**RS40 : TP - RSA**

**Compte Rendu**

ALBUZLU Gökdeniz

Pour le : 16 mai 2022

Semestre : Printemps 2022

**Introduction** :

Dans ce TP le but est de comprendre l’algorithme RSA dans un échange entre 2 utilisateurs qui sont Alice et Bob. Grâce à ce TP nous avons pu manipuler et observer les clés publiques/privées, les exposants publics, mais également les différentes fonctions qui composent l’algorithme RSA.

**Sommaire**

Table des matières

[**I.** **Les variables** 2](#_Toc103188804)

[**II.** **Les fonctions** 3](#_Toc103188805)

[II.1 home\_pgcd 3](#_Toc103188806)

[II.2 home\_string\_to\_int et home\_int\_to\_string 3](#_Toc103188807)

[II.3 home\_mod\_expnoent(x,y,n) 4](#_Toc103188808)

[II.4) home\_ext\_euclide 5](#_Toc103188809)

[II.5) longueurMaximale 5](#_Toc103188810)

[II.6) home\_crt 6](#_Toc103188811)

[III) Hachage 7](#_Toc103188812)

[Conclusion 7](#_Toc103188813)

# **Les variables**

Une image contenant texte

Description générée automatiquementVoici ce que nous avons vue dans le cours ;

Une image contenant texte

Description générée automatiquementVoici ci-dessous les différentes variables utilisées pour Bob dans le programme ;

* x1b et x2b sont les deux grand nombre premiers générées par Bob. Pour notre cas leurs taille sont de 300.
* la variable nb correspond donc à n=p\*q et phib est l’indicateur d’Euler
* eb correspond à l’exposant public il est égale à 65537
* db correspond à la clef secrète

Pour Alice nous avons les mêmes variables identiques.

Une image contenant texte

Description générée automatiquement

# **Les fonctions**

## II.1 home\_pgcd

Une image contenant texte

Description générée automatiquementCette fonction prend en paramètre deux entier a et b et nous retourne leur plus grand diviseur commun.

## II.2 home\_string\_to\_int et home\_int\_to\_string

Une image contenant texte

Description générée automatiquementCette fonction prend en paramètre une chaine de caractère et le converti en entier

Une image contenant texte

Description générée automatiquementLa deuxième fonction est une fonction réciproque elle convertie un nombre entier en chaine de caractère

## II.3 home\_mod\_expnoent(x,y,n)

Une image contenant texte

Description générée automatiquementLa fonction home\_mod\_expnoent(x,y,n) doit permettre de calculer x^y%n. Cette fonction sert à chiffrer un nombre x avec une clé publique (y, n) afin de compléter cette fonction, j’ai principalement utilisé l’algorithme de l’exponentiation modulaire rapide. Voici l’algorithme :

En effet, cet algorithme à été adapté en fonction du langage python, voici ce qu’on obtient :

Une image contenant texte

Description générée automatiquement

## II.4) home\_ext\_euclide

Cette fonction sert à calculer l’inverse modulaire, c’est-à-dire l’inverse de y (mod b). On applique donc l’algorithme d’Euclide étendu avec le théorème de Bézout que nous avons étudié dans le cours.

Une image contenant texte

Description générée automatiquement

## II.5) longueurMaximale

La fonction longueur Maximale prend en paramètre les deux grands nombres générés par Bob et indique le nombre de caractères maximale que l’utilisateur peut saisir. Le processus de calcul se fait de la manière suivante :

Une image contenant texte

Description générée automatiquement

Ici on cherche simplement le nombre de bits qui correspond au message, pour cela on utilise les puissances de 2 avec une boucle qui a pour condition d’arrêt n=p\*q (ici c’est x1\*x2). Et puisque les caractères sont codés sur 8 bits, on doit diviser le nombre qu’on obtient par 8 afin d’envoyer à l’utilisateur le nombre de caractères maximale qu’il peut saisir.

### II.6) home\_crt

Le théorème du reste chinois permet de calculer . C’est donc une alternative à l’exponentiation modulaire. Le théorème du reste chinois est plus léger que l’exponentiation modulaire et possède l’avantage également de se protéger de l’attaque par canal auxiliaire

Pour implémenter le théorème du reste chinois j’utilise l’algorithme ci-dessous :

Une image contenant texte

Description générée automatiquement

Voici la fonction python qui correspond à cet algorithme :

Une image contenant texte

Description générée automatiquement

On remplace donc la fonction home\_mod\_expnoent par la fonction home\_crt pour le déchiffrement du message, de la manière suivante :

# III) Hachage

La fonction de MD5 est une fonction de hachage qui n’est plus considérée comme sécurisée. Il existe des fonctions de hachage qui sont meilleurs, comme SHA-256 ou encore SHA-512. Ici, j’ai décidé de remplacer md5 par SHA-256. Donc on l’utilise pour le chiffrement et le déchiffrement comme ci-dessous :





# Conclusion

Ce programme de RSA codé en python, regroupe les différentes étapes du système RSA vue en cours comme la génération des clés, les différents algorithme (exponentiation modulaire, théorème du reste chinois…). Enfin, elle nous montre surtout l’importance de l’impact du choix de la taille des clés sur le message chiffrable ainsi que leur utilisation.