# RSA

## Définition

Le cryptage RSA est un algorithme de cryptographie asymétrique largement utilisé qui assure une communication sécurisée, la confidentialité et l'authentification. Il utilise une clé publique pour le chiffrement et une clé privée correspondante pour le déchiffrement.

## Comment fonctionne le cryptage RSA

### Génération des clés :

* Choisissez deux nombres premiers distincts, p et q.
* Calculez leur produit, n = p \* q. Ceci est le module utilisé pour le chiffrement et le déchiffrement.
* Calculez la fonction indicatrice d'Euler, φ(n), qui est le nombre d'entiers positifs inférieurs à n qui sont premiers entre eux (c'est-à-dire qui n'ont pas de facteurs communs) avec n. Dans RSA, φ(n) = (p - 1) \* (q - 1).
* Sélectionnez un entier, e, tel que 1 < e < φ(n) et e est premier avec φ(n). C'est l'exposant public utilisé pour le chiffrement.
* Calculez l'exposant privé, d, qui est l'inverse multiplicatif modulaire de e modulo φ(n). En d'autres termes, (e \* d) mod φ(n) = 1. C'est la clé privée utilisée pour le déchiffrement.

### Chiffrement :

* Obtenez la clé publique du destinataire, qui se compose du module (n) et de l'exposant public (e).
* Convertissez le message en clair en une représentation numérique. Généralement, chaque caractère a assigné un nombre unique à l'aide d'un schéma de codage des caractères (par exemple, ASCII ou Unicode).
* Divisez le message en blocs plus petits si nécessaire et représentez chaque bloc par un entier.
* Pour chaque bloc, calculez le texte chiffré à l'aide de la formule de chiffrement : texte chiffré = (texte en clair ^ e) mod n. Ici, ^ indique l'exponentiation.

### Déchiffrement

* Le destinataire utilise sa clé privée, composée de l'exposant privé (d) et du module (n).
* Recevez le texte chiffré.
* Appliquez la formule de déchiffrement : texte en clair = (texte chiffré ^ d) mod n. L'exposant privé, d, est utilisé pour le déchiffrement.
* Si le texte en clair a été divisé en blocs, reconstituez les blocs pour obtenir le message original.

## L’exécution

* La taille des nombres premiers : Lors de la génération des clés RSA, la taille des nombres premiers (p et q) a un impact direct sur la sécurité et les performances. Des nombres premiers plus grands renforcent la sécurité mais peuvent ralentir les opérations de chiffrement et de déchiffrement.
* La taille de la clé : La longueur de la clé RSA (c'est-à-dire la taille du module n) affecte également la vitesse d'exécution. Des clés plus longues offrent une sécurité accrue, mais elles peuvent ralentir les opérations.
* La puissance de calcul du système : Les performances de votre système informatique jouent un rôle important dans l'exécution de l'algorithme RSA. Des processeurs plus rapides et une puissance de calcul élevée peuvent accélérer les opérations de chiffrement et de déchiffrement RSA.

## Conclusion

La sécurité du RSA repose sur la difficulté de factoriser de grands nombres. Tant que les facteurs premiers du module restent secrets, il est difficile de déduire la clé privée à partir de la clé publique. Par conséquent, n'importe qui peut utiliser la clé publique du destinataire pour chiffrer des messages, mais seul le destinataire avec la clé privée correspondante peut les déchiffrer.

# XOR

## Définition

Le chiffrement XOR (OU exclusif), également connu sous le nom de chiffrement XOR bit à bit, est une technique de chiffrement simple qui opère au niveau binaire sur les données. Il utilise l'opération XOR pour combiner le texte en clair avec une clé secrète afin de produire le texte chiffré

## Comment fonctionne le cryptage XOR

### Génération de la clé :

* Choisissez une clé secrète, qui est une séquence binaire de la même longueur que le texte en clair.

### Chiffrement :

* Convertissez le texte en clair en forme binaire. S'il est initialement sous forme de texte, il est généralement encodé à l'aide d'un schéma de codage des caractères (par exemple, ASCII ou Unicode).
* Répétez la clé secrète si sa longueur est plus courte que le texte en clair. Si la clé est plus longue, tronquez-la pour correspondre à la longueur du texte en clair.
* Effectuez l'opération XOR entre chaque bit du texte en clair et le bit correspondant de la clé secrète. L'opération XOR produit 1 seulement si les bits d'entrée sont différents ; sinon, elle produit 0.
* Le résultat de l'opération XOR devient le texte chiffré, qui est une séquence binaire.

### Déchiffrement :

* Pour déchiffrer le texte chiffré, répétez le processus de chiffrement en utilisant la même clé secrète.
* Faites l'opération XOR entre le texte chiffré et la clé secrète, bit par bit, pour récupérer le texte en clair d'origine.

## Conclusion

Il est important de noter que le chiffrement XOR est une forme relativement faible de chiffrement et il n'est pas adapté pour sécuriser des informations sensibles à lui seul. Il présente plusieurs vulnérabilités, telles que la vulnérabilité aux attaques par texte clair connu et l'analyse de fréquence. Le chiffrement XOR est souvent utilisé comme un élément de base dans des algorithmes de chiffrement et des protocoles plus complexes plutôt que comme une méthode de chiffrement autonome.