15/01/2019

RAPPORT

CRYPTOGRAPHIE

JAY Damien / KROUK Hédhie / FAISY Corentin /
COURBET David / VINCENT-HARRISSON Yoann
IUT DIJON AUXERRE

SOMMAIRE

Descriptif du projet :

Le but du projet est de créer deux programmes :

- Le premier programme permet le codage et le décodage d'une clé transmise à l'aide du principe de Diffie-Hellman.
- Le second permet le codage et le décodage d'une clé transmise par chiffrement par transposition.

Pour réaliser ces deux programmes nous avons utilisé NetBeans pour les coder ainsi que le langage JAVA. Nous avions également besoin de comprendre les deux principes avant de pouvoir les coder.

Pour ce projet, nous mettons en scène deux agents : Bob et Alice, deux scientifiques très réputés qui souhaitent pouvoir communiquer régulièrement par mail. Ils souhaitent donc crypter leurs communications. Ils décident donc d'utiliser un protocole de cryptographie symétrique. Pour cela, les deux agents doivent se mettre d'accord sur une clé qui servira aussi bien à coder qu'à décoder leur message. Néanmoins, les deux agents ne peuvent pas se rencontrer pour se mettre d'accord sur une clé. Alice propose donc le principe de d'échange de clé de Diffie-Hellman et Bob celui d'un chiffrement par transposition.

Nous devons donc départager Alice et Bob dans leurs choix sur quelle méthode est la plus sécurisée pour qu'ils puissent s'envoyer une clé en toute sécurité. Pour cela, nous avons étudié les dangers du principe de Diffie-Hellman notamment l'attaque de l'homme du milieu et ce qui permet de contrer cette attaque avec un certificat et sa limite. Ensuite, nous avons étudié le problème du logarithme discret et son lien avec Diffie-Hellman. Pour résoudre le problème du logarithme discret nous avons également étudié l'algorithme « Baby step Giant step » mis en place par Daniel Shanks. Enfin nous avons étudié un protocole d'attaque pour le principe de chiffrement par transpositon.

Journal de bord :

11/12/2018 : - Choix du sujet (Cryptographie)

- Recherche sur le principe de Diffie-Hellman
- Recherche sur le chiffrement par transposition
- Début de codage du chiffrement par transposition
- Début de codage du principe de Diffie-Hellman

21/12/2018 : - Poursuite du codage

- Recherche sur les modèles mathématiques

07/01/2019 : - Poursuite du codage

- Début du rapport
- Début du diaporama final

09/01/2019 : - Finalisation des codes

- Poursuite du rapport
- Poursuite du diaporama

Conseils pour Alice et Bob :

Alice et Bob veulent choisir entre s'envoyer une clé K en l'encodant à l'aide d'un chiffrement par transposition ou en utilisant le principe d'échange de Diffie-Hellman.

Tout d'abord, pour utiliser le principe d'échange de clés de Diffie-Hellman, il faut que les deux agents se mettent d'accord sur un nombre sans que quiconque puisse découvrir ce nombre même en écoutant leurs échanges. Pour utiliser ce principe, il faut donc obligatoirement que les deux agents soient en accord sur un nombre commun sinon le chiffrement ne fonctionnera pas. Cette méthode présente l'avantage d'assurer la sécurité rétroactive, c'est-à-dire qu'en cas de divulgation d'un élément secret ou privé du système, seul l'échange en cours est affecté, mais pas les échanges précédents. Autrement dit une troisième personne qui aurait enregistré pendant une certaine période les chiffrés échangés, ne peut pas remonter dans le temps. Elle est très efficace comme système de cryptographie.

Ensuite, le principe d'un chiffrement par transposition consiste à changer l'ordre des lettres, donc à construire des anagrammes. Pour de très brefs messages, comme un simple mot, cette méthode est peu sûre car il n'y a guère de variantes pour redistribuer une poignée de lettres. Une transposition au hasard des lettres semble donc offrir un très haut niveau de sécurité. Néanmoins, pour que la transposition soit efficace, l'ordonnancement des lettres doit suivre un système rigoureux sur lequel l'expéditeur et l'envoyeur se sont préalablement entendus.

Les deux protocoles de partage possèdent un niveau de sécurité élevé et ils sont donc tous deux très intéressant à utiliser pour un partage de clé standard. Néanmoins, le protocole de chiffrement par transposition semble être plus intéressant et sécurisé à utiliser que le protocole de Diffie-Hellman. Alice et Bob devraient donc utiliser le principe de chiffrement par transposition pour leur partage de clé.

Si Alice et Bob ne s'étaient pas connu à l'université, ils n'auraient pas pu connaître la clé que Bob avait en tête car ils doivent avoir la même pour que la méthode proposé par Bob fonctionne. Même idée pour la méthode d'Alice.

Les modèles mathématiques :

Echange de clés Diffie-Hellman

(Réfléchir sur l'impact des choix de grands nombres)

L'échange de clés Diffie-Hellman est une méthode par laquelle deux agents, nommés par convention Alice et Bob, peuvent se mettre d'accord sur un nombre (qu'ils peuvent utiliser comme clé pour chiffrer la conversation suivante) sans qu'un troisième agent appelé Ève puisse découvrir le nombre, même en ayant écouté tous leurs échanges.

Principe

- Alice et Bob ont choisi un groupe fini (soit un corps fini, dont ils n'utilisent que la multiplication, soit une courbe elliptique) et un générateur g de ce groupe (ils peuvent aussi ne décider de ce choix qu'au moment de l'échange, et de ce le communiquer en clair, ce qui n'améliore pas les chances d'Ève) ;

- Alice choisit un nombre au hasard a, élève g à la puissance a, et dit à Bob g^a (calculé dans le groupe ; si par exemple ils travaillent dans le corps fini $\mathbf{Z}/p\mathbf{Z}$, ils échangeront les nombres modulo p) ;

- Bob fait de même avec le nombre *b* ;
- Alice, en élevant le nombre reçu de Bob à la puissance *a*, obtient g^{ba} (toujours calculé modulo *p* par exemple).
- Bob fait le calcul analogue et obtient g^{ab}, qui est le même ; mais puisqu'il est difficile d'inverser l'exponentiation dans un corps fini (ou sur une courbe elliptique), c'est-à-dire de calculer le logarithme discret, Ève ne peut pas découvrir, donc ne peut pas calculer g^{ab} [mod *p*] ;
- Finalement, Alice et Bob connaissent donc tous les deux le nombre g^{ab} [mod *p*] dont Ève n'a pas connaissance.

Problème du logarithme discret

Le problème du logarithme discret est celui de la résolution de l'équation:

$$a^x = b \bmod n$$

d'inconnue x où a, b, n sont des entiers donnés.

Si l'on connait a, x, n il est facile de calculer b, mais il est difficile de retrouver x en ne connaissant que a, b, n.

Dans le cadre du principe de Diffie-Hellman, même si un espion, Damien, intercepte les clés lors des échanges, il ne pourra obtenir que g^a mod *p* et g^b mod *p*. Or, comme indiqué plus haut, ici il ne sera pas aisé de retrouver *a* et *b*, les nombres secrets d'Alice et de Bob, pour obtenir g^{ab}, la clé secrète de chiffrement, à cause des modulos *p*, pour peu que p soit assez grand. Actuellement, mêmes les meilleures algorithmes seraient incapables de casser le code avec un nombre premier *p* d'environ 300 chiffres, et de nombres *a* et *b* d'environ 100 chiffres.

Algorithme Baby-Step Giant-Step

L'algorithme Baby-Step Giant-Step est un algorithme permettant la résolution du problème du logarithme discret, mis au point par Daniel Shanks en 1971.

Soit G un groupe cyclique de générateur α d'ordre n, dont la loi est noté multiplicativement. Le problème du logarithme discret revient à chercher, pour β dans G, un entier x tel que $\alpha x = \beta$. La méthode naïve serait d'essayer successivement les entiers à partir de 0 jusqu'à trouver l'entier x solutions, ce qui peut demander n essais dans le pire des cas, un temps exponentiel en la taille de n. Par division euclidienne par un entier m : $x = im + j$ avec $0 \leq j < m$ et $0 \leq i \leq n/m$. On a alors que $\alpha x = \beta$ si et seulement si $\alpha j = \beta(\alpha - m)i$. L'algorithme baby-step giant-step utilise ceci pour un m bien choisi, le plus souvent $m = \lceil \sqrt{n} \rceil$: pour trouver l'entier x on calcule la liste des $(j, \alpha j)$ (les baby-steps), puis les $(i, \beta(\alpha - m)i)$ (les giant-steps) jusqu'à trouver un second membre déjà présent dans la liste des baby-steps, le couple (i,j) correspondant donne x.

L’attaque de l’homme du milieu

Problème

L'une des faiblesses qui peut affecter la cryptographie utilisant une clé publique intervient au moment de la distribution de cette même clé. L'idée la plus simple est de les distribuer dans un annuaire. Cependant, il faut pouvoir garantir la fiabilité des informations contenues dans cet annuaire.

Imaginons en effet qu'Alice veuille communiquer à Bob, mais que Damien arrive à se faire passer pour Alice aux yeux de Bob, et pour Bob aux yeux d’Alice. Il envoie à Alice sa propre clé publique, qu'Alice pense être la clé publique de Bob. Lorsqu'elle chiffre son message, elle utilise donc la clé publique de Damien. Celui peut intercepter le message, le déchiffrer avec sa clé privée, le modifier à sa guise éventuellement, puis le renvoyer à Bob en utilisant la clé publique de Bob. On appelle cette attaque l'attaque de l'homme du milieu.

Solution

Les certificats numériques

Ainsi, lorsqu'Alice envoie un message en utilisant la clé publique de Bob, elle a besoin de savoir que cette clé appartient effectivement à Bob et à personne d'autre. Comme dans la vie courante, on a recours à des certificats. Pour passer un examen, il vous faut prouver votre identité, fournir une carte d'identité, passeport ou permis de conduire. Un organisme supérieur (l'Etat) a signé ces certificats, s'assurant auparavant (par un acte de naissance,...) qu'il s'agit bien de vous.

Les certificats numériques fonctionnent sur le même principe. Pour les gérer, on fait appel à une **autorité de certification**, qui gère des infrastructures de clés publiques (ou parle aussi de **PKI**, pour Public Key Infrastructure). Ces autorités de certification délivrent des certificats, qui contiennent plusieurs informations dont les plus importantes sont :

- le nom du propriétaire du certificat;
- sa date de validité;
- le système cryptographique associé;
- la clé publique associée;
- la signature de l'autorité de certification, qui doit garantir à la fois la justesse des informations du certificat, et leur origine.

Ainsi, si Bob ne veut pas qu'on puisse usurper son identité, il confie sa clé publique à une autorité de certification. Celle-ci doit vérifier que les informations que lui fournit Bob sont correctes, et lui délivre un certificat qui garantit son identité. Si Alice veut envoyer un message à Bob, elle récupère sa clé publique non pas auprès de Bob directement, mais auprès de l'autorité de certification, qui agit comme un tiers de confiance. En utilisant la clé publique du certificat, Alice est sûre que personne n'a usurpé l'identité de Bob.

Bien sûr, tout n'est pas parfait dans ce système, car la sécurité repose sur un tiers extérieur.

Le chiffrement par transposition

Principe

Les méthodes de cryptographie par transposition sont celles pour lesquelles on chiffre le message en permutant l'ordre des lettres du message suivant des règles bien définies. Autrement dit, on produit une anagramme du message initial.

Du fait qu'on ne change pas les lettres du message initial, on pourrait imaginer que ces procédés de chiffrement ne sont pas sûrs du tout. C'est effectivement le cas si on chiffre de petits messages, comme des mots, où le nombre d'anagrammes est très réduit. Mais dès que l'on s'intéresse à des messages assez grands, le nombre de transpositions possibles est extrêmement grand, et il est impossible de tester toutes les permutations possibles.

Cela dit, il faut que l'expéditeur et le destinataire se mettent d'accord sur une façon de permuter les caractères de façon assez régulière pour qu'elle puisse s'appliquer à n'importe quel message. C'est ce choix qui va rendre le chiffrement par transposition plus ou moins résistant aux attaques.

Cryptage

Pour effectuer un chiffrement par transposition rectangulaire, on commence par se mettre d'accord sur un mot-clé. Choisissons pour notre exemple le mot BIBMATH. On classe alors les lettres du mot BIBMATH par ordre alphabétique, et on attribue à chaque lettre son numéro dans l'ordre alphabétique. Ainsi, on donne à A le numéro 1, au premier B le numéro 2, au deuxième B le numéro 3, au H le numéro 4, etc….

On crée ensuite un tableau de la façon suivante :

- la première ligne est constituée par les lettres de la clé;
- la deuxième ligne est constituée par les numéros qui leur sont associés;
- on complète ensuite le tableau en le remplissant avec les lettres du message à chiffrer. On écrit sur chaque ligne autant de lettres que de lettres dans la clé. Eventuellement, la dernière ligne n'est pas complète.

Ensuite, on écrit d'abord le contenu de la colonne numérotée 1, puis le contenu de la colonne numérotée 2, etc…

Décryptage

Pour faire l'opération inverse (déchiffrer), il faut d'abord reconstituer pour chaque colonne le nombre de lignes que le tableau comprenait. Pour cela, on note n le nombre de lettres du message et c le nombre de lettres de la clé, qui est aussi le nombre de colonnes du tableau de chiffrement. Si n est un multiple de c, alors on a affaire à un tableau où toutes les colonnes ont la même hauteur, qui vaut n/c. Sinon, on note q le quotient dans la division euclidienne de n par c, et r le reste. Il y aura alors r colonnes (les premières) qui auront pour hauteur q+1, et c−r colonnes (les dernières), qui auront pour hauteur q.

Ensuite, on remplit le tableau en écrivant dans la colonne numérotée 1 les premières lettres du message, puis dans la colonne numérotée 2 les suivantes, et ainsi de suite… Le message clair se lit alors directement sur le tableau.

Attaque du cryptage

Attaquer un cryptage par transposition peut s'avérer complexe. En effet, utiliser une clé assez longue dans le cas d'un message long ou une clé courte dans le cas d'un message court permet de minimiser les chances pour un potentiel attaquant de briser le code, puisqu'aucune informations sur la clé ne sont données dans le message (longueur par exemple). Ainsi, une clé transmise par voie orale, ou par toute autre voie de communication sécurisée, et correspondant bien à la taille du message à crypter sera virtuellement impossible à craquer ou alors très difficile.

Conclusion

Difficultés rencontrées

La première difficulté rencontrée fut liée à la compréhension des principes de cryptographie. En effet, bien qu'au final assez simples, nous avons dû trouver plusieurs sites où regarder et surtout comprendre comment marchait ces principes.

Une autre difficulté a été la compréhension du logarithme discret, et plus particulièrement de l'algorithme « Baby-Step Giant-Step », qui n'a pas été totalement compris par tous les membres du groupe.

De plus, nous n'avons pas trouvé de vrai angle d'attaque sur la cryptographie par transposition, puisque si la clé est transmise par voie sécurisée, récupérer cette clé est plutôt difficile pour une personne extérieure.

Apports du projet

Ce projet nous a permis d'appréhender deux façons de crypter une clé et de l'échanger avec d'autres personnes, choses qui sont très intéressantes et utiles dans le monde d'aujourd'hui. De plus, il nous a aussi permis d'apprendre le problème du logarithme discret, utilisé lors de cryptage par clefs publiques.

Sources

Wikipédia
Bibmath