Bienvenue dans ce tutoriel consacré à la cryptographie!

Le but de ce tutoriel est de vous apprendre les bases et les techniques utilisées en cryptographie et surtout de vous sensibilisez au choix de mot de passe ainsi que d'éviter les pièges concernant la sécurité.

Ce cours va vous apprendre dans un premier temps:

1. L'Histoire de la cryptographie jusqu'à aujourd'hui.
2. Les différents types d'algorithmes de chiffrement.
3. Les différents algorithmes de chiffrement.
4. Comment casser ces algorithmes.
   I. L'analyse fréquentielle.
   II. Attaque par brute-force.
   III. Attaque par dictionnaire.
   IV. Attaque par table arc-en-ciel.
   V. Le vol de clé.
   VI. L'attaque de l'homme du milieu.
5. Conclusion

Ce tutoriel a été complètement réalisé par Neyort

Si aucune connaissance complexe n'est demandée pour suivre ce cours, il faudra tout de même être a l'aise avec différentes formules mathématiques dont je donnerais un bref rappel tout au long de ce

tutoriel

Le mieux pour suivre est de connaître au minimum ces critères tout en sachant les appliquer:

- Opérations arithmétiques de bases (addition, soustraction, multiplication, division, modulo..).
- Quelques connaissances sur l'aléatoire (nous y reviendrons ne vous inquiétez pas).
- La notion de fonction.
- Avoir du courage pour tout lire :P

Bien entendu ce cours s'accompagnera de nombreuses illustrations et ne sera pas qu'un pavé indigeste d'une trentaine de pages

Vous voici près pour affronter les pires algorithmes cryptographiques de votre vie

Cliquez sur le bouton "Commencer" et surtout attachez vos ceintures, car vous allez commencer par faire un saut dans le temps jusqu'à arriver à .... -1600 ans avant JC! :D

Comme promis nous allons commencer ce cours par un peu d'histoire

Le disque de Phaïstos :

Le disque de Phaïstos ou disque de Phaestos est un dique d'argile découvert en 1908 par Luigi Pernier en Crète. Il datterait selon les experts de l'âge de bronze soit approximativement -1600 ans avant Jésus Christ. Il est recouvert sur ses deux faces d'une multitude de hiéroglyphes de 45 sortes soit 241 signes au total.

Le disque de Phaïstos (Face A)

Si ça vous voulez tenter de percer son secret, vous allez devoir chercher car personne n'a encore su le déchiffrer de nos jours😄

Le Scytale:

Le scytale est une sorte de tube autour duquel l'expéditeur du message enroulait du papier et écrivait dessus. Une fois écrit il suffisait de l'enlever du tube pour que le texte soit illisible. Pour le lire, il fallait avoir le même tube que l'expéditeur et ré-enrouler le papier autour de celui-ci. Simple mais efficace! 😄
Ce procédé a été utilisé dans les environs de -404 ans avant Jésus Christ.

Image d'un scytale

Voila je pense que la photo rend les explications plus claires 🙂

Le chiffre de César:

Je pense que tous le monde connais Jules César (ou tout du moins de nom parce que je l'ai jamais rencontré😄) mais peu de gens savent qu'il a été l'un des premiers à utiliser la cryptographie pour des fins militaires.

Attention!
Je dis bien pour des fins militaires car avant lui (notamment certains scribes) ont utilisés la cryptographie pour rendre incompréhensible les écritures tombales de certains sarcophages.

Sa méthode était simple. Elle consiste à faire un chiffrement par substitution mono alphabétique. En d'autres termes, il s'agit de donner un mot et une clé pour que chaque lettre du mot soit déplacée du nombre par la clé.

Exemple:

Pour la lettre 'B' avec une clé de 5 cela donnera la lettre 'G'.
Pour la lettre 'O' avec une clé de 5 cela donnera la lettre 'T'.
Pour la lettre 'N' avec une clé de 5 cela donnera la lettre 'S'.
Pour la lettre 'J' avec une clé de 5 cela donnera la lettre 'O'.
Pour la lettre 'O' avec une clé de 5 cela donnera la lettre 'T'.
Pour la lettre 'U' avec une clé de 5 cela donnera la lettre 'Z'.
Pour la lettre 'R' avec une clé de 5 cela donnera la lettre 'W'.

Donc si vous chiffrez le mot "bonjour" avec une clé de 5 vous obtiendrez le mot "gtsotzw".
Donc seul les personnes connaissant la clé pouvaient déchiffrer votre message.
Mais comme nous le verrons dans la partie "Comment casser différents algorithmes" ce chiffrement est très ... obsolète

Petit truc marrant: Essayez de chiffrer le mot "oui" avec une clé de 10 grâce à l'algorithme de César. Vous allez être....surpris

Voilà! Nous avons fait le tour des principales méthodes de chiffrement jusqu'à nos jours
Le chapitre suivant sera surtout un chapitre avec du vocabulaire. Il faudra encore attendre un autre petit chapitre avant de découvrir les algorithmes d'aujourd'hui, qui eux vous le verrez, ne sont pas aussi simples que celui de César

Les différents algorithmes de chiffrement

Nous allons donc voir dans cette partie du tutoriel, différents algorithmes. Comme nous avons vu juste avant les différences entre certains algorithmes de chiffrement, nous allons voir maintenant

comment ces algorithmes fonctionnent. Pour cela je vais vous présenter les principaux car il serait impossible d'en faire une liste exhaustive.

Ces algorithmes sont assez particuliers, car nous avions, jusqu'à présent, étudié des algorithmes de chiffrement "classiques" et surtout réversibles. Les algorithmes que je vais maintenant vous présenter sont des algorithmes de hashage.

Un algorithme de hashage est une fonction qui calcule, à partir du texte à chiffrer, un hash (ou empreinte, les deux se disent).

Par exemple, si nous chiffrons grâce à l'algorithme MD5 le mot "Creanet" (sans les guillemets et avec la majuscule) nous obtenons cette empreinte là:

40c305bd2bb576b820f05a41ecb72401

Une fonction de hashage "parfaite" (c'est à dire n'existant pas ) devrait attribuer un hash unique pour chaque texte donné.

Une fonction de hashage est, par définition, normalement irréversible. C'est à dire qu'il est en théorie impossible (j'insiste sur en théorie!) de revenir au texte de départ grâce au hash.

Le chiffrement MD5:

L'algorithme MD5 est l'un des algorithmes de chiffrement le plus connu de nos jours. Seul problème, ... il est cassé!

Cassé, signifie (en cryptographie), que certaines personnes ont trouvé le moyen de déchiffrer le texte chiffré mais sans en avoir la clé!

Ce qui signifie que cet algorithme n'est surtout pas à utiliser!

C'est un groupe de chercheurs chinois qui ont découvert certaines failles potentielles dans l'algorithme pour, enfin, en arriver au bout en été 2004.
Malheureusement, cet algorithme est encore énormément utilisé par des système qui n'ont visiblement pas été mis à jours. Cela est extrêmement dangereux car du coup, depuis 2004, quasiment tout le monde peut avoir accès à des données chiffrées grâce au MD5!

 Attention 

Ne croyez pas qu'il suffit d'utiliser un algorithme précis pour casser n'importe quel hash MD5!
Ces chercheurs se sont concentrés des années sur leurs recherches et ont réussi à prouver que l'on pouvait créer une collision complète sans passer par une méthode de recherche exhaustive!

Le "seul" moyen actuel pour "casser" n'importe quel hash MD5 (et d'autre d'ailleurs) est de faire une recherche exhaustive aussi appelée brute-force.
Nous y reviendrons dans la partie "comment casser les algorithmes de cryptage", et nous y verrons aussi pourquoi elle n'est pas vraiment réalisable en pratique.

Voilà, ce chapitre est terminé. Oui il a été très cours car il fallait juste que je vous montre cet autre type d'algorithme qu'est le hashage. Je vous ai expliqué cela à travers l'exemple du MD5, mais vous pouvez trouver pleins d'autres algorithmes de hashage, comme:

-SHA-1 --> wikipédia.
-DES --> wikipédia.
-RSA --> wikipédia.

Si nous résumons tout ça, nous pouvons dire que:

-Il faut toujours utiliser des algorithmes de hashage.
-Les algorithmes de hashage sont plus sécurisés que les autres s'ils ne sont pas cassés!

Donc vérifiez toujours où en est l'algorithme que vous utilisez car il se peut que du jour au lendemain, l'algorithme le plus sur ne soit cassé et ne devienne aussi sur qu'un..... chiffrement de César.

Dans la suivante et dernière partie, nous verrons tous les moyens possibles pour casser un algorithme de cryptage.

Comment casser un algorithme de cryptage?

En voici une excellente question! Juste avant d'y répondre je tiens à préciser une chose:

Ce chapitre va vous montrer les aspects purement théoriques de ces techniques. Elles sont expliquées ici à des fins d'études et de compréhension de la sécurité des chiffrements en informatique. Elles ne doivent jamais être réalisées pour causer du mal, la perte ou destruction de données sans s'y limiter. De plus la loi punis sévèrement toute violation de la vie privée!

Que cela soit clair: Nous ne vous donnerons pas les outils nécessaires permettant de réaliser des attaques sur un système de chiffrement. Nous aborderons juste les aspects théoriques!

Les conditions étant posées, je vous propose de continuer normalement ce tutoriel 

Nous disions donc: mais comment certains arrivent-ils à trouver des failles de sécurité dans des algorithmes alors que c'est super compliqué? @_@

Et bien ils utilisent, la plupart du temps, ces techniques:

-L'analyse fréquentielle.
-Attaque par brute-force.
-Attaque par dictionnaire.
-Attaque par table arc-en-ciel.
-Le vol de clé.
-L'attaque de l'homme du milieu.

Et bien d'autres comme l'attaque par collisions, par pseudo-collisions, attaque par préimage, etc....
Cependant elles sont trop complexes pour être abordées dans ce cours qui était initialement prévu pour les personnes commençant la cryptographie.

Nous allons commencer des plus simples pour aller progressivement vers les plus poussées.
Nous allons tout de suite partir voir ... l'analyse fréquentielle avec notre bon vieux chiffrement de César 

L'analyse fréquentielle:

Comme son nom l'indique, l'analyse fréquentielle permet de trouver la clé de chiffrement de certains algorithmes juste en analysant la fréquence d'apparition des lettres dans le texte crypté.
Car il faut savoir que dans chaque langue, il y a des lettres que vous utiliserez plus que d'autre. Nous allons prendre, ici, l'exemple du français mais cette technique fonctionne pour n'importe quelle langue mais avec un nombre très petit d'algorithmes!

L'algorithme de César est faillible à cette technique, c'est pourquoi nous allons tenter de casser ce mot de passe:

gvierix

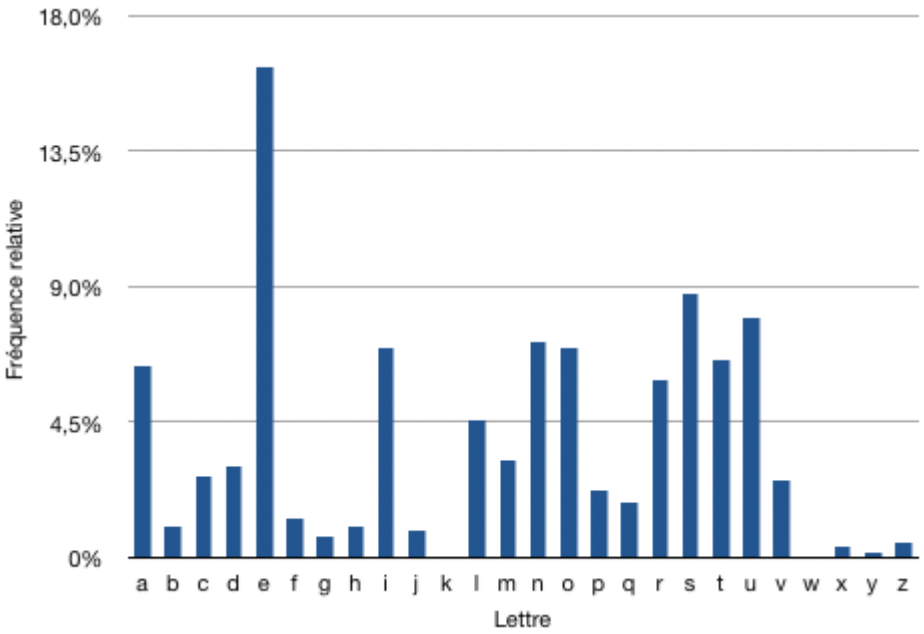
Bien sur n'essayez même pas de tenter la même chose avec un algorithme actuel comme le MD5 ou le RSA, sa ne servirait à rien 

Passons au cassage du texte "gvierix".

Nous avons déjà un indice super important: le texte chiffré l'est avec le chiffrement de César. Ce qui veut dire qu'il nous suffit d'avoir la clé (le nombre qui permet de décaler les lettres)

pour pouvoir déchiffrer le texte.

Nous allons pour cela utiliser cet histogramme:



Source: wikipédia

On peut y voir très nettement que le "e" est en première place dans le niveau d'apparition, suivi de près par le "s", etc...

Regardons maintenant de plus près le texte à déchiffrer:

gvierix

On y voit que la lettre le plus présente est le "i". Or dans la langue française on s'aperçoit que la lettre la plus présente est le "e".

Nous pouvons donc en conclure que le "i" du texte chiffré est en réalité un "e"!

Cela nous donne donc

gveerex

Les lettres vertes sont les lettres connues alors que celles en rouges sont celles encore "chiffrées".

Il ne nous reste plus qu'à "calculer" combien il y a d'écart entre le "e" et le "i" dans leur position dans l'alphabet. Il y en a 4. Donc la clé de chiffrement est tout simplement 4.

Tenez, on va s'amuser à déchiffrer le restant du texte:

g - 4 = c

v - 4 = r

e - 4 = a

r - 4 = n

x - 4 = t

Ce qui nous donne:

Creanet

Pas si robuste que ça cet algorithme

Ici nous avons testé le cassage avec un seul mot mais cela n'aurait peut être pas été possible! Prenez cet exemple de mot chiffré qui figure dans la langue française:

rgvot

et bien essayez de le casser avec la méthode que nous venons de voir

.....

Vous n'y arrivez pas (ou vous n'avez pas cherché :P)?

C'est normal, car toutes les lettres de ce mot ne sont présente qu'une seul fois!

C'est pour cela que pour casser un algorithme de substitution mono-alphabétique (algorithme qui remplace une lettre par une autre lettre) le mieux est d'avoir une phrase ou deux minimum de chiffrées avec l'algorithme.

Voilà, nous en avons fini avec l'analyse fréquentielle, qui était la méthode la plus simple. Nous allons maintenant rentrer dans des méthodes plus .... subtiles.

Au programme du chapitre suivant, nous verrons comment casser un algorithme de chiffrement par l'attaque brute-force, l'attaque au dictionnaire et l'attaque par tables arc-en-ciel

PS: Pour ceux qui aimerais savoir le mot qui est chiffré derrière "rgvot" il s'agit du mot "lapin" chiffré avec une clé de 6 :P

L'attaque par brute-force:

L'attaque par brute force a un avantage incontestable:

Elle permet de casser quasiment n'importe quel texte chiffré avec quasiment n'importe quel algorithme même le plus sécurisé! En plus elle est super simple à effectuer!

Mais elle a un énorme problème: elle est longue.

Quand je dis longue, c'est vraiment longue un! :P

Je vais vous expliquer d'abord le mode de fonctionnement puis vous dire pourquoi elle n'est d'aucune utilité pour casser un algorithme.

Fonctionnement:

Il s'agit d'une méthode dont le but est de tester **toutes** les combinaisons et d'attendre de trouver la bonne.

Exemple:

Vous vous trouvez devant une page protégée par un mot de passe. Vous voulez y accéder. Vous aller l'attaquer avec une attaque par brute-force. Le logiciel utilisé va tester toutes les possibilités de mot de passe comme ceci:

a
aa
aaa
aaaa
....

ab
aba
abaa
....

Avec tous les caractères! Donc tous les chiffres, les lettres minuscules, les lettres majuscules, les caractères spéciaux,...

Je pense que vous identifiez maintenant aisément le problème: si vous avez un ordinateur classique vous allez en avoir pour longtemps! Mais même avec un ordinateur puissant vous risquez de devoir attendre! 

En effet, si le mot de passe ne fait qu'un, deux, voir trois caractères, cela ne devrait pas poser trop de problèmes. Mais déjà à partir de quatre vous allez perdre patience et étant donné que la longueur minimale recommandée pour un mot de passe est de 8 caractères ....

L'attaque par dictionnaire:

C'est en fait quasiment la même que celle ci-dessus, à la différence que celle-ci est légèrement plus "intelligente" Au lieu de tester bêtement toutes les possibilités une par une, cette technique se base sur un dictionnaire complet de la langue utilisée avec les mots de passe les plus utilisés.
Quand on y regarde de plus près, ce n'est pas si absurde, car il n'y a pas beaucoup de personne dont leur mot de passe doit être quelque chose comme:

lkdqsldss541;;-((-(54

Souvent tout le monde fait la chose à ne pas faire, c'est à dire: utiliser un mot de passe simple, qui est dans le dictionnaire (comme le nom de son animal, etc..) et réutilise ce même mot de passe pour chaque chose nécessitant un mot de passe!

Ceci n'est surtout à pas faire!

Privilégiez donc un mot de passe difficile. (Voyez à la page de conclusion de ce tutoriel une méthode simple et efficace de création de mot de passe --> ici).

L'attaque par table arc-en-ciel:

Ce type d'attaque est la moins connue. Elle est pourtant puissante mais, comme toutes les méthodes de ce chapitre, est très longue.

C'est un mélange de l'attaque par brute-force et de l'attaque par dictionnaire sauf que le dictionnaire est un dictionnaire d'empreintes.

Rappel: Une empreinte (en cryptographie) ou hash, est une suite de caractères qui représente la forme chiffrée d'un texte.

Ce dictionnaire se présente donc ainsi:

hash motCorrespondant
hash motCorrespondant
hash motCorrespondant
.....

Ainsi, le logiciel compare le texte à déchiffrer avec tous les hashs de son dictionnaire. Et si il rencontre le même, il trouve automatiquement le texte non chiffré.

Le vol de clé

Pour cette technique, la description sera courte. Il s'agit tout simplement de voler la clé utilisée pour le chiffrement du message à déchiffrer. Ainsi, il permet de déchiffrer n'importe quel algorithme ne reposant que sur une seul clé (ce qui tente d'ailleurs à devenir obsolète de nos jours avec l'apparition du chiffrement asymétrique).

Le vol de cette clé (ou mot de passe) est la méthode la plus courante, car souvent assez simple. On peut citer comme méthodes: le phishing, keylogger, troyen, écoute de réseau, tempest, etc...

La méthode dite "tempest" est la moins connue, c'est pourquoi je vais vous en faire un bref résumé:

Il s'agit d'une méthode assez étonnante mais nécessitant beaucoup de moyens matériel.

Explications:

Chaque machine permettant le traitement de l'information ont des CE (pour Compromising Emission, ou en français, émissions compromettantes). Elles peuvent être d'ordre mécanique (machine à écrire), acoustique, électromagnétique ou électrique (ordinateur).

Le problème viens du fait que à chaque fois que vous appuyez sur une touche de votre clavier, celle-ci propage un signal électrique qui lui est propre et cela sur un champs d'action d'envions 20 mètres.

Pour ces raisons, un attaquant placé à une vingtaine de mètres pourrait intercepter ces signaux électriques et les interpréter grâce à un appareil spécialisé.

Cette technique à été testée avec succès par un groupe de chercheurs.
Vous pouvez regarder la vidéo ci-dessous dans laquelle ces chercheurs filment leur "exploit", vidéo qui est bien sur, sans trucages.

Leur performance se trouve à partir de 1 minute 25.

http://www.youtube.com/watch?feature=player_embedded&v=uNGCAiYeQSg

Comme vous pouvez le voir, le matériel nécessaire est assez conséquent sans parler du programme qu'ils ont du créer pour le décodage qui doit demander de solide connaissance en programmation dans le domaine électronique.

Leur site: ici

Derrière toutes ces techniques, le but recherché est toujours le même: voler des informations privées sans que l'utilisateur ne s'en aperçoive.

Dans la prochaine partie, nous verrons l'attaque de l'homme du milieu

L'attaque de l'homme du milieu

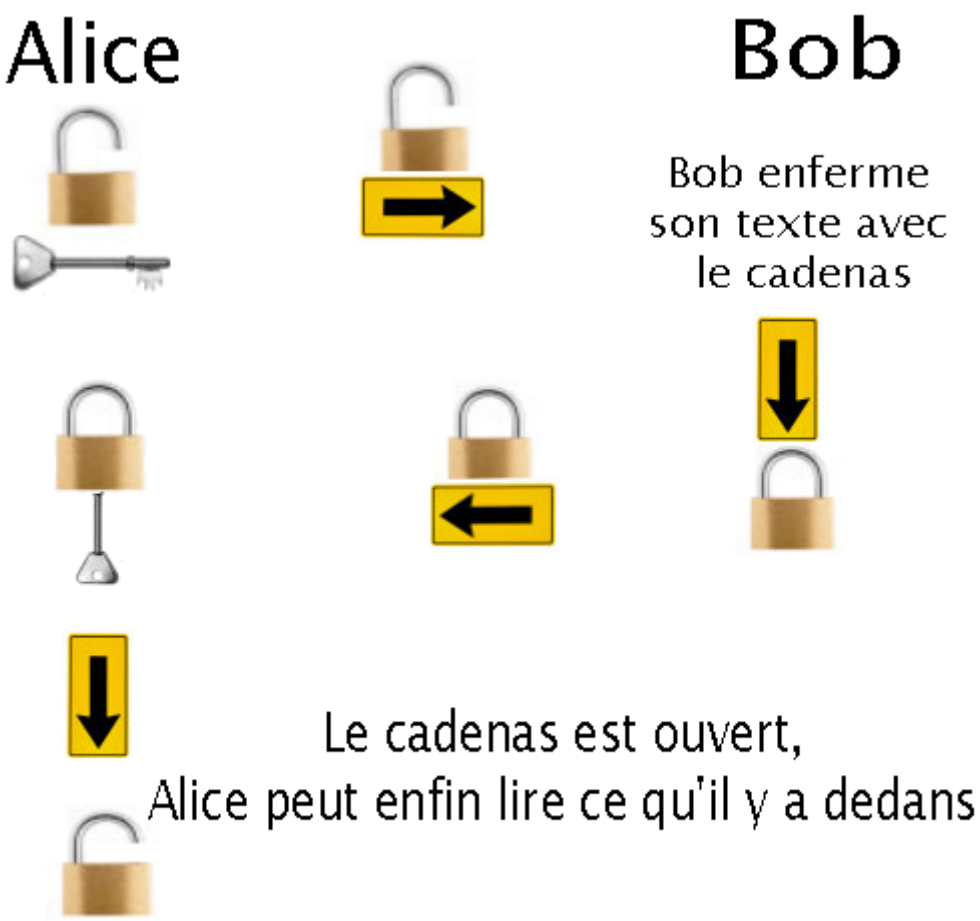
Cette technique est l'une des plus durs à mettre en place parmi toutes celles que l'on a déjà vues.

Elle permet de décrypter un échange (entre une ou plusieurs personnes) chiffré avec un algorithme de chiffrement asymétrique.

La particularité de ce type de chiffrement est, je vous le rappel, le fait qu'il utilise deux clés.

L'une est la clé publique et l'autre la clé privée.

Je vous remets le schéma en tête:



Nous allons, pour expliquer cette méthode, ajouter un dernier personnage: Mallory.

Petite parenthèse concernant ces prénoms:

Si vous avez déjà lu des articles parlant de cryptographie, il est quasiment certains que vous ayez déjà entendus parler de Bob et Alice.

C'est en fait une convention. Avant, les personnages fictifs utilisés pour expliquer en détail un algorithme ou un échange en informatique se schématisait par un personnage A et B.

Mais un jour on a remplacé le A par Alice et le B par Bob. C'est plus simple à retenir.

Mallory quand à lui, est le personnage qui cherche toujours à capturer les conversations entre Alice et Bob et c'est dans sa peau que nous allons nous mettre pour comprendre la technique de l'attaque de l'homme du milieu.

Le but de cet attaque est donc de récupérer les informations partagées.

Pour ce faire, récapitulons le fonctionnement d'un échange classique avec un chiffrement asymétrique:

1. Alice envoie à Bob l'algorithme de chiffrement et garde, pour elle seule, l'algorithme de déchiffrement.

2. Bob chiffre son message avec l'algorithme d'Alice et lui envoie le message chiffré.

3. Alice reçoit le message de Bob et le déchiffre avec son algorithme de déchiffrement pour pouvoir lire son message.

Bien maintenant passons à l'attaque:

Mallory se met entre Alice et Bob et récupère tout se qui passe sur le réseau.

Lorsque Alice envoie à Bob son algorithme de chiffrement, Mallory lui, va échanger cette algorithme contre le sien et l'envoyer à l'adresse prévue (chez Bob).

Bob ne se doutant de rien, va chiffrer son message avec l'algorithme de Mallory et renvoyer à Alice le message chiffré.

Avant qu'Alice ne reçoive le message chiffré, Mallory le déchiffre avec son propre algorithme et lit le texte écrit par Bob.

Enfin, pour ne pas éveiller les soupçons d'Alice et Bob, il chiffre le message de Bob avec l'algorithme d'Alice qu'il avait précédemment volé et lui envoie le message.

Alice déchiffre le message pour lire la même chose que Mallory.

Voilà, c'était certainement la méthode la moins évidente à mettre en place mais c'est important de savoir qu'elle existe et donc qu'elle peut être très dangereuse.

Je vous invite maintenant à cliquer sur "Conclusion" pour aborder la dernière page de ce tutoriel de cryptographie. 

Conclusion

Nous voilà (déjà :P) arrivé au terme de ce tutoriel consacré à la cryptographie et à son adaptation dans le monde informatique.

Comme vous avez pu le constater il y a encore énormément de progrès à faire que se soit sur les algorithmes de chiffrement eux-mêmes comme sur leur manière d'implantation.

Nous avons vu aussi que la sécurité dépendait aussi énormément de l'utilisateur. Ainsi, la meilleure solution serait, à mon sens, de sensibiliser les gens aux menaces comme le phishing ou sur le choix de leurs mots de passe!

A propos, savez-vous ce qu'est un vrai bon mot de passe?

Un bon mot de passe est donc un mot de passe qui:

-A beaucoup de caractères.
-N'est pas dans le dictionnaire.
-Est composé de caractères spéciaux.
-N'a pas de suite logique dans ses caractères.
-N'est pas logique comme 123456, sa date de naissance, ect...

J'ai d'ailleurs, pour vous, une méthode vous permettant de vous créer un mot de passe répondant à tous ces critères et qui sera, en plus, simple à retenir, je vous le promets

C'est simple et efficace:

Prenez une phrase que vous aimez bien et simple à retenir. Dans mon exemple je prendrais "J'adore le site internet de Creanet!!!"

Prenez tout d'abord les premières lettres de chaque mots et laissez la ponctuation telle quelle, ce qui me donne dans mon cas:

J'alidC!!!

Et bien je peux vous garantir que c'est déjà plus dur à trouver comme mot de passe

Car se n'est ni dans le dictionnaire, ça n'a pas de sens et il y a un mélange de lettres minuscules, majuscules, et signes de ponctuations.

Vous savez maintenant normalement choisir un mot de passe très robuste!

Sur ces derniers mots j'espère que vous avez appris des choses et pris du plaisir à lire ce tutoriel. Si

vous avez des questions n'hésitez pas à les poser dans la catégorie questions à propos d'un tutoriel je ferais mon maximum pour y répondre et ce dans les plus brefs délais possibles 

Tutoriel écrit entièrement par Neyort 

Site internet officiel du tutoriel : http://creanet.wifeo.com