Agnon Kurteshi, Zabiullah Ahmadi, Dawid Dymarczyk
Groupe 12

h e p i a
Haute école du paysage, d'ingénierie et d'architecture de Genève

DÉCODAGE DU RSA G12

Par les agents secrets :

Ahmadi Zabiullah,

Agnon Kurteshi,

Dawid Dymarczyk

formant le groupe 12

Agnon Kurteshi, Zabiullah
Ahmadi, Dawid Dymarczyk
Groupe 12

h e p i a
Haute école du paysage, d'ingénierie
et d'architecture de Genève

Introduction :

Dans le cadre d'une mission secrète nous avons intercepté un message potentiellement dangereux pour la Suisse qui a été codé par le chiffrement RSA, le message indiquait aussi la clé publique qui s'écrit avec le symbole « **e** », et un produit de deux nombres premiers qui est le module de chiffrement des données « **n** ». Le code RSA est quasiment inviolable. Pour le décoder, il faudrait plusieurs milliers d'années. Fort heureusement, pour sauver notre mère Patrie, la Suisse, nous avons trouvé un moyen de le décoder.

Le message:
[**42318337,180709179,249903126,65180209,327950084,99259190,180709179,294437945,99259190,167785286,53556675,280936332,70111939,304943414,9588392,184329508,183935437,180709179,249903126,65180209,215228647,99259190,180709179,294437945,99259190,304943414,200412084,14383554,349273037,265539067,58616223,70111939,338704128,259342028,254511429,338704128,336620996,144543198,41084827,160800993,43242702,162403213,38972580,209096878,201977786,44299941,7494976,184554649,322693638,227046200,119769708,108103468,848083,243188671,337505767]**

La clé publique (n, e) => n=364825199 e=3809

Rappel RSA :

Pour débuter, un rappel de ce qu'est le RSA et ce qui le rend inviolable.

Le chiffrement RSA est un algorithme qui permet d'échanger des données confidentielles sur l'internet. Les fondateurs de ce dernier sont Ronald Rivest, Adi Shamir et Leonard Adleman (les lettres de leur nom de familles forme « RSA ».

C'est un chiffrement asymétrique qui utilise une paire de clés :

- Une clé publique pour le chiffrement « **e,n** »
- Une clé privée pour le déchiffrement « **z,d** »

La clé publique est accessible par tout le monde, mais la clé privée, quant à elle, doit être gardée précieusement par l'auteur des deux clés (sinon tout le monde pourra déchiffrer les messages). Avec juste la clé publique, il est presque impossible de retrouver la clé privée.

Ce qui est magique avec le chiffrement RSA, c'est que dans un sens l'algorithme est facile à calculer, mais dans l'autre sens, il est quasiment impossible de le calculer. Ceci se résume au fait que la multiplication de deux nombres premiers donne un résultat. Dans ce sens, rien de bien compliqué. Mais si l'on veut faire le chemin inverse, cela se complique un peu. Avec des petits nombres, on pourrait à la limite s'en sortir, mais avec des nombres immenses (512 bits), il est quasiment impossible (trop de calcul et de temps) de retrouver les deux nombres premiers qui forment le résultat.

Agnon Kurteshi, Zabiullah
Ahmadi, Dawid Dymarczyk
Groupe 12

h e p i a
Haute école du paysage, d'ingénierie
et d'architecture de Genève

Déchiffrement de la clé Privé

Pour commencer le chiffrement, nous commençons par générer les deux nombres premiers (p et q) qui vont nous permettre de générer le module de chiffrement (n). Dans notre cas, on doit faire le chemin inverse, on avait déjà le « n ». On a dû implémenter une fonction qui nous retourne justement le « p » et le « q » à l'aide de « n ».

D'habitude le module de chiffrement [n] est immenses il fait plus de 617 chiffres en chiffrement RSA 2048, par une chance inouïe dans notre cas, ce nombre **n** = **364825199** ne contient que 9 chiffres !

Il suffit de trouver le premier diviseur de **n** qui est forcément plus petit que la racine(n), soit environ 19000. Avec notre superbe processeur de 3.5Ghz, nous pourrons trouver les facteurs **q** et **p** en une fraction de seconde.

Dans notre cas, p égal à 10'007, et q égal à 36'457

Maintenant que nous avons le « p » et le « q », on peut calculer la valeur indicatrice d'Euler en n que nous utiliserons avec le symbole « **z** ». C'est la multiplication entre « p-1 » et « q-1 ».

On prend (p-1) x (q-1) = z = **364778736** dans notre cas.

Maintenant, on prend le « **e** » qui est la clé publique.

Grace à la méthode d'Euclide étendu vue dans notre formation secrète, nous pouvons facilement trouver notre clé privée qui est nécessaire pour le déchiffrement du message.

L'Euclide étendu, en plus de calculer le PGDC (de a et b), il permet de retourner les coefficients de Bezout (u et v). Ce qui est intéressant, c'est l'égalité suivante au + bv = PGCD(a, b). Lorsque nous avons le cas où a et b sont premiers entre eux, le résultat de u est l'inverse multiplicatif de a modulo b, donc z et e qui donne la clé privée. La clé privée étant l'inverse modulaire de z et e par n.

La clé privée que nous utiliserons sous le symbole « **d »** se calcule en faisant l'inverse de **e** modulo **z,** qui nous donne **d = 90500369.**

Désormais nous avons toutes les clés en main : la paire (**n,e**) clé publique et la paire (**z,d**) la clé privée.

Déchiffrement du message

Un message en RSA est chiffré comme suivant la formule $M^e \equiv C \ mod \ n$.

M est le message, **C** le message chiffré.

Le déchiffrement se fait avec une autre formule $M \equiv C^d \ mod \ n$.

Comme vous pouvez le constater nous allons élever le message à la puissance **d**. Malheureusement pour nous, cela donne un nombre beaucoup trop grand. Notre ordinateur ne supporterait pas un tel calcul, et prendrait surtout trop de temps à l'effectuer. Afin de pouvoir effectuer ce calcul, il faudra qu'on utilise l'algorithme de l'exponentiation rapide. C'est un algorithme qui va nous permettre d'obtenir la puissance d'un nombre beaucoup plus rapidement.

Après décodage nous obtenons ces données brutes.

[6909773, 6955052, 6648615, 6906226, 7741555, 7566703, 6955052, 7741541, 7566703, 11125536, 7696227, 7497076, 7563617, 3026478, 7294496, 2108526, 6909805, 6955052, 6648615, 6906226, 7217267, 7566703, 6955052, 7741541, 7566703, 3026478, 6902560, 7302432, 2108521, 10090, 7645635, 7563617, 7304736, 2913141, 6646304, 7304736, 2126709, 6531523, 7632239, 6386277, 2126697, 5120033, 2911855, 7103776, 2123116, 2123117, 6906214, 6496372, 6646120, 2108530, 7628131, 2123124, 7497840, 6648673, 8480]

Pour un humain ces données ne veulent pas dire grand-chose. Mais n'oublions pas que c'est le langage des ordinateurs, car nous l'avons intercepté entre 2 ordinateurs. L'ordinateur code les chiffres et les lettres en binaire, et une lettre fait 8 bits, nous avons remarqué que chaque bloc du message correspond à 3 caractères. Sois 3 octets, soit 3x8=24 bits, nous avons lu le premier mot qui nous donne :

Premier bloc mis en hexadécimal : hex(6909773) => '0x696f4d'

Il faut lire le message en « little endian » de droite à gauche en séparent les données en octets (8 bits)

Séparé en 3 morceaux de 8 bits. Et traduit de l'hexadécimal à un caractère à l'aide de la Table ASCII

La fonction « chr() » traduit en (utf-8) les octets.

Donc, chr(0x4d) ='M', chr(0x96) = 'o', chr(0x69) = 'i', le premier bloc nous donne le mot "Moi".

Désormais, il suffit de coller tous les autres blocs à la suite pour trouver notre message.

Conclusion

Pour arriver à notre résultat final !

Notre message intercepté décoder !!!

"Moi, j'serais vous, je vous écouterais... Non, moi, j'serais nous, je vous... Si moi, j'étais vous, je vous écouterais ! Non, elle me fait chier, cette phrase !"

Ce message énigmatique nous a permis de trouver la source de l'information : ceci est une citation d'une personne fictive de la série Kamelott, dit par Perceval. L'interlocuteur doit probablement être un fan de la série. Ce qui qui limite notre recherche aux 300'000 fans sur leur page Facebook !

Nous n'avons trouvé aucun danger pour la Suisse. Tout va bien.

Si vous avez la chance d'intercepter un autre message, où vous auriez la clé publique et le message.

Nous avons mis à votre disposition sur notre git un code écrit en python.

Le lien : https://gitedu.hesge.ch/math-applique/rsa.

Pour l'utiliser, c'est très simple, il suffit de copier dans la variable « message_list » votre message et de changer **e** et **n** avec les votre.

Ensuite, lancez le programme dans la console avec la commande « python main.py », et votre message apparaitra.