

## Chiffrements asymétriques

Le chiffrement symétrique permet de communiquer de manière *confidentielle*, ou de préserver la *confidentialité* de nos données sur un disque dur.

Le problème avec le seul chiffrement symétrique, c'est qu'il ne permet pas de répondre aux besoins d'*authentification* (reconnaitre l'identité) et d'*authenticité* (le message n'a pas été modifié).

Pour comprendre ces termes, on utilise le scénario suivant. Alice et Bob sont 2 personnes qui veulent echanger des messages de manière sécurisée. Une 3e personne, appelée *Trudy*, ou Eve, joue le rôle d'un attaquant.

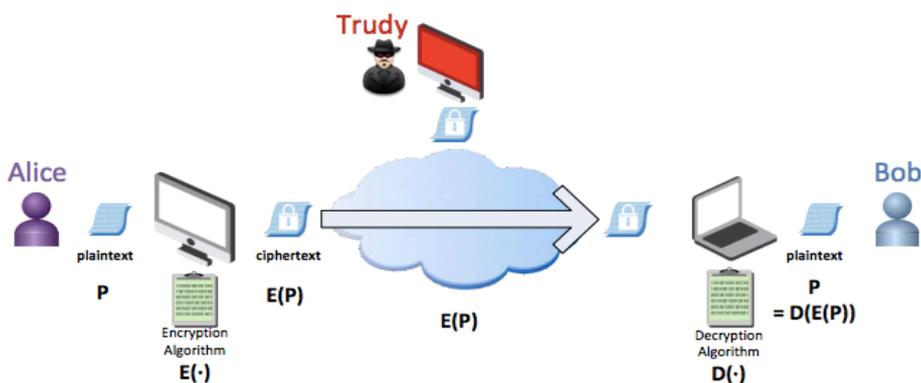


FIGURE 1 – Le trio Alice - Bob - Trudy

Le chiffrement asymétrique utilise plusieurs clés, une pour le ....., et une autre pour le ..... Il repose sur des propriétés mathématiques de l'arithmétique modulaire. (voir principe du chiffrement RSA)

Complément sur RSA (Grand Oral)

Lors de la confection de la clé publique  $e$ , le propriétaire de la clé utilise la valeur de 2 entiers  $p$  et  $q$  premiers entre eux (secrets). La clé privée  $d$  est alors calculée comme inverse modulaire de  $e$  (modulo  $(p - 1).(q - 1)$ ).

Pour chiffrer le caractère de valeur numérique  $M$  on fait :  $M^e \pmod n$  et pour déchiffrer :  $M^d \pmod n$  ou  $n=p \cdot q$   
 On exploite la propriété qui veut que  $(M^e)^d = M \pmod n$

Or, casser la clé revient à rechercher l'inverse modulaire de  $e$ , ce qui est impossible sans connaître  $p$  et  $q$ , du moins en un temps limité. Et avec les algorithmes classiques, casser la factorisation croit exponentiellement avec la longueur de la clé.

### 1.1 Principe du chiffrement RSA

#### 1.1.1 Confidentialité

Il doit y avoir 2 clés. L'une de ces clés doit rester secrète, et conservée par l'expéditeur du message : c'est la clé **privée**. L'autre clé utile pour le chiffrement et le déchiffrement est partagée : elle est dite **publique**. On peut indifféremment chiffrer avec la clé publique et déchiffrer avec celle privée, ou l'inverse.

Ainsi, comme la clé de Bob est publique, Alice peut utiliser cette clé pour chiffrer le message qu'elle veut lui envoyer, un peu comme si elle mettait le message dans un coffre (*celui de Bob*). Et Bob, qui possède la clé privée correspondante (la clé de *son* coffre) peut alors déchiffrer le message d'Alice.

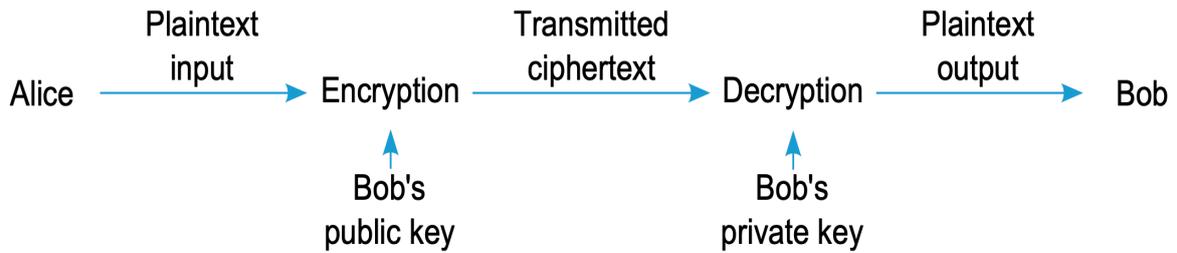


FIGURE 2 – Chiffrement asymétrique

1.1.2 Authentication

On peut utiliser cet algorithme pour authentifier l’auteur du message, et réaliser une **signature numérique**.

Imaginons que Bob utilise sa clé *privée* pour chiffrer un message M. Il l’envoie à Alice, qui, en utilisant la clé publique de Bob, pourra le déchiffrer. Elle pourra ainsi s’assurer que Bob en est bien à l’origine et que son contenu n’a pas été modifié.

En effet, comme il est le seul à posséder la clé *privée*, ce message M vient bien de Bob.

Compléter la situation où Bob envoie un message de type Bonjour à Alice, en utilisant sa clé privée :

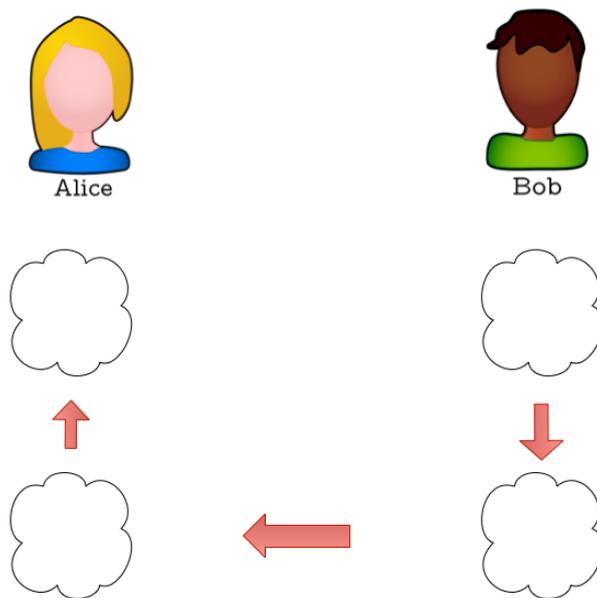


FIGURE 3 – Bonjour Alice - via chiffrement asymétrique

Le point faible de cet échange vient de la distribution de la clé publique entre Bob et Alice.

Cet échange n’est réellement sécurisé qu’à la condition que la clé publique de Bob, vienne bien de Bob. Et que personne ne se soit introduit dans la discussion pour substituer la clé publique de Bob par la sienne (attaque de *l’homme du milieu*).

Compléter la situation suivante où Eve réalise une attaque dite de l’homme du milieu. Elle envoie au préalable sa propre clé publique à Alice et Bob. Puis elle se fait passer pour chacun des 2 lors de la discussion.

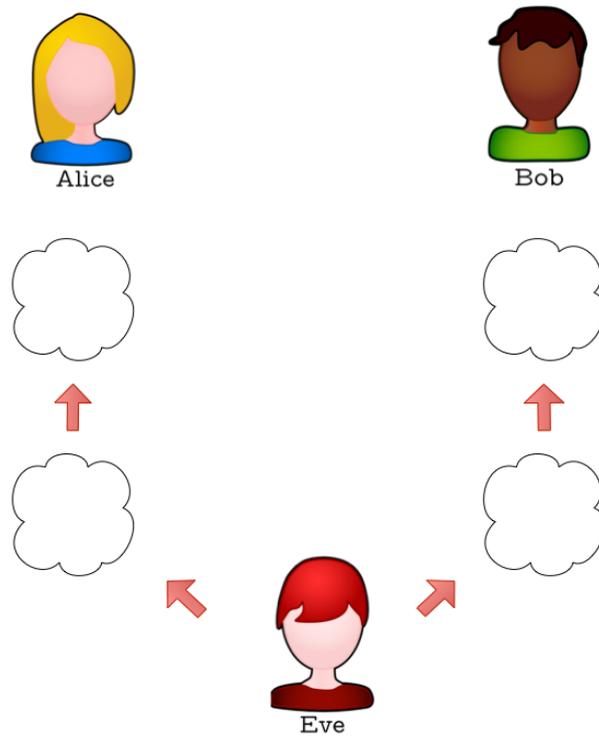


FIGURE 4 – Bonjour Alice, bonjour Bob - Eve

Pour s’authentifier, Bob doit d’abord envoyer à Alice un *certificat*, qui contient sa clé publique mais aussi d’autres informations qui permettent de vérifier la validité de ce certificat, en particulier l’*emetteur* : le nom de l’autorité de certification, une *infrastructure à clé publique* (PKI) qui a validé l’identité de Bob.

**Certificats** : les navigateurs font confiance à certaines autorités de certification, qui ont au préalable, certifié leurs clients (possesseurs de sites web).

Si la chaîne de confiance est rompue, votre navigateur vous en informe.

En pratique, le certificat est un ensemble d’informations sur le propriétaire du domaine, la durée de validité, l’algorithme utilisé pour l’authentification, la clé publique, et surtout l’*emetteur*. Lorsqu’Alice reçoit le certificat que lui envoie Bob, elle consulte l’*emetteur* du certificat pour vérifier qu’il s’agit bien du bon certificat, que celui-ci appartient bien à Bob.

### 1.1.3 Intégrité

Une fonction de hachage est un algorithme qui prend en entrée une chaîne de caractères de longueur quelconque, et qui retourne en sortie une chaîne de caractères de longueur n fixée, appelée empreinte. Il est presque impossible de retrouver la chaîne de caractères d’origine à partir de cette empreinte, tant la diffusion est importante suite à la modification d’une partie même infime du texte à chiffrer par cette méthode.

Une application classique de l’utilisation de la fonction de hachage est le stockage de mots de passe, ou bien la vérification de l’intégrité d’un message envoyé (comme un checksum). Mais aussi pour la constitution d’une blockchain :

Dans la blockchain, chaque bloc contient le hash du bloc précédent, les données des nouvelles transactions, et le nouveau hash constitué à partir de ces données (hash du bloc précédent + transactions).

## Principe d'une communication sécurisée HTTPS

Lors d'une communication HTTPS : les objectifs d'authentification, authenticité et de confidentialité sont remplis. Lors du protocole de communication, les chiffrements asymétriques et symétriques sont tous 2 utilisés :

- le client demande au serveur que celui-ci s'authentifie (1) et (2)
- le client envoie de manière sécurisée une clé de session qui servira pour la communication (3). L'envoi de la clé se fait à l'aide du chiffrement asymétrique. Le client utilise la clé publique du serveur pour cet envoi.
- Cette clé sera utilisée pour des chiffrements symétriques (4).



FIGURE 5 – établissement d'une communication sécurisée - OVH

## Exercices

### 3.1 RSA

1. Quel est la valeur de  $8^7$  modulo 40 ?
2. Module de chiffrement  $n=p*q$  : Vérifier que 55 est decomposable en produit de 2 nombres premiers  $p$  et  $q$ .
3. Calculer l'indicatrice d'Euler  $f = (p-1)*(q-1)$ . Quelles sont les valeurs de  $e$  que l'on peut choisir ? Donner en 3 possibles ( $e$  doit être premier avec  $f$ ).
4. Soit la clé publique  $e = 3$ . Quel est l'inverse de  $e$  modulo 40 ? Appeler ce nombre  $d$  (la clé privée).
5. Chiffrer le message  $M = 5$  avec la clé publique  $e$ , selon  $M^e(mod 55)$
6. Dechiffrer  $M^e(mod 55)$  avec la clé privée  $d$ , selon la même fonction que précédemment.

### 3.2 Cybersécurité

3. VRAI / FAUX : Dans un protocole de chiffrement asymétrique, toutes les personnes possédant la clé publique de Bob peuvent lui envoyer un message ?
4. VRAI / FAUX : Dans un protocole d'authentification, toutes les personnes possédant la clé publique de Bob peuvent vérifier sa signature ?

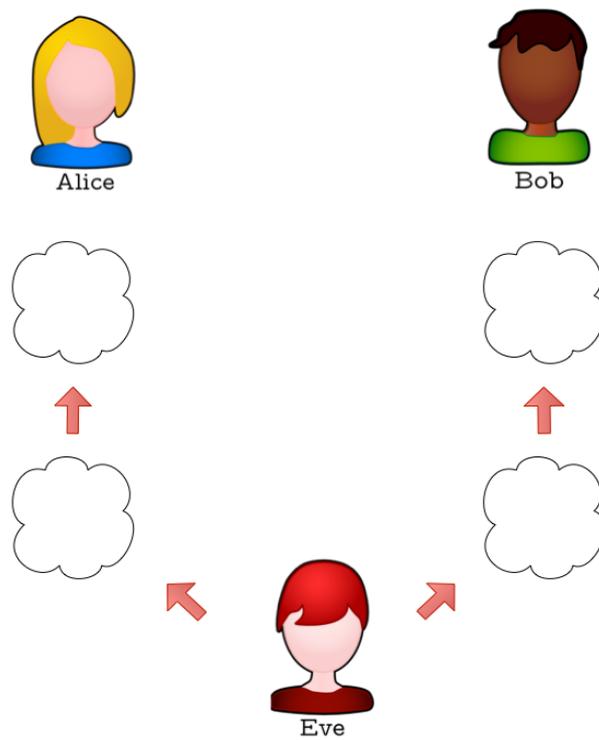


FIGURE 6 – Bonjour Alice, bonjour Bob - Eve

5. Supposons que Bob envoie à Alice un message chiffré de la manière suivante :  $C = \text{chiffrement}(m, \text{Bob\_public\_key})$ . Est-ce qu’Alice peut **authentifier** Bob avec ce message ?
6. Supposons que Bob envoie à Alice un message chiffré de la manière suivante :  $C = \text{chiffrement}(m, \text{Bob\_private\_key})$ . Est-ce que ce message est confidentiel ?
7. Alice reçoit la clé publique `Bob_public_key`. Comment peut-elle être sûre que c’est bien la clé publique de Bob, et non celle d’une autre personne ? Préciser le rôle du *tiers de confiance*.
8. Comment Eve peut-elle exploiter cette faille de sécurité ? (l’absence d’un tiers de confiance).
9. Bob envoie a Alice un message, accompagné du Hash  $H(M)$  de ce même message, et il chiffre le tout avec sa clé privée `Bob_private_key`. Quel est l’intérêt d’avoir ajouté  $H(M)$  ?