Université Claude Bernard Lyon 1
Institut de Science financière et d'Assurances (ISFA)

50 avenue Tony Garnier
69007 Lyon, France

Master 1 informatique
Université Claude Bernard Lyon 1

Cryptologie : TP noté

— *Ce TP noté se fait en binôme. NOM1 et NOM2 sont les noms de famille des membres du binôme.*

— *Les réponses aux questions théoriques (marquées* **t***) seront rédigées dans un fichier à part appelé « NOM1_NOM2.txt »(ou pdf si vous faites du LaTeX). Vous pouvez aussi y faire apparaître les résultats des attaques et les temps de calculs demandés.*

— *Votre code (qui correspond aux questions marquées* **p***) est à nous rendre par email :*
`gerald.gavin@univ-lyon1.fr` et
`ida.tucker@ens-lyon.fr`.

Pour nous simplifier la correction, vous ne nous enverrez que 2 fichiers (un pour chaque exercice) qui devront compiler/s'interpréter.

Vous nous enverrez une archive portant le nom `NOM1_NOM2.tar.gz` (ou zip) contenant les fichiers correspondants à chacun des deux exercices et le fichier de réponse aux questions.

Le sujet *du mail sera « [TP noté MIF29 2019] ».*

— **La note tiendra compte du respect des consignes.**

— *Le cours concerné se trouve ici :*
`http://perso.ens-lyon.fr/fabien.laguillaumie/teaching/M1_Lyon1_Algo_Crypto.pdf`

Un fichier challenge `TP_challenges.txt` vous est fourni avec le sujet du TP. Il comporte un challenge pour l'exercice 1 et deux challenges pour l'exercice 2.

Exercice (RSA pour paranoïaques)**.** Le cryptosystème RSA est souvent utilisé pour transmettre des clés secrètes (typiquement de 256 bits) de façon sécurisée. Dans ce cas particulier où les messages sont assez petits, il existe la variante exotique suivante.

— La génération de clé se déroule ainsi : deux nombres premiers sont générés. Le premier, p fait 500 bits et le second, q, fait 4500 bits. Le module RSA $N = pq$ fait alors 5000 bits.
Un (petit) exposant e, compris entre 20 et 100, est choisi de façon uniforme, puis (lorsqu'il existe) son inverse d modulo $\varphi(N)$ est calculé. On note $d_p = d \mod p - 1$.

— Le chiffrement se fait de la façon classique, mais les messages ne doivent pas dépasser 300 bits.

— Le déchiffrement se fait en utilisant la technique des restes chinois, sauf que la partie « modulo q » est inutile à évaluer. En effet, le calcul de $(c \mod p)^{d_p} \mod p$ donne directement m.

Q1 **t** Justifiez que le déchiffrement fonctionne effectivement.

Q2 **p** Implantez cette variante de RSA dans le langage de votre choix (vous devez avoir 3 méthodes : KeyGen, Encrypt et Decrypt).

Q3 **p** Utilisez votre méthode de chiffrement pour calculer le chiffré du message m du fichier challenge.

Q4 **p** Mesurez le temps de calcul du déchiffrement sur plusieurs exemples et comparez avec un déchiffrement RSA classique pour un module de 2048 bits.

Une attaque est possible si Bob renvoie à Alice un message que celle-ci a truqué. En effet, si Alice chiffre avec la clé publique de Bob un message m qui est en fait plus grand que p, le déchiffrement renverra un message $\tilde{m}$ différent de m. Supposons que Bob s'attende à ce que le contenu du message d'Alice soit une clef secrète pour un chiffrement symétrique (AES par exemple) pour discuter de façon confidentielle avec Alice. Alors Bob pourrait constater que cette clef secrète ne fonctionne pas, et renvoyer à Alice un message pour vérification du type « Tu m'as envoyé cette clé $\tilde{m}$ qui ne fonctionne pas, c'est normal ? ».

Q5 **t** Après avoir justifié que $\tilde{m} = m \mod p$, déduisez une attaque qui retrouve la clé secrète de Bob.

Q6 **p** Implantez cette attaque et testez-la sur des exemples de votre choix.

Q7 **p** Appliquez votre attaque sur le fichier challenge.

Exercice (Cryptanalyse RSA avec messages reliés)**.** On suppose que Bob chiffre successivement avec RSA pour Alice deux messages de la forme m et $m+\delta$. Les messages chiffrés correspondants sont $c1$ et $c2$ (comme dans le fichier qui contient les challenges).

Q1 **t** Combien vaut :

$$\frac{(m+1)^3 + 2m^3 - 1}{(m+1)^3 - m^3 + 2} \mod N?$$

Dans les questions suivantes, vous allez en déduire une attaque que vous programmerez en vérifiant qu'elle fonctionne sur un exemple de votre choix, puis vous lancerez la cryptanalyse sur le challenge du fichier `TP_challenges.txt`.

Q2 **p** Sachant que l'exposant l'exposant public d'Alice est $e = 3$, et que $\delta = 1$, retrouver le message m en clair (un BigInteger), à l'aide des chiffrés $c1$ et $c2$ présent dans le fichier des challenges – DELTA = 1.

Q3 **t & p** Sachant que l'exposant l'exposant public d'Alice est $e = 3$, et que $\delta = 5$, retrouver le message m en clair (un BigInteger), à l'aide des chiffrés $c1$ et $c2$ présent dans le fichier des challenges – DELTA = 5.