Devoir 1 - IFT3275

Philippe Caron

22 février 2016

1 Question 1

1.1 Déchiffrement

Le message a pu être décodé en utilisant une attaque fréquentielle sur le texte chiffré. L'attaque a réévlé que le chiffre n'était pas une substitution monoalphabétique, mais plutôt un chiffre de vigenere avec un clef dont la longueur probable est 25.

Par la suite, il a suffit de trouver le décalage de la lettre la plus fréquente sur l'ensemble des positions $25 * i + k$ avec la lettre 'E', où i vaut de 0 à $L \div 25$, L représente la longueur du document et k vaut de 0 à 25. Par cette méthode on obtient la clef suivante :

LSJFHTFSLMBSVOHBOSVTTMBOE

Ce qui permet de déchiffrer le texte.

1.2 Importance du texte

Sans lire allemand, la recherche de la première phrase sur Google permet de trouver que ce texte est en fait un discours que Hitler a prononcé le 22 juin 1941. Une traduction est disponible ici : `http://research.calvin.edu/german-propaganda-archive/hitler4.htm`. Dans ce discours il annonce que l'Allemagne entrera en guerre contre la Russie, il parle également d'une «attaque au proportions jusqu'à lors inégalées» (traduction libre de "At this moment, an attack unprecedented in the history of the world in its extent and size has begun").

Sachant que l'Allemagne a attaquée l'Union Sovétique le 22 Juin, si ce message a effectivement été reçu le 19 Juin 1941, on ne peut vraiment pas dire qu'il soit sans importance ! En fait, il donne la possibilité d'avertir les Russes de l'attaque imminente (ou de ne pas les avertir du tout). Il permet aussi de se positionner stratégiquement sachant qu'un fort contingent allemand risque de se diriger vers le front Ouest. Des machinations politiques à la stratégie militaire, ce message est une perle rare et ne devrait surtout pas être ignoré.

2 Question 2

Supposons que l'adversaire a accès au canal de transmission, par exemple, il a réussi à installer un dispositif qui sépare le signal du fil reliant la banque et le guichet de sorte qu'il peut recevoir une copie exacte des transmissions entre la banque et le guichet. Il ne connais peut-être pas exactement la manière dont l'information est codée, mais sa connaissance lui permet tout de même de séparé les message qui proviennent de la banque et ceux qui proviennent du guichet. Sinon il est fort probable que l'information soit transmise sur deux fils différents pour l'envoi et la réception, il n'a qu'à identifier ces fils, il peut également espionner le guichet et déduire que l'information qui survient immédiatement après une transaction vient du guichet, il pourrait éventuellement même installer un voltmètre dans son dispositif afin d'identifier le sens du courant.

Tout cela n'a guère d'importance cependant, l'important est qu'il est possible pour l'adversaire de connaitre exactement l'information qui est envoyée du guichet à la banque pour un temps donné et vice-versa, que ce

soit par un canal secondaire ou non. Si le flux n'utilise par d'IV, il lui suffit de copier le comportement du guichet pour une transaction passée. Sans même comprendre ce qu'il fait, il va pouvoir recevoir de la banque le même résultat qu'à la transaction précédente, exemple : un retrait de 20$. Puis il peut répéter en volant des transactions jusqu'à vider le guichet. En volant des petits montants, 20$ par personne par exemple, il y de bonne chances que ni la banque, ni les clients ne se rende compte de la supercherie avant très longtemps. Il est toutefois très probable que le flux soit initialisé aléatoirement à chaque fois, sans quoi la sécurité est serait vraiment très réduite. À ce moment, il aurait l'option de recueillir les message envoyés par le gichet et tenter de comprendre le fonctionnement du système en effectuant la même transaction à répétition. Au nombre de retrait de 20$ qui sont faits à chaque jour, il ne risque pas de prendre trop de temps avant d'accumuler un répertoire de textes chiffrés assez considérable. Plus haut, on parlait de mimer les répones du guichet, mais on peut aussi mimer le comportement de la banque, c'est d'ailleurs favorable pour un coup rapide car aucune limite de 200$ par jour n'est applicable si la banque n'est pas au courant des retrait. À chaque transaction le guichet demande une autorisation et la banque doit répondre par une forme de ≪OUI≫ ou ≪NON≫, si cette réponse n'est pas trop longue, l'adversaire peu faire une attaque exhaustive sur le guichet. Il peut rajouter des bits au messages, essayer de déclencher des erreurs en envoyant des messages aléatoires.

3 Question 3

Le message m envoyé par Alice est authentifié par RSA comme suit :

$$D(m) = m^d \pmod{N} \tag{1}$$

Pour le décrypter, Bob qui possède e et N fait :

$$E(m) = m^e \pmod{N} \tag{2}$$

A priori, Bob n'a que récupéré le message, mais ce message contient plus d'information qu'il n'en parait. Dans le cas qui nous intéresse, $PGCD(m, N) \neq 1$. Or on sait que $N = pq$ et que p et q sont premier. Donc forcément, disons que $m = p$, $N = mq$. Quand bob recevra le message authentifié par Alice, il aura m en sa possession, puisqu'il a déjà N, il peut déduire q en faisant $q = N \div m$. Il aura alors p et q, à ce moment, déduire d sera trivial puisque :

$$d = e \pmod{(p-1)(q-1)}^{-1} \tag{3}$$

Bob aura alors en sa possession les clefs publique et privée $((N, e), d)$ ce qui permet à Bob de lire tout ce qu'Alice lit, et de signer des message en son nom sans qu'il soit possible pour les autres de les différencier. Ce cas précis constitue évidemment une faille du système, mais est-ce vraiment une faiblesse dans la sécurité de RSA ?

Si l'on considère qu'Alice ne fait que transmettre des informations à Bob par rapport à un sujet quelconque, par exemple la hauteur de la neige dehors. Elle ne vérifie jamais la valeur de m par rapport a p et q puisque cette comparaison n'a aucun intérêt pour elle. Il est alors possible que $m = p$, puisque la valeur de m est déterminé par quelque chose d'aléatoire (dans ce cas-ci la hauteur de la neige, mais n'importe quel transmission qui n'est pas organisée risque de comporter une forte composante aléatoire) alors Bob a autant de chances de trouver p en regardant les valeurs de m qu'en devinant des nombres premiers au hasard. Au final, cette faiblesse a tellement peu de chance de se produire qu'elle n'en est pas vraiment une.

Elle pourrait être comparée à la probabilité qu'une clef XOR ne soit composée que de 0, ce qui pour un générateur totalement aléatoire est équiprobable à n'importe quelle autre clef. Malgré cette éventualité, la probabilité est si faible que ce n'est pas réellement considérer comme une faiblesse.

4 Question 4

Voici la clef RSA générée :

——————————— RSA PUBLIC KEY ———————————

N=

612675321359862015769772390031646003276117188974258244252414092148800815455643060410968385019240328243882197127186121816251724231920196955039985311559325838817206026738633520287114675534871381747268305069363538859557647343283159373314470658143717320037554745654036987665432681420717368766312060148279202080480555367852941901417331095289964600984510573517210000268788724268083360562808115378066702890205203701829067280920412907030341357078409255361986757381049435813699305158662941419831548046820733688652478943005243276669099142065853807689050404937099659741985260677338180178496974182033749419191328473443896183187 9

e=

683929

——————————— RSA PRIVATE KEY ———————————

d=

208834683521321296275316413120305440441529736753459833428047762252971790493823805583804418079150531416083110320930802890429027656955181246229909139699327208228843495984764096150876402945393873129647495301968723224220347234816562361627911711256255648001173860743608658515103682963282569484950354456631779574619524014415892056178544915504552888835195191701845158162406808568627251741065488180052577876945907680981851355217668286498134875928067890292645714880079196140133509139553779288901752463875407294436738004453904240313631265293644662906662197424258618782271728976262378137923784784090542653905491300549706316137

═══

p=

508098095802397509840983948104981039480193740193409132750193840913049710239470139774019374091374091374091374091734091730491730497120937401974031913904701937403197409137409137409137409137409173409198095720982043923795748750924750947507290902013909317509117305917057103947103957100984609382908502980248950973458785727405827835720872038729703013802975278906967

q=

120582093580239520395723095723095702397502397509327502937502937502937502393750293750392750239750239750239750239750293750293754029384023984029357029751392085905270359840293750898230937280927328357209385702325224232187058293049237407932895720489037508425209375233 7

5 Question 5

Si nous n'avions que la clef $k_1 \in K$ à déduire, il suffirait de faire une attaque exhaustive en essayant toutes les clefs, du genre

$$\text{DECODE}_{k_1}(c) \stackrel{?}{=} m$$

Cette attaque nécessite d'évaluer DECODE un maximum de $|K|$ fois, avec une espérance de $\frac{|K|}{2}$. Mais on ne possède pas $\text{CODE}_{k_1}(m)$, rendant impossible cette approche. Cependant ont peut mettre cette expression en relation avec la seconde clef :

$$c = \text{CODE}_{k_2}(\text{CODE}_{k_1}(m))$$
$$\text{DECODE}_{k_2}(c) = \text{CODE}_{k_1}(m)$$

À ce moment, si on suppose que pour tout x, $\text{DECODE}_{k_x}(c)$ retourne une valeur différente, on peut essayer successivement toutes les clefs k_1 de l'ensemble K pour chacune des clefs k_2 du même ensemble, ce qui revient à élavuer CODE $|K|^2$ fois, et DECODE $|K|$ fois (ou l'inverse dépendamment de quelle clef on met dans la boucle) donc $|K|^2+|K|$ évaluations au total. Puisqu'on a présumé qu'il n'y avait pas de recoupement, au moment ou $\text{DECODE}_{k_2}(c) = \text{CODE}_{k_1}(m)$ on sait que k_1 et k_2 ont les bonnes valeurs. Pour accélérer le processus, si cela vaut la peine, on peut d'emblée rejeter les éléments qui ne sont ni dans M ni dans C, puisque par définition $\text{DECODE}_{k_2}(c) = \text{CODE}_{k_1}(m) \in (M \cap C)$.