Initiation à la cryptographie

Correction TD de cryptographie n°1

—TELECOM Nancy 2A Formation par Apprentissage—

1 Substitutions

▸ **Exercice 1.** Chiffrement par décalage (César)

1. Chiffrer le message “la rencontre est prévue à la cafétéria” à l'aide du chiffrement par décalage et de la clé $K = 5$.
2. Décrypter le message “RGNEIDVGPEWXTRAPHHXFJT” sachant qu'il a été créé par un chiffrement par décalage.
3. Dans un texte en français les lettres les plus fréquentes sont le A (8.4%) et le E (17.26%). Sachant que le message est en français, chiffré en utilisant le chiffrement par décalage sur les 26 lettres de l'alphabet, déterminer la clef et décrypter le début du message :
   SVOXFYIKNKXCVKVSQEBSOKMRODOBNOCCYVNKDC

✂ -

Correction :

1. *QF WJSHTSYWJ JXY UWJAZJ F QF HFKJYJWNF*
2. *On obtient le message clair : ”cryptographie classique” par un décalage de 15.*
3. *L'analyse des fréquences d'apparition des lettres dans le message chiffré montre que ce sont les lettres K et O les plus fréquentes. Ils correspondent donc probablement aux lettres A et E, et on obtient un décalage de 10. Le texte clair donne :*
   il envoya dans la ligurie acheter des soldats.

- ✂

▸ **Exercice 2.** Chiffrement par substitution

1. Chiffrer le message “la rencontre est prévue à la cafétéria” à l'aide du chiffrement par substitution et de la clé suivante :

a	b	c	d	e	f	g	h	i	j	k	l	m
X	N	Y	A	H	P	O	G	Z	Q	W	B	T

n	o	p	q	r	s	t	u	v	w	x	y	z
S	F	L	R	C	V	M	U	E	K	J	D	I

2. Est-il possible de décrypter le message “YHVMQUVMH” chiffré par un chiffrement par substitution sans connaître la clé ? Déchiffrer ce message sachant qu'il a été créé avec la clé précédente.

✂ -

Correction :

1. *On obtient le message chiffré : BX CHSYFSMCH HVM LCHEUH X BX YXPHMHCZX*
2. *Le message est trop court pour qu'une analyse fréquencielle donne suffisamment d'information. En connaissant la clé en revanche on obtient le message clair : ”c est juste”*

- ✂

▸ **Exercice 3.** Chiffrement de Vigenère

1. Chiffrer le message "la rencontre est prévue à la cafétéria" à l'aide de la méthode de Vigenère et du mot clé POULE.
2. Est-il possible de décrypter le message "BAUNBEKLZLQSKQKEBGCJYHVSKR" chiffré par un chiffrement de Vigenère sans connaître la clé ? Déchiffrer ce message sachant qu'il a été créé à l'aide du mot clé TNCY.

Ici la lettre "A" correspond à un décalage de 0.

✂ -

Correction :

1. *Le message chiffré est : AO LPRRCHEVT SME TGSPFI P ZU NEUSNPVXO*
2. *Encore une fois, le message est trop court pour porter suffisamment d'information. Il existe une méthode pour décrypter un texte chiffré par le chiffrement de Vigenère, mais celui-ci demande au moins plusieurs phrases consécutives. Ici, avec la clef TNCY nous retrouvons le message clair : "inspiringyourdigitalfuture".*

- ✂

2 Protocoles

▸ **Exercice 4.** Message confidentiel et authentifié

Alice doit envoyer un message confidentiel et authentifié à Bob, mais ne dispose que d'un canal public. Elle utilise le protocole suivant :

— Bob envoie sa clé publique à Alice.
— Alice envoie sa clé publique à Bob.
— Alice produit son message, le signe avec sa clé privée, et le chiffre avec la clé publique de Bob.
— Bob reçoit le message, le déchiffre avec sa clé privée, et vérifie que la signature colle avec la clé publique d'Alice.

Où est le lézard ?

✂ -

Correction : *Dans ce schéma qui semble à première vue parfait, il y a un élément crucial qui n'a pas été pris en compte, et ce dès le début : Bob (resp. Alice) n'a aucune certitude qu'il parle bien à Alice (resp. Bob). Un attaquant, disons Charlie, peut alors se créer lui aussi une paire de clés publique / privée. Il intercepte la clé publique de Bob et envoie la sienne à Alice, sans laisser transiter celle de Bob. De même il intercepte la clé publique d'Alice et envoie la sienne à Bob. Dans toute la suite des échanges, il peut déchiffrer les messages qu'il reçoit (puisque chiffrés avec sa clé publique), et les rechiffrer pour le bon destinataire, sans que ces deux là ne se rendent compte de rien. Cette attaque est appelée l'attaque* de l'homme au milieu *ou man in the middle en anglais*

Pour l'éviter, il faut par exemple que les clés publiques soient certifiées par une tiers authorité, ou que Bob et Alice trouvent un autre moyen de vérifier leurs clés publiques (attestées par un autre correspondant en lequel ils ont confiance au préalable, vérification occullaire derrière un écran etc).

- ✂

▸ **Exercice 5.** SSH

Dans le livre *SSH, the secure shell*[1], les auteurs décrivent la phase d'identification d'un client

1. de D.J. Barrett & R.E. Silverman, publié chez O'Reilly en 2001

SSH auprès d'un serveur SSH.

La suite d'échanges suivante correspond à la conversation entre un client et un serveur :

(*i*) le client : ≪ Bonjour serveur, je voudrais obtenir une connexion SSH sur un de vos comptes, plus particulièrement sur le compte dénommé SMITH. ≫

(*ii*) le serveur répond : ≪ Eh bien, pourquoi pas. Je vais tout d'abord vous proposer un défi pour m'assurer de votre identité ! ≫ Le serveur envoie alors des données, correspondant au défi, au client.

(*iii*) le client répond : ≪ J'accepte le défi. Voici la preuve de mon identité. Je l'ai calculée moi-même à partir de votre défi et de ma clé privée. ≫ Cette réponse au serveur est appelée un *authentificateur*.

(*iv*) le serveur répond : ≪ Merci. Je vais maintenant examiner le compte SMITH pour voir si vous pouvez vous connecter. ≫ Plus précisément, le serveur examine les clés publiques de SMITH pour vérifier si l'une d'elle correspond bien à l'authentificateur que le client a donné Si c'est le cas, le serveur répond alors : ≪ OK, vous pouvez accéder à votre compte ≫ ; sinon : ≪ L'authentification a échoué. ≫

Nous allons identifier les différentes primitives cryptographiques utilisées dans ce programme.

1. Décrivez de manière plus précise les opérations intervenant dans ces différentes phases de l'identification. En particulier, quel(s) type(s) de système(s) cryptographique(s) permet(tent) de réaliser ce protocole ? Montrez avec un exemple le déroulement de ce protocole.
2. Avant qu'une telle connexion entre un client et un serveur SSH ne puisse s'établir, qu'est-il nécessaire de mettre au point au préalable ?
3. Si le protocole n'avait pas précisé clé privée, clé publique, quel autre mécanisme pourriez-vous proposer ? Commentez.

✂ -

Correction :

1. *Le protocole utilisé est un protocole de type défi/réponse utilisant un algorithme de signature. L'utilisation des expressions* clé privée *et* clé publique *doit tout de suite renseigner sur le fait que c'est de la cryptographique à clé publique (dite encore asymétrique) qui est utilisée. Plus précisémement, le serveur envoie un défi au client. le client signe ce défi en utilisant sa clé privée (connue de lui seul). Il envoie cette signature au serveur qui vérifie que cette signature a bien été produite par le possesseur de la clé privée correspondant à la clé publique stockée chez lui ainsi qu'au défi qu'il a envoyé. (Le calcul de la valeur v correspond simplement à la réponse binaire : la signature est bonn ou mauvaise)*
2. *Au préalable, il faut que Smith ait pris le soin d'enregistrer sa clé publique auprès du serveur de manière sûre c'est-à-dire grâce à un canal authentifié. En pratique, cela peut se faire lorsqu'il se trouve dans le réseau interne où se trouve le serveur. Smith doit également être sûr qu'il s'adresse au bon serveur. Ce problème est partiellement réglé par l'enregistrement de la clé publique du serveur par le client de Smith ;* `ssh` *affiche un warning en cas de changement de clé publique. Ce système ne fournit absolument pas une sécurité totale, il vise simplement à améliorer la sécurité des protocoles de type* `telnet`*,* `rlogin`*, etc.*
3. *On pourrait se servir d'un mécanimse de MAC à la place d'un mécanisme de signature. Mais dans ce cas le serveur connaît la clé secrète du client (et de tous les autres clients), ce qui impose que des contraintes de sécurité plus fortes sur le serveur.*

- ✂

3 Se familiariser avec les ordres de grandeur

▸ **Exercice 6.** Mot de passe

Un système est protégé par un mot de passe. Après un essai infructueux le système attend 1 seconde avant de redemander le mot de passe. Combien de temps faudra-t-il pour pénétrer le système dans les cas suivants :

1. le mot de passe est un prénom ;
2. c'est un mot du dictionnaire ;
3. il est composé de 4 chiffres ;
4. il fait 8 caractères alphanumériques (y compris les 15 signes de ponctuations)

✂ -

Correction :

1. *L'INSEE publie la liste des 20 000 prénoms donnés en France depuis 1946. En pratique, seul un millier de prénoms suffit à désigner plus de la moitié de la population française. Il faudrait ainsi, en moyenne moins de 17 minutes et dans le pire des cas (prénom très rare) moins de 5 heures et 30 minutes pour retrouver le mot de passe.*
2. *Le français compte environ 200 000 mots dont seulement 3 000 sont utilisés couramment, soit donc 2 jours et 8 heures au maximum et vraisemblablement moins de 50 minutes.*
3. *Il y* $10^4 = 10000$ *mots de passe différents constitués de 4 chiffres, ce qui représente 2 heures et 45 minutes pour tous les tester.*
4. *Si l'on s'autorise les minuscules, les majuscules, les chiffres et quinze signes de ponctuations il faut :*

$$77^8 \ s \ \approx 4 \cdot 10^6 \text{ années.}$$

- ✂

▸ **Exercice 7.** La force brute

Le *facteur de travail* d'un algorithme est le nombre d'instructions élémentaires nécessaire à son exécution. La puissance d'une machine est le nombre d'instructions qu'elle exécute par unité de temps. Nous allons approcher la puissance d'un PC actuel à environ 2000 Mips (millions d'instructions par seconde).

Le facteur de travail d'un algorithme optimisé pour tester une clé de 128 bits de l'algorithme AES est d'environ 1200 instructions élémentaires.

On dispose d'un couple clair/chiffré connu et on désire retrouver la clé utilisée par force brute, c'est-à-dire en testant toutes les clés les unes après les autres. Une clé est constituée d'un mot de 128 bits. On suppose que toutes les clés sont équiprobables.

1. En combien de temps une machine de 2000 Mips teste-t-elle une clé ?
2. Combien y a-t-il de clés possibles ? Quel est le nombre moyen de clés à tester avant de trouver la bonne ?
3. À quel temps moyen de calcul cela correspond-il si on suppose qu'un seul PC effectue la recherche ? Si les 1 milliard de PC de l'Internet sont mobilisés à cette tâche ?

✂ -

Correction :

1. $t = \frac{\text{facteur de travail}}{\text{puissance}} = \frac{1200}{2000} = 0,6\mu s.$
2. *Nbre de clés possibles* $= 2^{128}$. *On considère les clés possible comme étant les entiers de* 0 *à* $2^{128}-1$, *et la clé secrète est notée* k. *On a deux scenarios d'attaque par force brute possible. On note* $n = 2^{128}$.

— *Si on essaie tous les entiers les uns après les autres. La probabilité, pour un entier i donné, d'avoir $k = i$ (et donc d'avoir exactement $i+1$ tirages à effectuer si on part de 0), est égale à $\frac{1}{n}$. L'espérance du nombre d'essais est donc :*

$$\sum_{i=0}^{n-1}(i+1)\frac{1}{n} = \frac{n(n+1)}{2n} \approx \frac{n}{2}.$$

— *Si on effectue un grand nombre de tirages aléatoires parmi 2^{128}, on a une loi binomiale. Chaque tirage a une probabilité de succès $\frac{1}{n} = p$. La probabilité qu'on trouve la clé au bout de i tirages est :*

$$(1-p)^{i-1}p$$

. On a donc l'espérance du nombre de tirages nécessaires :

$$\begin{aligned} \mathrm{E} &= \sum_{i=1}^{\infty} i(1-p)^{i-1}p = pf'(1-p), \text{ où } f(x) = \tfrac{1}{1-x}. \\ &= p\frac{1}{(1-(1-p))^2} = \frac{1}{p}. \end{aligned}$$

3. *On use et abuse des approximations* $10^3 = 1000 \approx 2^{10}$, 1jour $= 2^{16}$s, 1an $= 2^9$jour $= 2^{25}$secondes, *etc. On calcule d'abord le nombre d'instructions calculées en un an à la fréquence de* 2000 *Mips.*

$$\begin{aligned} \textit{2000 Mips.années} &\approx 2000 \times 2^{20} \times 2^9 \times 2^{16} \approx 2^{45} \textit{instructions}, \\ &\approx 2^{11+20+9+16} \approx 2^{56}. \end{aligned}$$

Le nombre d'instructions à effectuer pour trouver la clé est : $1200 \times 2^{127} \approx 2^{138}$. *Soit un temps de* $\approx 2^{138-56} \approx 2^{81}$ *années (ou, en base* 10 *:* $2 \times (2^{10})^8 \approx 2 \times 10^{24}$*).*

Les un milliard ($\approx 2^{30}$*) de PC d'Internet permettent de gagner un facteur* 2^{30}*, ou* 10^9*. Soit quelque chose comme* 2×10^{15} *années, soit un petit million de fois l'âge de l'univers.*