Chapitre 4 : La cryptographie à clés publiques

9 juin 2023

Sommaire

Introduction

Problématique

Un défaut des systèmes à clef privée est qu'ils nécessitent la communication préalable de la clef de chiffrement e_K entre les entités communiquantes. Cet échange doit se faire nécessairement par un canal sûr. De plus

- La durée de vie des clés est assez courte.
- On a le problème de la non-répudiation.
- La distribution des clés devient trés compliquée si on a plusieurs entités communicantes.

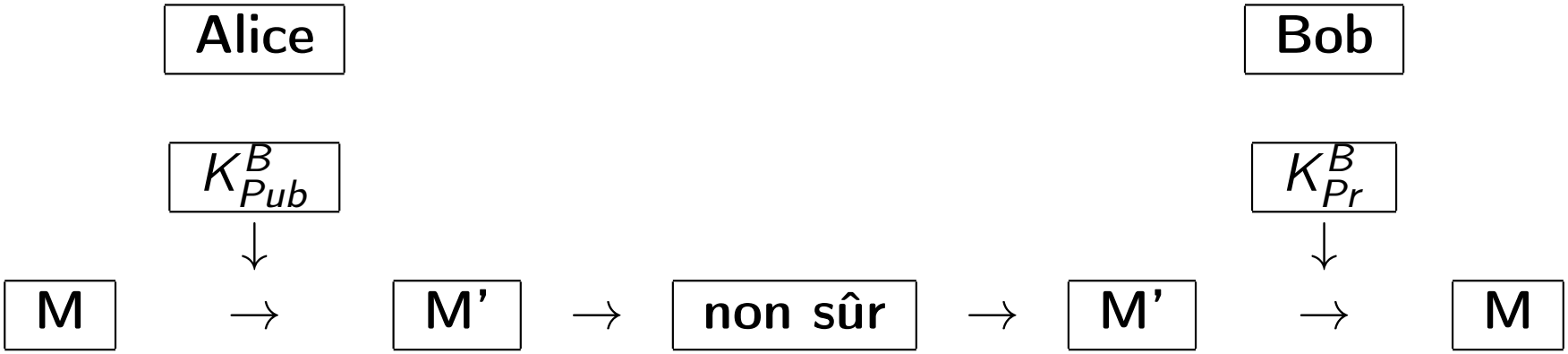
Solutions

Une solution aux problèmes confrontés par les schémas de chiffrement symétrique est de rendre la règle de déchiffrement d_K impossible à retrouver à partir de la règle de chiffrement e_K. Ainsi

- e_K peut être publiée dans un répertoire
- Alice (ou toute autre personne) peut envoyer un message à Bob chiffré par e_K sans communication privée préalable.

D'où le nom de système à clef publique. On rencontre aussi les termes cryptographie non symétrique ou asymétrique.

Schémas d'un système à clé publique



Historique

L'idée de système à clef publique date de 1976. Elle est due à Whitfield Diffie et Martin Hellman.

La cryptographie et les problèmes calculatoires

Les système à clef publique reposent sur divers problémes calculatoires. Les plus connus sont les suivants :

1. La difficulté de la factorisation des grands entiers.
2. La difficulté de résoudre le subset sum problème ou sac-à-dos.
3. Le probléme du décodage d'un code linéaire.
4. La difficulté de calculer le logarithme discret dans un corps fini.
5. La difficulté de trouver un plus court vecteur dans un réseau.

Remarque

- Un système à clef publique ne peut jamais être inconditionnellement sûr. En effet un opposant peut chiffrer chaque texte clair possible x avec la règle de chiffrement e_K publique jusqu'à ce qu'il trouve l'unique x tel que $y = e_K(x)$.

- Il est conceptuellement utile de voir le système à clef publique comme une fonction à sens unique à trappe.

- Si l'on construit un système cryptographique à clef publique, on ne souhaite pas que la fonction e_K soit à sens unique tout court. Il est nécessaire que le destinataire (Bob) connaisse une trappe cachée, qui est en fait une information secrete qui permet l'inversion facile de e_K.

Les fonctions à sens unique jouent un rôle central en cryptographie.

Fonction à sens unique avec trappe

Définition

Une fonction f est à sens unique si connaissant x, $f(x)$ est facile à calculer, mais connaissant $f(x)$, il est pratiquement impossible de calculer x, c'est-à-dire de résoudre l'équation $f(x) = y$. Par pratiquement impossible, on entend que le temps de calcul est de l'ordre de quelques siècles, même sur des ordinateurs assez puissants.

Exercice :

Donner un exemple de fonction qui n'est pas à sens unique.

Notion de trappe

Une fonction à sens unique

$$f : \mathcal{P} \to \mathcal{C}$$

est dite avec trappe s'il existe une information (appelée trappe) qui permet "facilement" , pour tout $y \in \mathcal{C}$, de determiner $x \in \mathcal{P}$ tel que $y = f(x)$.

Définition

La trappe est considérée comme une fonction $g : \mathcal{C} \to \mathcal{P}$ inverse à gauche de f, c'est dire $g \circ f = Id_{\mathcal{P}}$.

Pour fabriquer un cryptosystème à clé publique, on prend f comme la clé publique et g la clé privée. Ainsi la trappe représente un secret détenu uniquement par le destinataire du message.

Exemples de fonctions à sens unique

Ces exemples reposent sur des problèmes considérés comme difficiles (calculatoirement).

1. Soit p un nombre premier et g un générateur de $(\frac{\mathbb{Z}}{p\mathbb{Z}})^*$ alors

$$f_1 : (\frac{\mathbb{Z}}{(p-1)\mathbb{Z}}, +) \to (\frac{\mathbb{Z}}{p\mathbb{Z}}, \times) : x \mapsto g^x \pmod{p}$$

   est une fonction à sens unique pour certaines conditions sur p.

2. Soient p et q deux nombres premiers et $n = pq$ alors

$$f_2 : \frac{\mathbb{Z}}{n\mathbb{Z}} \to \frac{\mathbb{Z}}{n\mathbb{Z}} : x \mapsto x^2 \pmod{n}$$

   est une fonction à sens unique pour certaines conditions sur p et q.

Application : stockage du mot de passe.

De nombreux logiciels demandent à l'utilisateur un « code d'accès ». Ce code étant inscrit sur le disque dur, n'importe quel utilisateur un peu averti peut le lire ! Une fonction à sens unique résout le problème.

Solution

Le logiciel contient la fonction f (Algorithme public). Si x est le « code d'accès », on écrit f(x) sur le disque dur. Un utilisateur averti aura accès à f(x), mais ne pourra jamais trouver x.

Exemple pratique 1

La fonction f est : $f(p, q) = p * q$. Chaque utilisateur choisit deux « grands » nombres premiers comme code d'accès et écrit $p * q$ sur le disque dur.

Exemple pratique 2

La fonction qui à un sous ensemble J de $S = \{a_1, a_2, \cdots, a_n\}$ associe $t = \sum_{a_i \in J} a_i$ est une fonction à sens unique. Il s'agit du problème de sac à dos.

Autre solution pour le disque

Pour le problème du code d'accès, S et t sont écrits sur le disque dur. L'utilisateur choisit les numéros des éléments de S comme code d'accès. En pratique, il saisira un nombre N comme code d'accès. Ce nombre sera converti en binaire par l'ordinateur. Seuls les a_i correspondants aux 1 de l'écriture binaire de N seront additionnés.

Exercice

On pose $S = \{14, 28, 56, 82, 90, 132, 197, 284, 341, 455\}$. Donner vous un code d'accés N et fournir les données qui seront inscrites sur le disque dur. Donner une méthode pour travailler avec des codes d'accés comportant des lettres.

Les protocoles cryptographiques

De façon générale, un protocole est une série finie d'étapes conçue pour accomplir une tâche avec au moins deux partenaires.

Définition

Un protocole cryptographique est une suite de règles déterminant l'ensemble des opérations cryptographiques nécessaires et leur séquence pour sécuriser une communication (une transaction, un échange de données, ..) entre plusieurs entités.

Un protocole cryptographique doit permettre à des personnes qui ne se font pas confiance en général d'échanger en toute sécurité et en présence de personnes malveillantes. Il doit donc empêcher l'espionnage et la tricherie sous toutes leurs formes.

Un protocole de cryptographie doit avoir un objectif de securité bien précis : échange de clés, preuve de connaissance, identification, etc.

Echange de clé Diffie-Hellman

Pour que Alice et Bob aient une clé partagée, ils s'entendent d'abord sur un groupe **G** (noté multiplicativement) puis effectuent les étapes suivantes.

1. Alice et bob choisissent un élément g d'ordre premier p suffisamment grand dans **G** ;
2. Alice choisit un paramétre secret (valeur aléatoire $a < p$) et calcule g^a dans **G** et transmet publiquement g^a à Bob ;
3. Bob choisit un parametre secret (valeur aléatoire $b < p$) et calcule g^b dans G et transmet publiquement g^b à Bob ;
4. Chacun d'eux cacule la valeur commune $k = (g^b)^a = (g^a)^b$ qui constituera leur clé privée ou comme outil pour fabriquer la clé privée.

Echange de clé Diffie-Hellman

Remarque

L'espion Charlie qui suit la communication connait les paramétres publiques **G**, le sous groupe H d'ordre p et de générateur g. Il connait aussi g^a et g^b. Cependant il ne doit pas pouvoir calculer $k = (g^b)^a = (g^a)^b$. Autrement dit il ne doit pas recupérer a ou b connaissant g^a et g^b. Ce problème est basé sur le logarithme discret.

La signature numérique

Signature numérique

Première définition

Une signature (digitale-manuelle ou numérique-cryptographique) est un procédé, qui, appliqué à un message, garantit la non répudiation par le signataire et donc réalise les deux objectifs suivants

- identification unique du signataire,
- et preuve d'accord sur le contenu du document.

Une signature numérique doit posséder les propriétés suivantes :

1. Unique,
2. Impossible à usurper,
3. Impossible à répudier par son auteur,
4. Facile à vérifier par un tiers,
5. Facile à générer.

Modélisation des systèmes de signatures numériques

Un système de signature est composé d'un quintuplet $(\mathcal{P}, \mathcal{S}, S_{k'}, V_k, \mathcal{K})$:

- $\mathcal{P}$ est un ensemble appellé espace des textes clairs ;
- $\mathcal{S}$ est un ensemble appellé espace des signatures ;
- $S_{k'} : \mathcal{P} \to \mathcal{S}$ est une fonction injective dite fonction de signature (non nécessairement bijective) qui dépend d'un parametre k' appelé clé privée.
- $V_k : \mathcal{P}x\mathcal{S} \to \{vraie, faux\}$ est la fonction de vérification de signature binaire telle que $V_k(m, s) = \text{vraie}$ si et seulement si $S_{k'}(m) = s$ (dépendant de la clé publique k) .
- $\mathcal{K}$ l'ensemble des paramétres utilisés est l'espace des clés.

Comparaison des signatures numériques et manuelles

Différences entre les deux types de signature

Signature manuelle

1. Associé physiquement au document signé ;
2. Identique pour tous les documents venant d'un même signataire ;
3. Habituellement à la dernière page.

Signature numérique

1. Peut etre stockée et envoyée indépendemment du document signé ;
2. Fonction du document même si on signe avec la même clé privée ;
3. Couvre l'ensemble du document (dépend de tout le message).

Exercice : Donner une méthode permettant de concevoir un algorithme à signature numérique.

Exemple de signatures numériques

Une façon simple d'avoir un algorithme de signature est d'utiliser un algorithme de chiffrement à clé publique bijective. On rappelle que pour un chiffrement à clé publique on a :

- $C_k : \mathcal{P} \to \mathcal{C}$ fonction de chiffrement
- $D_{k'} : \mathcal{C} \to \mathcal{P}$ fonction de déchiffrement telle que $D_{k'} \circ C_k(m) = m$.

On peut le transformer en algo de signature si C_k est aussi l'inverse à gauche de $D_{k'}$ et donc $D_{k'} = C_k^{-1}$.

Astuce

Dans ce cas, on prend l'algorithme de "déchiffrement" comme étant l'agorithme de signature. Pour signer m on calcule $D_{k'}(m) = s$ et pour vérifier on calcule $C_k(s)$ et on compare avec m.

Les fonctions de hachage

Définition

Une fonction de hachage est une fonction publique

$$h : \{0,1\}^* \rightarrow \{0,1\}^n$$

telle que :

a) h transforme un message (binaire) de longueur quelconque en un message de longueur fixe ; (Fonction de Compression)

b) pour tout x, $h(x)$ est facile à calculer ; (Facilement calculable)

Les images $h(x)$ sont appelés ***haches*** ou ***empreintes***.

La fonction h en tant que telle n'est forcément pas utilisable pour des services de sécurité.

Fonction de hachage cryptographique

On dira qu'une fonction de hachage est une fonction pour la cryptographie si elle est

1. Á sens unique sans trappe : pour presque tout y, il est difficile de trouver x tel que $h(x) = y$;
2. Faiblement résistante au collisions : pour presque tout x, il est difficile de trouver x' tel que $h(x) = h(x')$;
3. Fortement résistante au collisions : il est difficile de trouver un couple (x, x') tel que $h(x) = h(x')$.

Usage des fonctions de hachages

Les fonctions de hachage sont notamment utilisées pour :

- l'intégrité et
- la modélisation théorique des fonctions à sens unique.

Intégrité

Si x est un message alors pour garantir l'intégrité de x on envoie ou on stocke le couple $(x, h(x))$ où $h(x)$ est l'empreinte de x via une fonction de hachage h. Le message est considéré intégre s'il est bien accompagné par son empreinte et lui correspond.

On fait souvent signer l'empreinte d'un document plutôt que le document lui-même.

Exemples standards

Parmi les algorithmes standards de hachages utilisés par des logiciels cryptographiques nous pouvons citer **MD** (Message Digest), **SHA** (Secure Hash Algorithm), **SHS** (Secure Hash Standard) avec **leurs différentes versions** :

- MD4, Rivest, 1990
- MD5, Rivest, 1992, empreinte sur 128 bits, RFC 1321
- SHA, NIST-1993, FIPS 180, SHA1, empreinte sur 160 bits,
- SHS, 2001, FIPS 180-2 inclut SHA1 et SHA2 (SHA-256, SHA-384, SHA-512.