Crypto, Sécurité, Réseaux

TD 3

La cryptographie à clé publique

La cryptographie à clé publique se base sur des structures algébriques spéciales et sur la difficulté de quelques problèmes calculatoires – la factorisation de larges nombre et trouver des logarithmes discrètes dans des groupes cycliques d'ordre premier.

La factorisation et le chiffrement RSA (Rivest-Shamir-Adleman)

Prenons deux larges nombres premiers p, q. Mettons $N = pq$ (cette valeur s'appelle un module RSA). Nous avons besoin d'une valeur spéciale, qui compte le nombre de numéros entre 1 et N qui sont co-premiers avec N : $\varphi(N) = (p-1)(q-1)$. Les valeurs $p, q, \varphi(N)$ restent privées, tandis que N fait partie de la clé publique.

Nous voulons maintenant avoir des paires de nombres (e, d) tel que :

$GCD\big(e, \varphi(N)\big) = 1$ (GCD = le plus grand diviseur commun)

Ceci implique qu'il existe une valeur d tel que $e \cdot d = 1 \; mod \; \varphi(N)$.

Chaque participant publie (e, N) en tant que leur clé publique, tandis que d devient leur clé privée.

- Prenons un très petit exemple (à ne jamais utiliser en réalité : les nombres sont trop petits) avec $p = 5$ et $q = 13$. Quelle est la valeur de $\varphi(n)$? Pouvez-vous trouver une paire (e, d) tel que $e \cdot d = 1 \; mod \; \varphi(N)$?
- Pourquoi doit-on toujours choisir p et q tel qu'ils soient larges ?

Le chiffrement RSA est un chiffrement a clé publique beaucoup utilisé en pratique. Disons que Bob a déjà publié un module et une clé publique (N, e) tel qu'il a la clé privée d correspondante à e. Si Alice veut chiffrer un message M pour Bob, elle calcule :

$$c = M^e \; mod \; N$$

Bob peut trouver le message en déchiffrant :

$$c^d = M^{d \cdot e} = M \; (mod \; N)$$

Pour la dernière égalité vous allez devoir me faire confiance : la raison pour laquelle $M^{d \cdot e} = M$ est le fait que $d \cdot e = 1 \; mod \; \varphi(N)$.

- Etant donne la façon de déchiffrement, quelle est la taille maximale du message M ?

- La valeur du chiffré c dépend seulement de la clé publique de Bob et du message. Est-ce que ceci donne une bonne sécurité ? (pensez à la répétition des messages)
- Pourquoi doit-on être sûrs qu'il est difficile de factoriser des larges N pour assurer la sécurité du chiffrement RSA ?

En réalité RSA est utilisé avec une méthode de randomisation qui fait le chiffrement *probabiliste* (ceci veut dire que si on chiffre le même message deux fois avec la même clé publique on obtient deux chiffrés différents). Il a été prouvé que la meilleure façon de faire le chiffrement RSA probabiliste est la méthode RSA-OAEP. Vous pouvez trouver plus d'infos sur RSA-OAEP ici : https://en.wikipedia.org/wiki/Optimal_asymmetric_encryption_padding

Le logarithme discret et le chiffrement ElGamal

Dans ce deuxième cas, nous allons choisir un nombre premier p qui est large. Nous voulons nous concentrer sur les nombres modulo p. Plus précisément, nous aurons besoin d'un sous-groupe de nombres parmi les nombres modulo p, notamment un *groupe cyclique multiplicative d'ordre premier q*.

Autrement dit, nous avons besoin d'un ensemble G qui contient q éléments modulo p tel que :

Le produit (mod p) de deux nombres dans G donne également un élément un G

Nous voulons que la multiplication mod p soit associative : $(g \cdot h) \cdot k = g \cdot (h \cdot k)$

Il y a un élément neutre I dans le groupe G, tel que $I \cdot g = g \cdot I = g$

Chaque élément de G a un inverse (mod p) en G .

Le groupe G que nous avons en tête va avoir un générateur g d'ordre q : ce qui veut dire $g^q = 1$ mod p et il n'existe pas un autre entier $l < q$ tel que $g^l = 1 \; mod \; p$. Les éléments du groupe G seront :

$$\{g^0 = 1, g, g^2 \; mod \; p, \dots g^{q-1} \; mod \; p\}$$

Nous disons que G est généré par g.

- Sachant que $g^q = 1$ mod p, étant donné g^2, quel est son inverse mod p ? Et l'inverse de g^4 ? Quel est l'inverse de g^a pour n'importe quel $0 \leq a \leq q - 1$?
- Prenons $p = 13$ et $g = 2$. En trouvant $g^2, g^3, \dots$ trouvez l'ordre de g mod 13. Est-il premeir

Pour le chiffrement ElGamal, on choisit aléatoirement un entier $x \in \{1, \dots, q-1\}$, qui sera la clé privée. La clé publique sera le tuple (p, q, g, g^x). Disons que Bob a déjà choisi sa clé privée et a déjà publié la clé publique.

Si Alice veut lui envoyer un message M, elle choisit un numéro aléatoire $r \in \{1, \dots, q-1\}$ et elle calcule

$$c_1 = g^r \; mod \; p; \quad c_2 = \; M \cdot (g^x)^r \; \; mod \; p$$

Elle envoie (c_1, c_2) à Bob.

- Comment peut Bob déchiffrer ce message, étant donné qu'il connait la clé privée et la clé publique ?
- Qu'est-ce qui se passe si Alice choisit toujours le même r lorsqu'elle envoie un message à Bob ?

La sécurité du chiffrement ElGamal dépend de la difficulté des problèmes suivants :

Le problème du logarithme discret : étant donnés (p, q, g, g^x), trouver x.

Le problème Diffie Hellman computationnel : étant donnés (p, q, g, g^x, g^y), trouver g^{xy}

Le problème Diffie Hellman décisionnel : étant donnés (p, q, g, g^x, g^y, g^z), décider si $g^{xy} = g^z$

- Disons qu'on aurait un algorithme qui peut résoudre le problème du logarithme discret de façon efficace. Comment peut-on alors casser le chiffrement d'ElGamal ?
- Et le problème Diffie Hellman computationnel ?

L'échange de clé Diffie-Hellman

Même si le chiffrement à clé publique a des diverses avantages (rappelez-vous le premier TD), il est beaucoup moins efficace que le chiffrement à clé symétrique.

- Rappelez les différences entre le chiffrement à clé symétrique et celui à clé publique. Quels sont les avantages du chiffrement à clé publique ?

Le chiffrement à clé symétrique nécessite une clé partagée a priori par les deux participants. Si deux parties n'ont pas partagé une clé en avance elles peuvent utiliser dans un premier temps un protocole d'échange de clé à clé publique. Le but d'un tel protocole est de permettre à deux participants de calculer une même clé symétrique, qui reste secrète, même si les deux participants échangent leurs messages sur un canal non-sécurisé.

Une des façons principales de faire un échange de clé c'est d'utiliser la méthode Diffie-Hellman, ainsi nommée pour Whitfield Diffie et Martin Hellman, ses inventeurs.

Alice		Bob
		Choisit x $\xleftarrow{\$} \{0,1\}^{128}$
	$\xleftarrow{g^x}$	
Choisit y $\xleftarrow{\$} \{0,1\}^{128}$ Calcule $ms = (g^x)^y$ (mod p) Calcule $K = \mathrm{HMAC}(ms; g^x \mid g^y)$	$\xrightarrow{g^y}$	Calcule $ms = (g^y)^x$ (mod p) et $K = \mathrm{HMAC}(ms; g^x \mid g^y)$

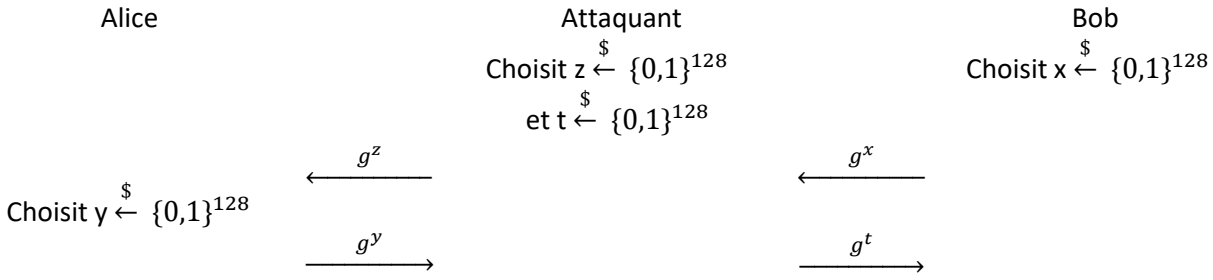
Dans ce protocole, Alice et Bob sont censés utiliser le même groupe F d'ordre premier p, qui a un sous-groupe cyclique G d'ordre premier q (où q a une taille de 128 bits), avec un générateur g publiquement connu. En pratique, les deux participants se mettent d'accord sur le groupe à utiliser dans le cours du protocole. Alice et Bob choisissent des entiers x, y aléatoirement entre 1 et q, et ils échangent les éléments g^x, g^y mod p. Finalement les deux parties calculent la valeur $ms = g^{xy}$, qui servira comme un secret commun qui leur permettra de calculer des clés, par exemple en utilisant la fonction HMAC avec la clé ms et en entré la concaténation de g^x et g^y.

Pour l'échange de clé Diffie Hellman on dit qu'un attaquant qui observe le protocole et connaît p,q, G et g n'est pas capable de distinguer la valeur de g^{xy} de la valeur d'un élément de G aléatoirement choisi.

- On se rappelle que G est un groupe cyclique d'ordre q avec générateur g. Comment peut-on représenter un élément quelconque de G ?
- Quel problème difficile nous garantit que g^{xy} est indistinguable d'un élément aléatoire de G ?

Une attaque active contre Diffie-Hellman

En cryptographie on parle de deux types d'attaquants : passifs et actifs. Un attaquant passif peut observer la communication (sniffer le canal par exemple), mais ne peut pas intervenir. Un attaquant actif peut également intervenir dans la communication, par exemple en envoyant des messages. Voyons un exemple d'un attaquant actif qui s'interpose entre Alice et Bob : une stratégie qu'on appelle Homme au Milieu (Man-in-the-Middle).



- Quel est l'effet de cette attaque ?
- Pourquoi cette attaque marche-t-elle ?
- Quelle serait une solution pour se prémunir contre ces attaques ?

L'authentification à clé publique : les signatures et les certificats

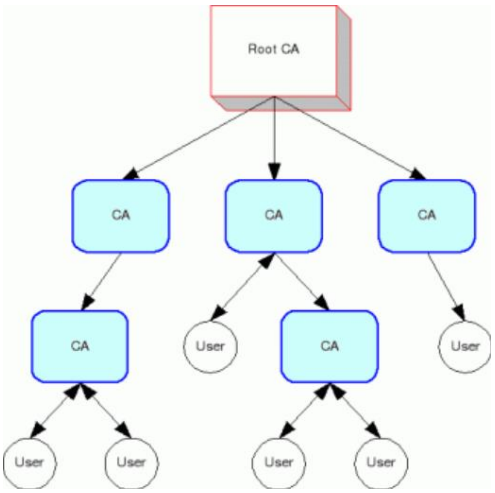
L'authentification de messages à clé publique se fait avec des signatures. Pour un schéma de signatures, chaque participant génère un tuple de clé : une clé privée et une clé publique. Informellement, le but d'un schéma de signatures est : tout le monde peut vérifier une signature, mais seulement le participant lui-même peut signer.

- Est-ce que l'algorithme de vérification nécessite la clé publique ou la clé privée ? Et l'algorithme de signature ?

Pour prouver son identité, l'utilisateur signe un message avec un algorithme de signature, qui retourne une signature pour ce message. Attention : une signature ne garantit pas la confidentialité du message signé ! Une fois la signature et le message envoyés, n'importe qui peut vérifier la signature en utilisant l'algorithme de vérification, la clé pertinente, le message et la signature. L'algorithme de vérification retourne 1 si la signature est vérifiée et 0 autrement.

Ajouter des signatures peut nous aider nous prémunir des attaques du type Hommes-au-Milieu – nous allons voir cela au TD prochain.

Toute schéma cryptographique à clé publique – qu'il s'agit d'un schéma de chiffrement ou d'un schéma de signatures – souffre d'un désavantage important, notamment il faut bien pouvoir s'assurer que la clé publique appartient bien à un certain participant. Ceci se fait en utilisant une structure de confiance qu'on appelle infrastructure de clés publiques (Public key infrastructure, PKI).



Source : galexia.com

Dans cet infrastructure, tout arbre de certification a un nœud racine (Root Certificate Authority – Root CA), qui a un tuple de clés de signature, une publique et une privée. Un niveau en dessous, il y a des autorités de certification intermédiaires (CA), qui ont également des clés publiques et privées de certification. Avec sa clé, la Root CA certifie la clé de certification de toute CA qui est un niveau en dessous.

Ce processus se répète plusieurs fois. Les utilisateurs reçoivent des certificats signés par des CA intermédiaires.

- La structure pyramidale des CAs oblige les usagers de présenter une chaine de certificats au lieu d'un seul : la clé de l'usager est certifiée par la CA en dessus de lui, mais la clé de la CA est certifiée par la CA un niveau en dessus, etc. jusqu'à la Root CA. Indiquez dans la figure précédente les chaines de certifications de chaque usager.
- Un certificat contient des informations importantes, comme par exemple la clé publique qu'ils certifient, ainsi qu'une signature de la CA intermédiaire sur cette clé publique. Quelles autres informations doit un certificat contenir, selon vous ?
- Les certificats utilisés de façon standard aujourd'hui sont les certificats X509. Comparez la structure d'un tel certificat avec vos réponses à la question antérieure. Que constatez-vous ?