Protocole de sécurité

Bouzara Reguia Lamia

Université de Médéa

Décembre 2023

Protocole de sécurité

Le but de la cryptographie est la protection de l'information en respectant la confidentialité, la non-répudiation, l'authenticité et l'intégrité de l'information.

Confidentialité:

Le chiffrement du message assure la confidentialité (chiffrement symétrique ou asymétrique).

Chiffrement Symétrique

Chiffrement Asymétrique

Intégrité:

Avoir l'intégrité veut dire que le document n'a pas été altéré entre l'envoie et la réception.

Pour assurer l'intégrité on utilise des fonctions appelées "fonctions de Hachage".

Fonction de Hachage:

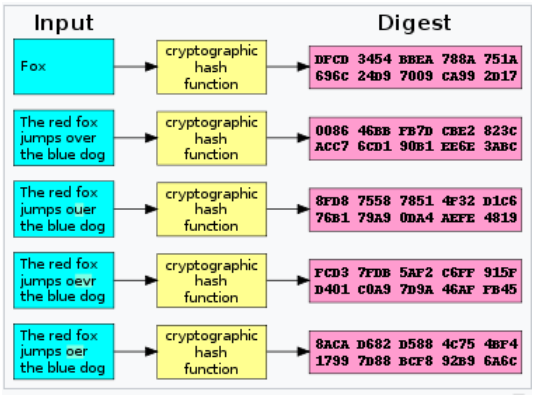
Le but d'une fonction de hachage est d'obtenir une empreinte du document, elle est définie comme suit:

Définition .1. *Une fonction de hachage $H(m)$ est une fonction mathématique qui associe à une chaîne binaire d'une longueur variable une chaîne binaire d'une longueur fixe.*
Une fonction de hachage doit vérifier les propriétés suivantes:

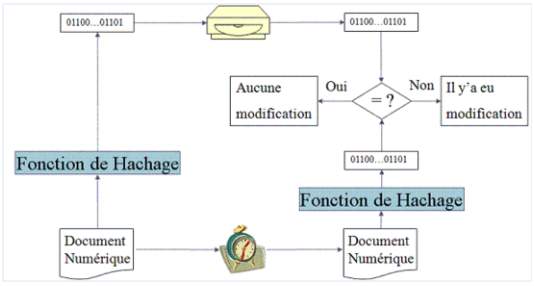
- *Un message doit avoir toujours la même valeur de hachage cela veut dire que la fonction doit être déterministe.*
- *$H(m)$ doit être facile à calculer.*
- *la variable peut être d'une taille arbitraire et la sortie doit être d'une taille fixe.*
- *Il doit être impossible de trouver deux message ayant la même valeur de hachage (résistance à la seconde pré-image).*
- *$H(m)$ doit être à sens unique d'une façon qu'il soit impossible de trouver m à partir de $H(m)$.*
- *résistance aux collision:*

$$m_1 \neq m_2 \Rightarrow H(m_1) \neq H(m_2)$$

Exemple .1. *L'une des fonctions de hachage les plus connues est la fonction SHA-1*



Pour vérifier l'intégrité on suit le schéma suivant:



le récepteur calcule le haché du message reçu et il le compare avec le haché du message envoyé par l'émetteur. Si les deux valeurs sont égaux alors le document n'a pas été modifié.

Authentification de l'origine des données:

On doit vérifier la provenance ou la source du message. L'authentification de l'origine doit permettre au lecteur d'un document d'identifier la personne ou l'algorithme qui a effectué l'action.

Cela veut dire qu'on doit vérifier l'authenticité à plusieurs niveau:

Authenticité au niveau des communiquants:

Pour garantir l'authentification d'entité le système d'authentification peut utiliser un protocole de type défi/réponse, ce protocole procède comme suit:

Le vérificateur envoie un challenge (défi) en montrant à la personne qui veut prouver son identité (le prouveur) qu'il a un secret à déclarer et ce dernier doit prouver son identité par une réponse correcte au défi.

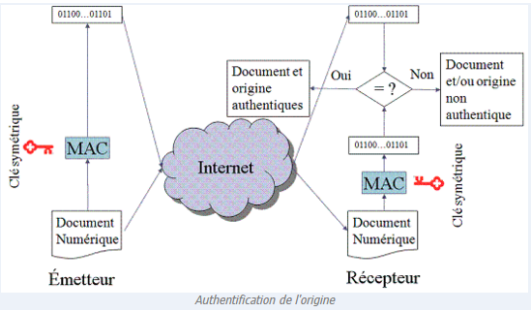
On peut faire le défi par plusieurs façon, par exemple:

- Par une nonce: nombre pseudo aléatoire non prévisible.
- Par un numéro de séquence.
- Par une combinaison de nonce avec numéro de séquence.

Authenticité au niveau du message:

On peut vérifier l'authenticité du message et leur intégrité en utilisant une famille H_k associée a une clé k qui vérifie:

- Le $H_k(m)$ doit être facile à calculer.
- Le calcule de $(m, H_k(m))$ pour m quelconque doit être difficile. On apelle ce mécanisme un MAC (code de hachage).
  Le schéma suivant montre ses étapes:



  - Le récepteur et l'émetteur utilisent une clé symétrique pour assurer l'authenticité de la source.
  - L'émetteur calcule le code de hachage MAC; $H(clésymétrique \mid m)$ du message qu'il va envoyé.
  - Le récepteur calcule de la même manière le MAC du message reçus et il compare avec le MAC envoyé par l'émetteur.

– Si les deux valeurs sont égaux on dit que le message et l'origine sont authentique.

Non-répudiation:

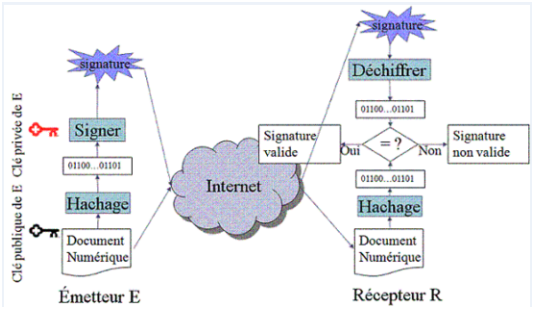
Un crypto-système doit assurer la non-répudiation cela veut dire que l'émetteur ne peut pas attribuer l'envoie du message à une autre personne. Pour garantir la non-répudiation on utilise la signature digitale.

0.0.1 La signature digitale:

La signature à pour but de lier un message à son auteur. Elle fonctionne d'une manière inverse au chiffrement à clé publique, cela veut dire au lieu de chiffrer un document avec la clé publique on chiffre avec la clé secrète le résultat est appelé la signature du document.
Pour vérifier la signature il suffit d'utiliser la clé publique.

Signer un document:

Pour signer un document on utilise la cryptographie asymétrique et symétrique à la fois. Supposons que Alice veut envoyer un document signer à Bob, elle doit suivre les étapes suivantes:



- Alice utilise une fonction de hachage pour générer l'empreinte du document.

- Puis elle utilise la clé privée pour signer le code de hachage. Elle obtient la signature de son document.

- Alice envoie le document avec la signature du document à Bob.

- Bob reçoit le document avec la signature et il recalcule le code du hachage du document et déchiffre la signature avec la clé publique et il compare avec le code de hachage déchiffré avec la clé publique envoyé par Alice et le code de hachage qu'il a obtenu en chiffrant le document.

Signature RSA

Soit m un entier qui peut être un court document pour signer. La signature du document m est donnée par:

$$s \equiv m^d[n]$$

Si on veut vérifier la signature s:
On utilise la clé publique (n, e) et on calcule:

$$s^e \equiv (m^d)^e[n] \equiv m^{de}[n] \equiv m[n].$$

Exemple .2. *Alice choisit* $p = 11$, $q = 23$ *et* $e = 3$.

- *Clé publique:* $(253, 3)$

- *Clé privée:* $(220, 147)$

Alice veut faire un virement de 229 *DA.*
Quelle est sa signature?
Comment la banque peut savoir quelle est la somme qu'Alice veut virer?

Solution:
$s = 229^{147}[253]$ *on a* $147 = 2^7 + 2^4 + 2 + 1$ *en utilisant l'écriture en mode binaire de* 147 *on calcule* $229^{147}[253]$

$$\begin{aligned}
229^2 &\equiv 70[253] \\
229^{2^2} &\equiv 93[253] \\
229^{2^3} &\equiv 47[253] \\
229^{2^4} &\equiv 185[253] \\
229^{2^5} &\equiv 70[253] \\
229^{2^6} &\equiv 93[253] \\
229^{2^7} &\equiv 47[253]
\end{aligned}$$

alors le résultat est donné par:

$$\begin{aligned}
229^{147} &\equiv 47 \cdot 185 \cdot 70 \cdot 229 \\
&\equiv 114[253]
\end{aligned}$$

Si on veut connaitre la somme on fait:

$$s^e \equiv m[n].$$

Bibliography

[1] https://moodle.utc.fr/file.php/498/SupportIntroSecu/co/CoursSecurite$_1$3.*html*