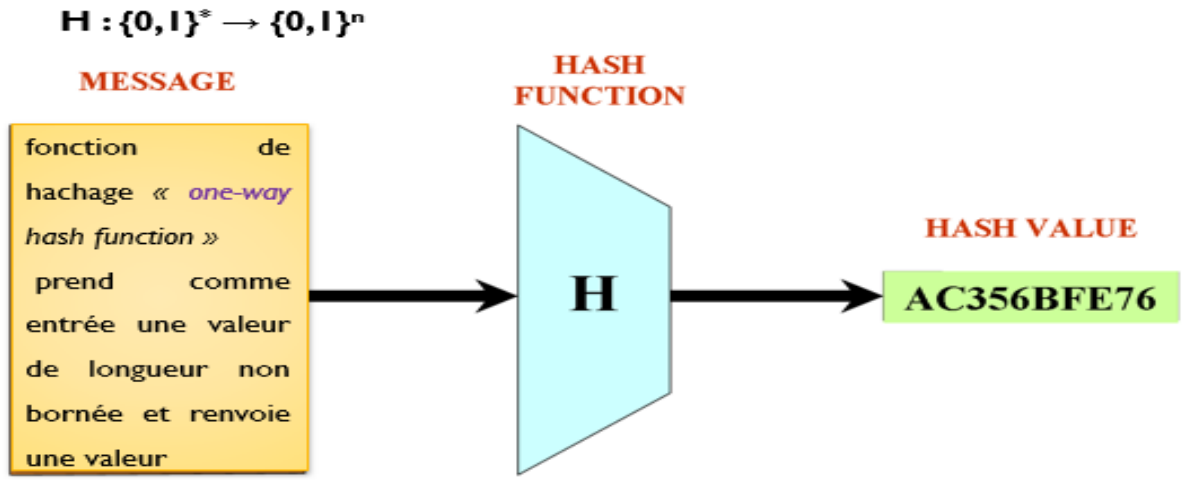
Concepts et définitions

Définition:

- Fonction de hachage est une primitive cryptographique qui produire un message de longueur borne et condensé à partir d'un message de longueur non fixe.

Concepts et définitions

- La sortie (output) de la fonction porte le nom: empreinte, valeur de hachage, haché, ou condensé, (en anglais, *digest*, *message digest* ou *hash*).
- La longueur de l'empreinte est déterminée par le concepteur de la fonction de hachage.



Concepts et définitions

Texte original: Computer Security
Hexadécimal: 43 6f 6d 70 75 74 65 72 20 53 65 63 75 72 69 74 79

F. de Hachage	Hash	n
MD5	F88c612e493e78ab649191e411da7e86	128
SHA-1	d678cf370ad77eb7f7d209d1311c160037423546	160
SHA-256	a3fa73779dd6d4c3fbe123d8ee1b18ed5702ab9a1ff4e5 6fc96393636062d3cf	256
RipeMD-128	c7a050c1dfdfb6bf83dca669a86a3429	128
SHA3-256	b8d60c6c40062dadbb6e551bbb8bc02afd5cac2fb24447 9979c330f2abcf817c	256

Concepts et définitions
Applications

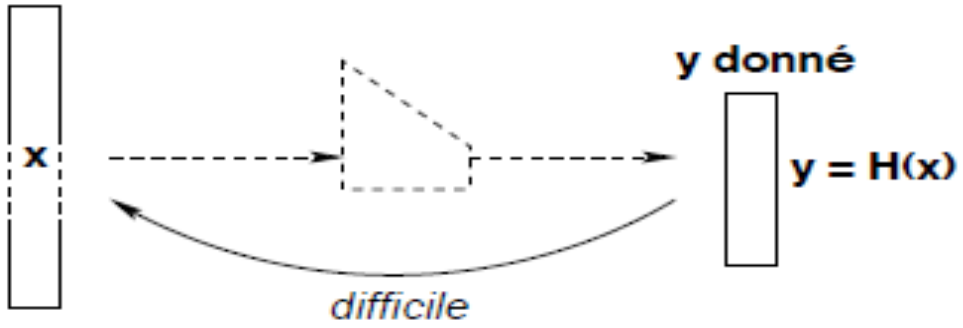
Les fonctions de hachage sont utilisés, par exemple:

- Protection des mots de passe,
- Générateur de nombres pseudo-aléatoires,
- Code d'authentification de message (MAC),
- Pour dériver des clés de session.
- Blockchain.
- Compresser les messages dans la génération et la vérification de la signature numérique.

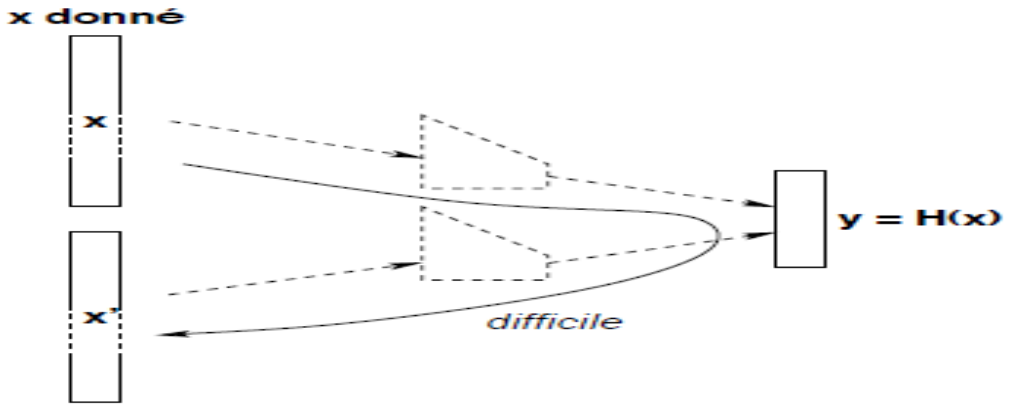
Concepts et définitions
Propriétés

Propriétés:

1) Résistance à la pré-image: Étant donné $y=H(x)$, il est difficile de trouver x.

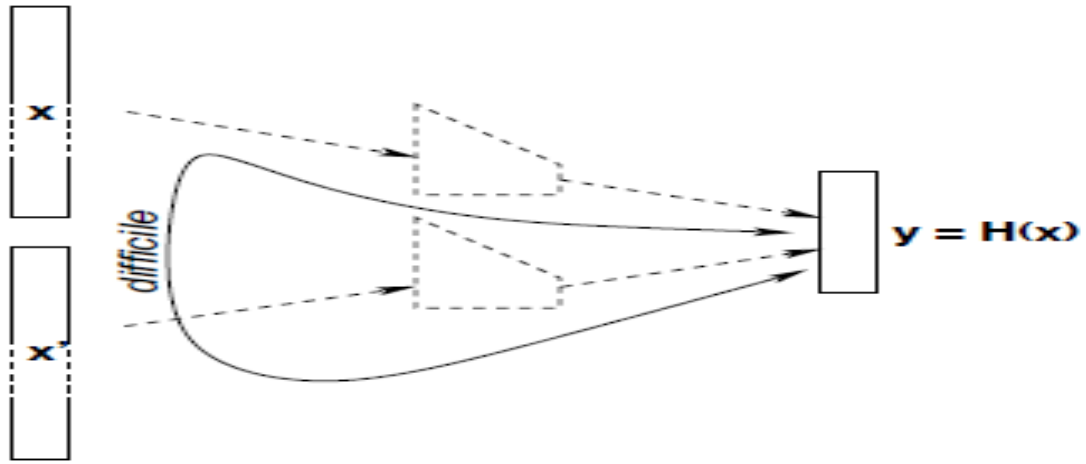


2) Résistance à la seconde pré-image: Étant donné x et H(x), il est dur de trouver x ≠ x' vérifiant H(x)=H(x').

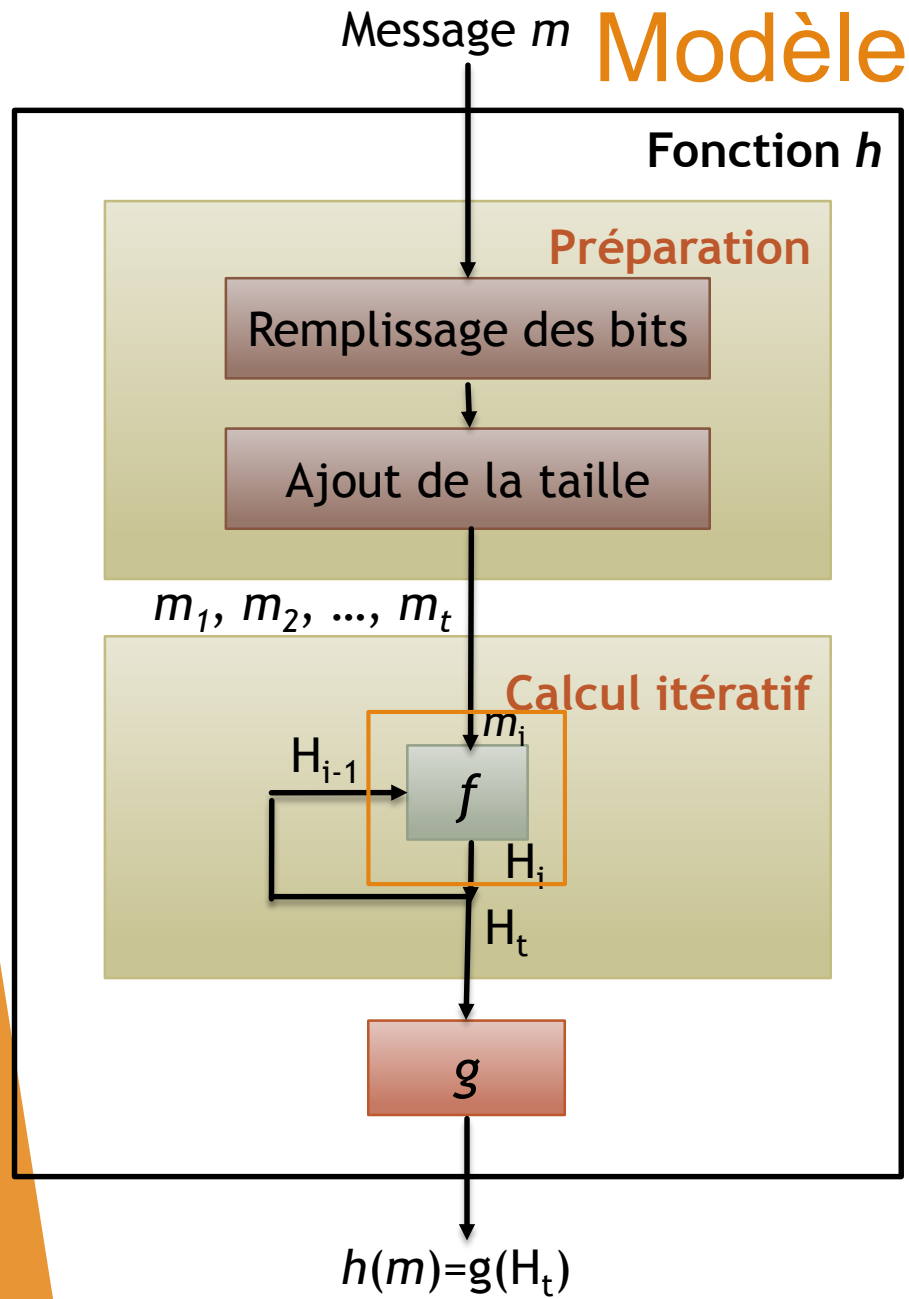


Concepts et définitions
Propriétés

3) Résistance à la collision: il est difficile de trouver deux entrées différentes *x* et *x'* tel que : *h (x) = h (x')*.



Concepts et définitions
Modèle général de construction



- Préparation:
  - compléter le message par des bits. Le message étendu ait une longueur multiple de taille du bloc.
  - Ajouter la taille du message d'entrée
- Calcul itératif:
  - $H_i = f(H_{i-1}, m_i)$, $1 \le i \le t$
  - H_0=VI (vecteur d'initialisation)
  - f : fonction de compression
- Résultat de transformation:
  - $H(m) = g(H_t)$
  - C'est une transformation optionnelle

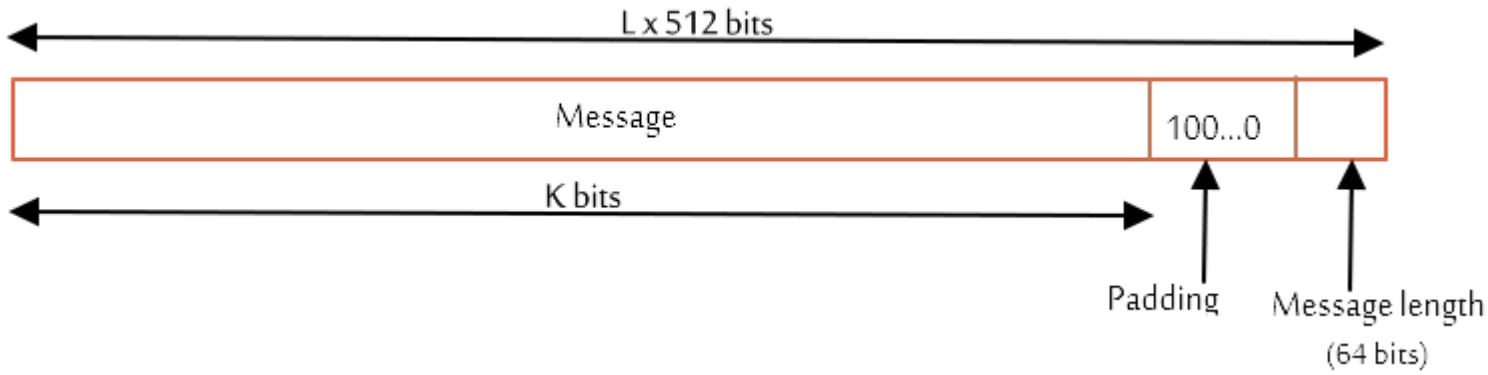
Fonction MD5

- MD5: Message Digest 5
- Conçu par Ronald Rivest en 1992
- Amélioration de la fonction MD4.
- Taille de l'empreinte: 128 bits.
- Taille de bloc: 512 bits.
- Norme IETF: RFC 1321.

Fonction MD5
Algorithme

1. Préparation:

- Compléter le message par un 1, et suffisant de 0 (= 448 mod 512)
- La longueur du message étendu est multiple de 512 bits.
- Diviser chaque bloc en 16 sous-blocs Mi de 32 bits.



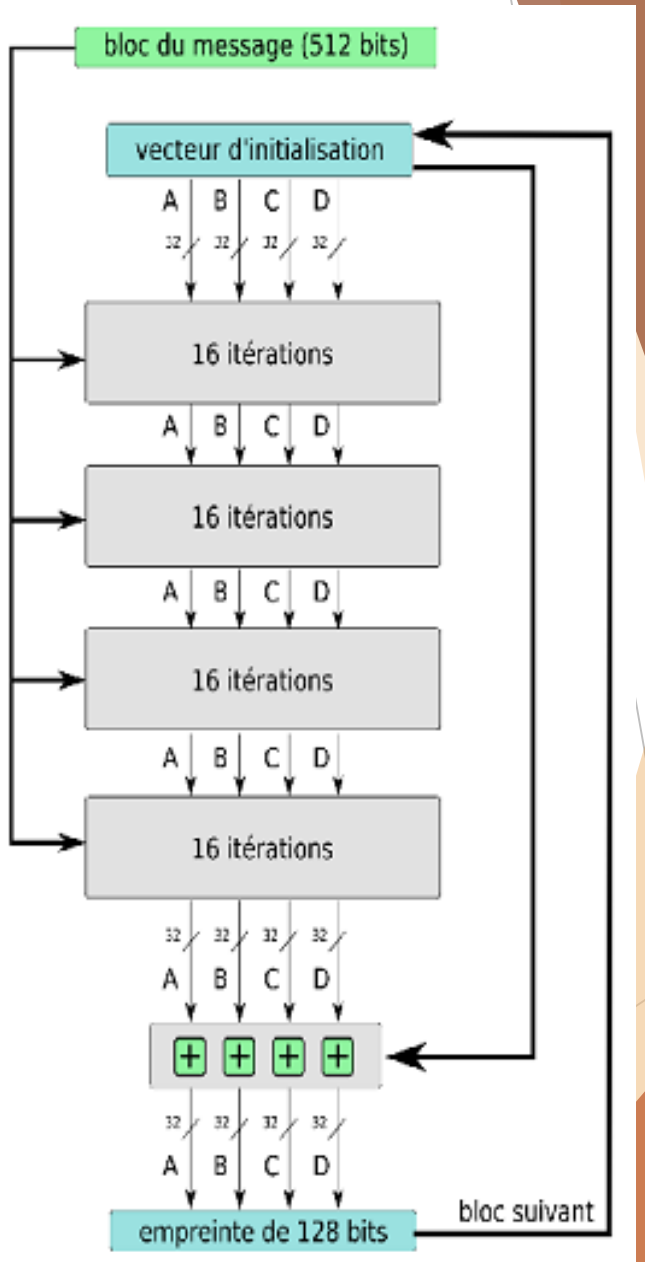
Fonction MD5 Algorithme

2. Calcul itératif:

- Traiter le message par blocs de 512 bits.
- 4 rondes de 16 opérations (64 itérations) pour bloc de 512 bits.
- Le vecteur d'initialisation (VI) de taille 128 bits
- VI divisé en 4 mots (A, B, C, D) de taille 32bits.
- Le résultat est utilisé pour initialiser les mots suivants.

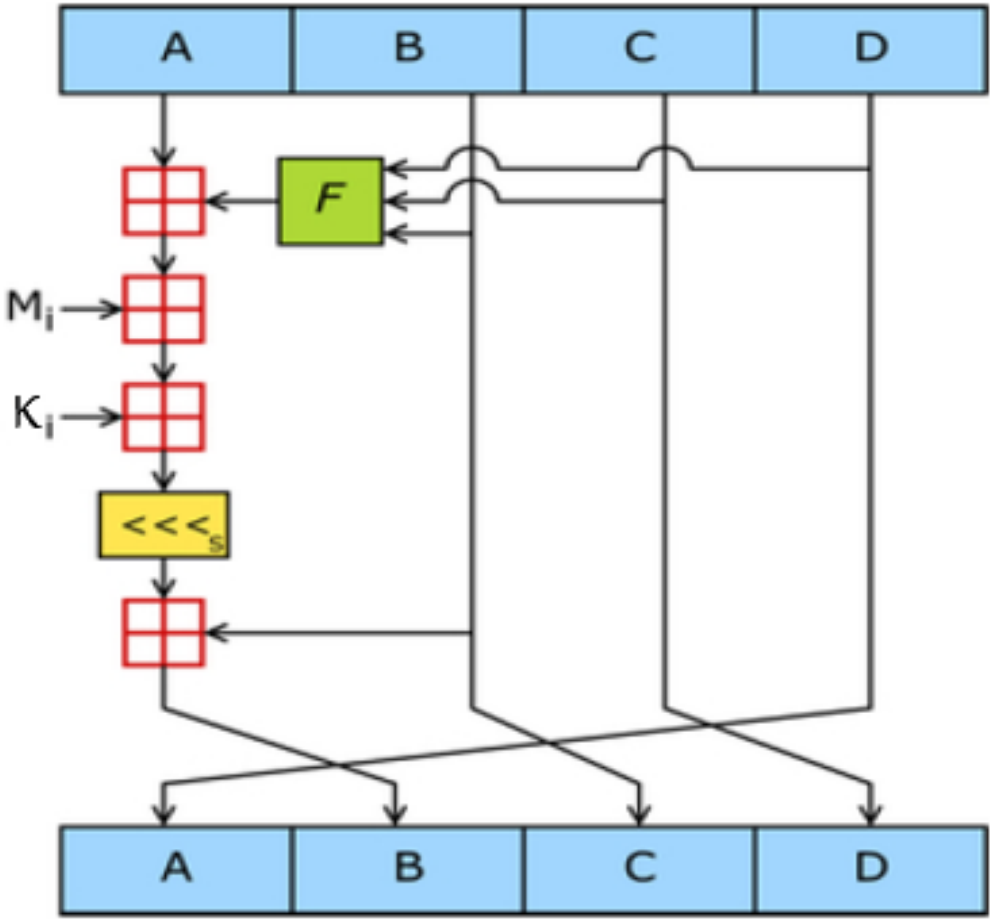
3. Résultat:

- Le résultat final est obtenu par la concaténation des A, B, C, et D.



Wikipedia

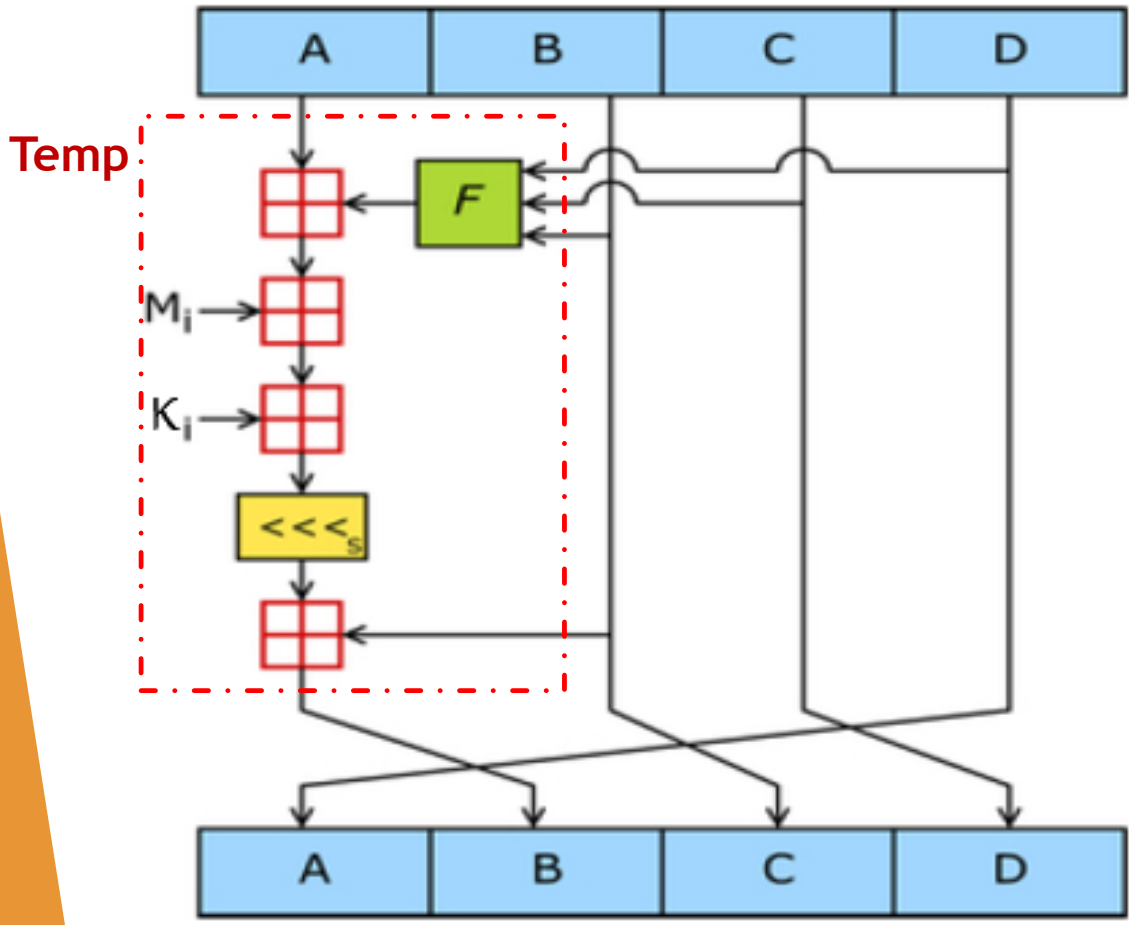
Fonction MD5
Algorithme: fonction de compression



Wikipedia

- F: fonction non linéaire, varie selon la ronde.
- Mi: sous- bloc de 32 bits.
- K_i :constante de 32 bits,
- $\lll_s(.)$: rotation de s bits vers la gauche.
- K_i *et* <<<$_s$ sont différentes pour chaque opération.

Fonction MD5
Algorithme: fonction de compression



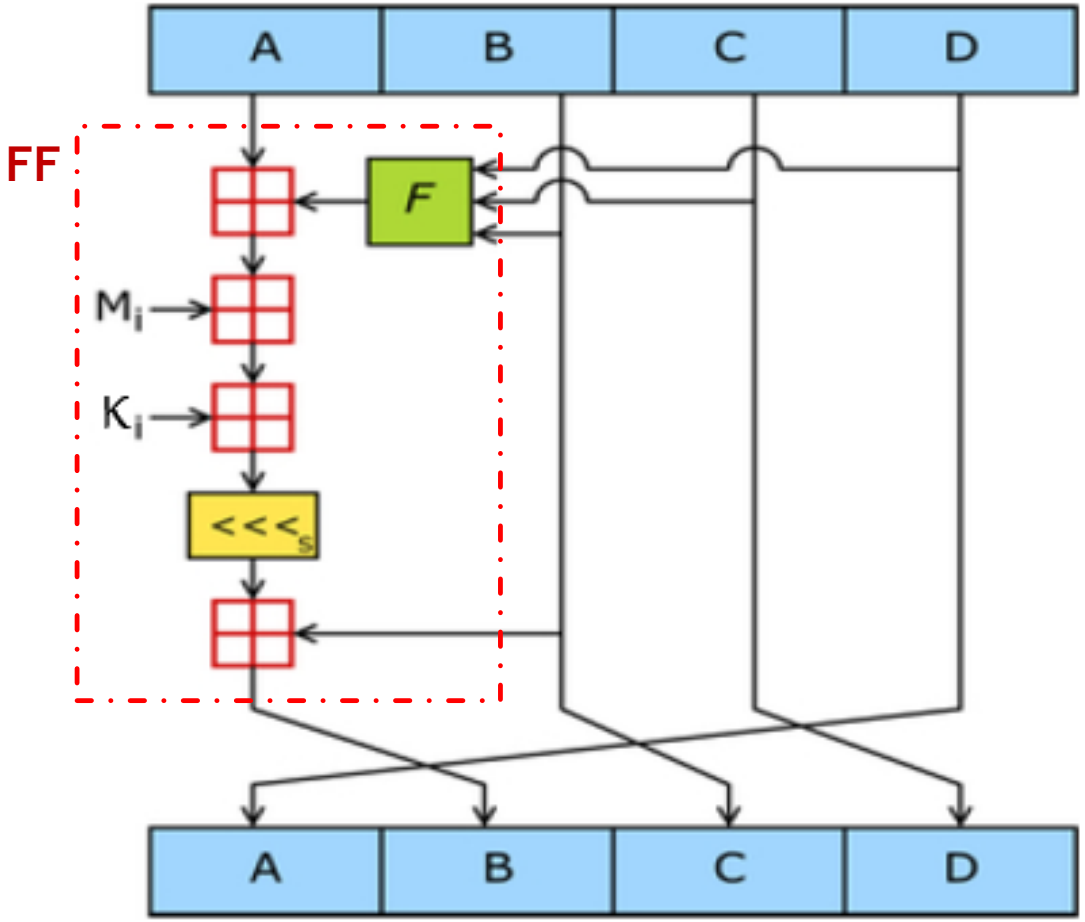
Fonction F selon la ronde:

- $F_1(B,C,D) = (B \wedge C) \vee (-B \wedge D)$
- $F_2(B,C,D) = (B \wedge D) \vee (C \wedge -D)$
- $F_3(B,C,D) = B \oplus C \oplus D$
- $F_4(B,C,D) = C \oplus (B \vee -D)$

Les nouvelles valeurs du VI:

- Temp= $\lll_s(M_i+F(B,C,D)+A+t_i) + B$
- A=D
- B= Temp
- C=B
- D=C

Fonction MD5
Exemple: exécution de la première ronde



(A,B,C,D)=(D, FF, B, C)

FF (a,b,c,d, M(0),7, D76AA478)
FF (d,a,b,c, M(1),12, E8C7B756)
FF (c,d,a,b, M(2),17, 242070DB)
FF (b,c,d,a, M(3),22, C1BDCEEE)
FF (a,b,c,d, M(4),7, F757C0FAF)
FF (d,a,b,c, M(5),12, 4787C62A)
FF (c,d,a,b, M(6),17, A8304613)
FF (b,c,d,a, M(7),22, FD469501)
FF (a,b,c,d, M(8),7, 698098D8)
FF (d,a,b,c, M(9),12, 8B44F7AF)
FF (c,d,a,b, M(10),17, FFFF5BB1)
FF (b,c,d,a, M(11),22, 895CD7BE)
FF (a,b,c,d, M(12),7, 6B901122)
FF (d,a,b,c, M(13),12, FD987193)
FF (c,d,a,b, M(14),17, A679438E)
FF (b,c,d,a, M(15),22, 49B40821)

Fonction MD5
Sécurité

- En 2004, X. Wang et al. ont permis de trouver des collisions sur MD5. ils ont détecté une attaque est complètement réussite.
- MD5 est déconseillée pour les applications de haute sécurité:
- MD5 reste utilisé dans certaines applications qui ne demande pas un niveau haut de sécurité.

Catégories des fonctions de hachage

1) Fonction de hachage sécuritaire sans clé

- MDC: *Message Digest Code* => Empreinte de message.
- Une fonction de hachage *H* qui peut être calculée sans connaissance d'un secret.
- Assure l'intégrité.
- La fonction est définie comme suit: H(m)

Catégories des fonctions de hachage

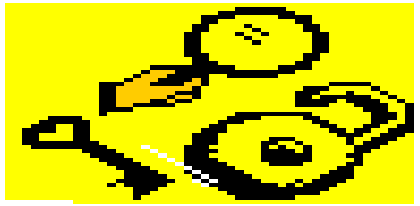
2) Fonction de hachage sécuritaire avec clé

- HMAC: *Keyed-Hash Message Authentication Code*
- Assure l'intégrité et l'authentification
- est un type de code d'authentification de message (MAC), calculé en utilisant une *fonction de hachage* en combinaison avec une *clé secrète*.
- La sécurité de HMAC est dépendante de la fonction de hachage utilisées.
- Exemple type: HMAC-SHA-1 ou HMAC-SHA-256

Catégories des fonctions de hachage

2) Fonction de hachage sécuritaire avec clé

- La fonction HMAC est définie comme suit: $HMAC_K(m) = H((K \oplus opad) \parallel H(K \oplus ipad) \parallel m))$
  - H : fonction de hachage sans clé
  - k : clé privée complétée avec des zéros
  - Ipad: prédéfinie par 0x363636...3636 (nombre de répétitions des octets dépendant la taille du bloc).
  - opad: prédéfinie par 0x5c5c5c...5c5c (nombre de répétitions des octets dépendant la taille du bloc).

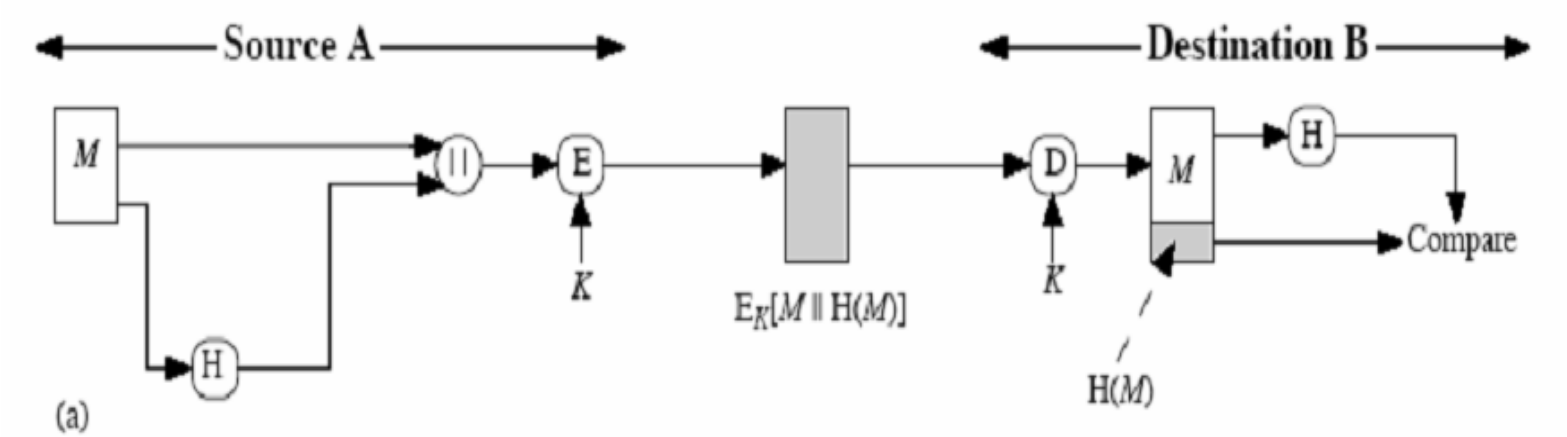


CrypTool

Indiv. Procedure → Hash

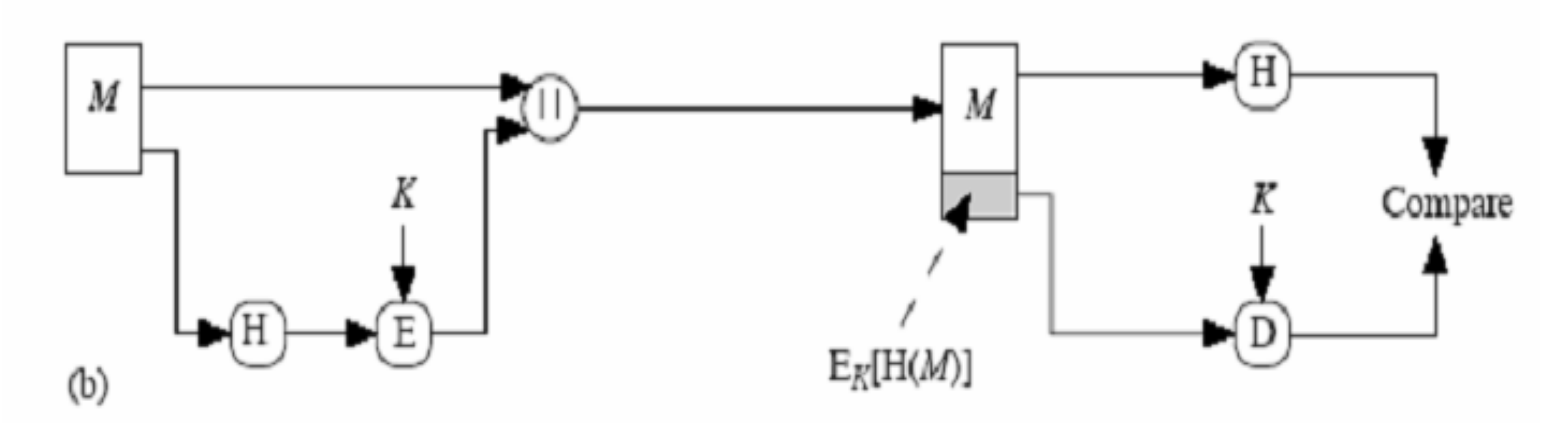
- Hash Demonstration
- Generation of HMAC

Exemple d'utilisations (1)



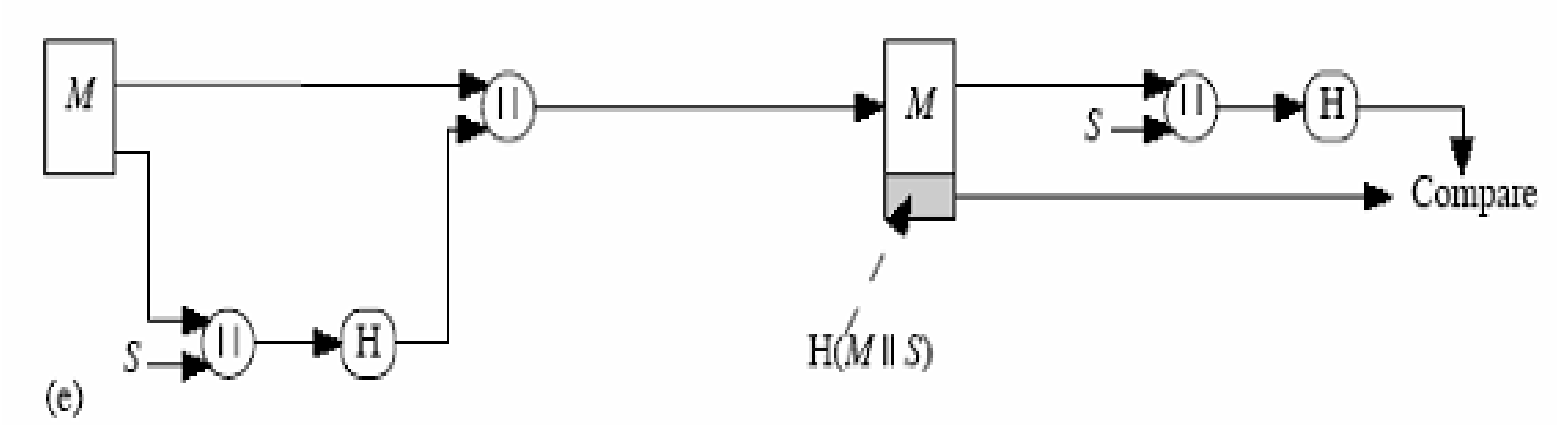
Le message concaténé à un hash code est chiffré en utilisant le chiffrement symétrique. Puisque seuls A et B partagent la clef secrète, le message doit provenir de A et n'a pas été modifié. Le haché fournit la structure exigée pour réaliser *l'authentification*. Puisque le chiffrement est appliqué au message entier et au code de hachage, la *confidentialité* est également fournie.

Exemples d'utilisations (2)



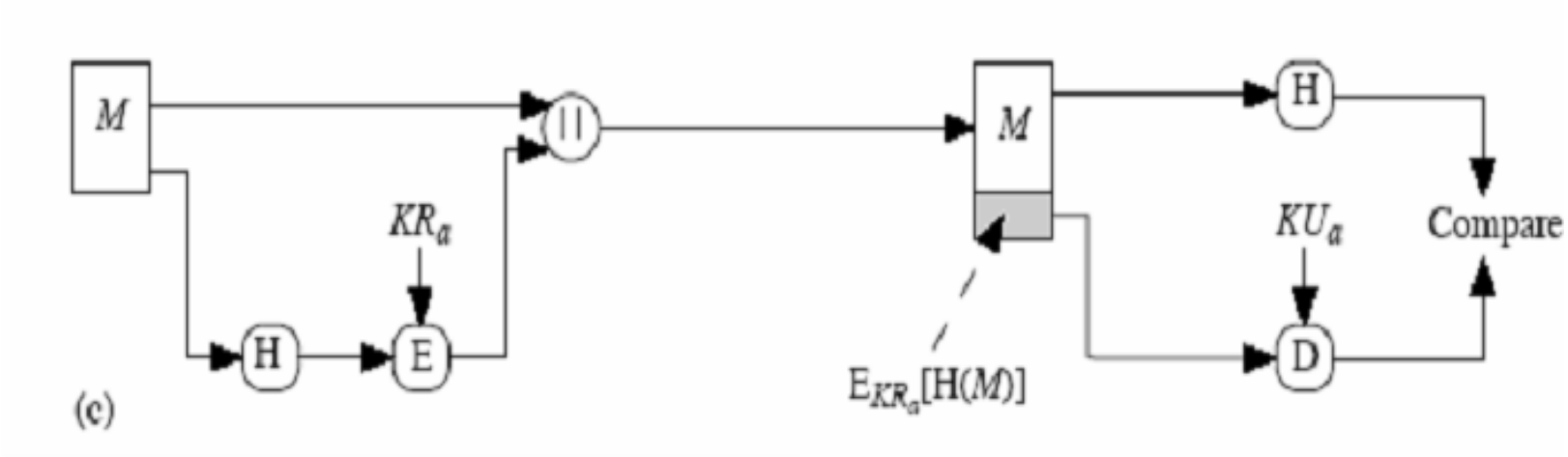
Seul le haché est chiffré, en utilisant le chiffrement symétrique. Ceci réduit le traitement pour les applications qui n'exigent pas la confidentialité. Ce point illustre en réalité le principe d'une fonction MAC (fonction de hachage + clé secrète).

Exemple d'utilisations (3)



- La technique suppose que les deux parties communicantes partagent une valeur secrète commune S.
- A calcule la valeur de hachage à la suite de la concaténation de M et de S et ajoute la valeur de hachage résultante à M.
- Puisque B possède S, il peut recalculer la valeur de hachage à vérifier. Puisque la valeur secrète elle-même n'est pas envoyée, un adversaire ne peut pas modifier un message arrêté et ne peut pas produire un message faux.

Exemple d'utilisations (4)



Seul le code de hachage est chiffré, en utilisant le chiffrement par clef publique et la clef privée de l'expéditeur. Ceci fournit l'authentification. Il fournit également une signature numérique, car seul l'expéditeur pourrait avoir produit le code de hachage chiffré.

Recommandations
Exemples des fonctions de hachage

Fonction de Hachage	Date	Bloc	n	Auteur	Sécurité
MD5	1992	512	128	RSA	- (2004)
RIPEMD	1992	512	128,160, …	H. Dobbertin et al.	- (2004)
SHA-1	1995	512	160	NIST	- (2017)
SHA-512 (SHA-2)	2001	1024	SHA-512	NIST	+
SHA-3	2015	1152, 1088, 832, ou 576	224, 256, 384 et 512	NIST	+

Recommandations

- NIST (2019)

Date	Niveau de Sécurité	Algorithme symétrique	Factorisation Module	Logarithme discret Clef	Logarithme discret Groupe	Courbe elliptique	Hash (A)	Hash (B)
Legacy [1]	80	2TDEA	1024	160	1024	160	SHA-1 [2]	
2019 - 2030	112	(3TDEA) [3] AES-128	2048	224	2048	224	SHA-224 SHA-512/224 SHA3-224	
2019 - 2030 et au-delà	128	AES-128	3072	256	3072	256	SHA-256 SHA-512/256 SHA3-256	SHA-1 KMAC128
2019 - 2030 et au-delà	192	AES-192	7680	384	7680	384	SHA-384 SHA3-384	SHA-224 SHA-512/224 SHA3-224
2019 - 2030 et au-delà	256	AES-256	15360	512	15360	512	SHA-512 SHA3-512	SHA-256 SHA-512/256 SHA-384 SHA-512 SHA3-256 SHA3-384 SHA3-512 KMAC256

Hash (A): Signatures digitales et autres utilisations nécessitant une résistance à la collision.
Hash (B): HMAC, KMAC, fonctions de dérivation de clef et génération de nombre aléatoire.

https://www.keylength.com/fr/4/

Références

- A Menezes, P Van Oorschot, S Vanstone, Handbook of applied cryptography, CRC Press Inc, 1997. http://cacr.uwaterloo.ca/hac/about/chap9.pdf (chapitre 9)
- G. Florin, S. Natkin, Cryptographie, 2007. (présentation) http://deptinfo.cnam.fr/Enseignement/CycleProbatoire/RSX112/cours_cryptographie.pdf