Authentification de messages

Authentification de Message

- L'authentification de message concerne :
  - La protection de l'intégrité d'un message
  - La validation de l'identité du créateur du message
  - La non répudiation de l'origine (résolution de conflit)
- 3 types de fonctions possibles :
  - chiffrement de message
  - fonction de hachage
  - code d'authentification de message (MAC – Message authentication code)

Dangers à contrer

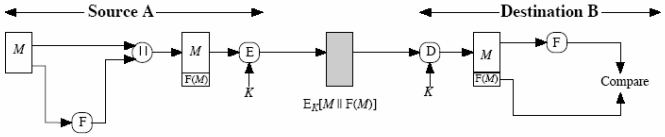
- révélation
- analyse de trafic
- mascarade
- modification du contenu
- modification de séquence
- modification de la synchronisation
- répudiation de la source
- répudiation de la destination

Chiffrement de message pour l'authentification

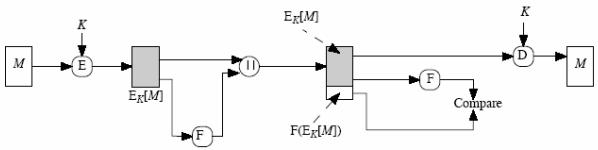
Chiffrement de Message

- Le chiffrement de message en lui-même fournit également une mesure de l'authentification
- Si un chiffrement symétrique est employé alors :
  - Le récepteur sait que seul l'expéditeur peut l'avoir créé
    - puisque seuls l'expéditeur et le récepteur connaissent la clé utilisée
  - Il faut s'assurer que le contenu ne peut pas être changé
    - si le message a la structure appropriée, on peut utiliser la redondance ou une somme pour détecter un changement
    - Contrôle d'erreur interne et externe

Contrôle d'erreur interne et externe



(a) Internal error control



(b) External error control

Chiffrement de Message

- Si un chiffrement à clé publique est employé :
  - Aucune information sur l'expéditeur
    - puisque n'importe qui peut potentiellement utiliser la clé publique
  - Cependant si
    - l'expéditeur signe le message en utilisant sa clé privée
    - chiffre ensuite avec la clé publique du destinataire
    - On a le secret et l'authentification
  - Possibilité d'identifier les messages corrompus
  - Mais au coût de deux utilisations de ce chiffrement sur le message

Fonction de hachage

Fonction de hachage

- Objectif en cryptographie :
  - Fournir un condensé de taille fixe
  - Détecter des changements dans le message
  - Représenter des données de façon certaine tout en réduisant la taille utile qui sera réellement chiffrée
- Intérêt :
  - Permet l'usage de la cryptographie asymétrique sans engendrer trop de ralentissement
  - Permet d'assurer la provenance d'un fichier et son intégrité
  - Permet d'authentifier des données, des clés, …
  - Peut être employé de diverses manières avec le message
  - Utilisée le plus souvent pour créer une signature numérique
- Algorithmes publiques

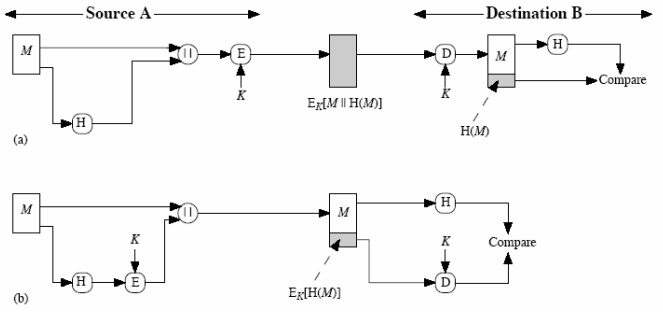
Propriétés

- Haché : caractéristique d'un texte ou de données uniques
- Différentes données donneront toutes des condensés différents
- Le haché ne contient pas assez d'informations en lui-même pour permettre la reconstitution du texte original
- But : être représentatif d'une donnée particulière et bien définie

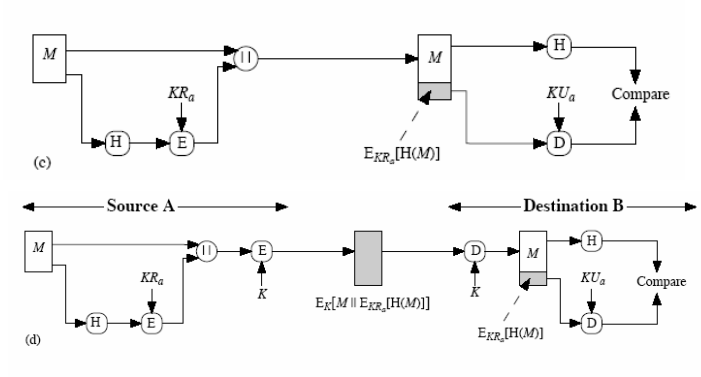
Attentes relatives aux hachages

- Peut s'appliquer à n'importe quelle longueur de message M
- Produit un résultat de longueur constante h
- Il doit être facile de calculer h = H(M) pour n'importe quel message M
- Pour h donné, il est impossible de trouver x tq. H(x) = h
  - propriété à sens unique
- Pour x donné, il est impossible de trouver y tq. H(y) = H(x)
  - résistance faible de collision
- Il est impossible de trouver x,y tq H(y) = H(x)
  - résistance forte de collision

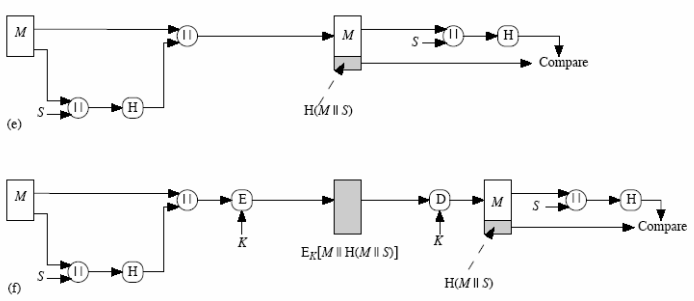
Exemples d'utilisation (1)



Exemples d'utilisation (2)



Exemples d'utilisation (3)



Utilisation des fonctions de hachage

$A \rightarrow B: E_K[M \| H(M)]$ •Provides confidentiality —Only A and B share K •Provides authentication —H(M) is cryptographically protected (a) Encrypt message plus hash code	$A \rightarrow B: E_K[M \| E_{KR_a}[H(M)]]$ •Provides authentication and digital signature •Provides confidentiality —Only A and B share K (d) Encrypt result of (c) - shared secret key
$A \rightarrow B: M \| E_K[H(M)]$ •Provides authentication —H(M) is cryptographically protected (b) Encrypt hash code - shared secret key	$A \rightarrow B: M \| H(M \| S)$ •Provides authentication —Only A and B share S (e) Compute hash code of message plus secret value
$A \rightarrow B: M \| E_{KR_a}[H(M)]$ •Provides authentication and digital signature —H(M) is cryptographically protected —Only A could create $E_{KR_a}[H(M)]$ (c) Encrypt hash code - sender's private key	$A \rightarrow B: E_K[M \| H(M) \| S]$ •Provides authentication —Only A and B share S •Provides confidentiality —Only A and B share K (f) Encrypt result of (e)

Birthday Attacks

- On pourrait penser que des hash de 64- bit sont sûrs
- mais vu le paradoxe des anniversaires ce n'est pas vrai
- attaque d'anniversaire :
  - l'adversaire produit de $2^{m/2}$ variations d'un message valide tous avec essentiellement la même signification
  - l'adversaire produit également de $2^{m/2}$ variations du message frauduleux désiré
  - les deux ensembles de messages sont comparés pour trouver une paire donnant les mêmes hash (probabilité > 0.5 par paradoxe des anniversaires)
  - faire signer à l'utilisateur le message valide, puis substituer le contrefaçon qui aura une signature valide
- Conclusion : nécessité d'utiliser de plus grands Hash

MAC

Message Authentication Code (MAC)

- Algorithme qui crée un petit bloc (authentificateur) de taille fixée
  - Fonction du message initial et d'une clef
  - Réversibilité inutile
- Apposé au message comme une signature mais ce n'est pas une signature
- Vérification : le récepteur exécute le même calcul sur le message et le compare avec le MAC reçu

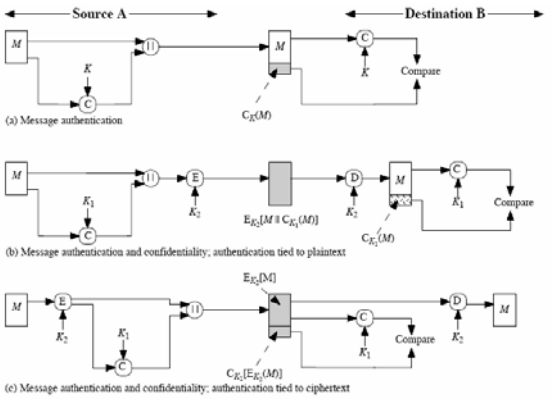
Utilisation

- fournit l'assurance que le message est inchangé (intégrité) et vient de l'expéditeur (authentification)
- peut également être employé comme un chiffrement supplémentaire (rare)
  - employer généralement les clefs séparées pour les 2 chiffrement
  - peut être calculé avant ou après le chiffrement principal
  - il est généralement conseillé de le faire avant

Message Authentication Codes

- Conditions d'utilisation
  1. Connaissant un message et un MAC, il devrait être impossible de trouver un autre message avec le même MAC
  2. Les MACs devraient être uniformément distribués
  3. Le MAC devrait dépendre également de tout les bits du message
- Intérêt d'un MAC ?
  - Quand seule l'authentification est nécessaire
  - Quand nécessité d'une authentification persistant plus longtemps que le chiffrage (par exemple dans le cadre d'archives)

Message Authentication Code



Utilisation du MAC

$A \rightarrow B: M \| C_K(M)$ •Provides authentication —Only A and B share K (a) Message authentication
$A \rightarrow B: E_{K_2}[M \| C_{K_1}(M)]$ •Provides authentication —Only A and B share K_1 •Provides confidentiality —Only A and B share K_2 (b) Message authentication and confidentiality: authentication tied to plaintext
$A \rightarrow B: E_{K_2}[M] \| C_{K_1}(E_{K_2}[M])$ •Provides authentication —Using K_1 •Provides confidentiality —Using K_2 (c) Message authentication and confidentiality: authentication tied to ciphertext

Utilisation de chiffrement symétriques

- On peut employer n'importe quel mode chaîné - et le bloc final étant le MAC
- **L'algorithme d'authentification de données** (DAA) est un MAC largement répandu basé sur DES-CBC
  - utilisation d'IV=0 et un padding de 0 pour le bloc final
  - chiffrer le message en utilisant le DES en mode de CBC
  - et envoyer le bloc final comme MAC
    - ou les M bits extrême gauche ($16 \leq M \leq 64$) du bloc final
- Mais le MAC résultant est trop petit pour la sécurité

Sécurité

Sécurité des fonctions de hachage et MACs

- Attaque par force brute
  - La recherche de collision (coûte $2^{m/2)}$
    - Les hash 128 bits semblent vulnérables $\rightarrow$ 160 bits
  - Attaque sur des paires connues message/MAC
    - On peut attaquer l'espace de clé ou le MAC
    - Utiliser au minimum un hash de 128 bit
- Attaques cryptanalytiques
  - Souhait : que la force brute soit la meilleure alternative
  - Se focalisent sur des collisions dans la fonction f
  - Exploitent les propriétés des rondes dans les algorithmes

Questions

- Chiffrements de message
- Fonction de hachage
- MAC

Références

- [stallings] – ch 11
- [schneier] – ch2, ch18
- [stinson] (2e ed) – ch 7
- http://www.securiteinfo.com/crypto/hash.shtml
- http://home.ecn.ab.ca/~jsavard/crypto/mi0605.htm
- http://www.rsasecurity.com/rsalabs/faq/
- http://www.tcs.hut.fi/~helger/crypto/link/hash/
- http://www.acm.jhu.edu/~upe/member_sites/zarfoss/HMAC.html