# “VCES” & “true-RSA”

## GS15 - A15 - Projet Informatique

## Sujet présenté en cours le 25/10

## Rapport à rendre avant le 09/01 - Soutenance entre le 09/01 et le 13/01

---

# 1 Description du projet à réaliser

Le but de ce projet informatique est de vous faire créer un outil offrant la possibilité d'utiliser deux algorithmes différents de chiffrement : l'un à symétrique, l'autre à clé publique et un algorithme de signature. Le choix de l'algorithme utilisé est laissé à l'utilisateur, qui pourra par exemple l'indiquer en entrant un nombre spécifique au clavier avec une interface du type :

```
Selectionner votre fonction de chiffrement
->1<- Chiffrement symétrique VCES
->2<- Chiffrement RSA avec module multiple
->3<- Signature RSA avec module multiple
->4<- Déciffrement RSA
->5<- Vérifier une signature RSA
```

L'utilisateur entre son choix (1, 2, 3, 4 ou 5) ... et le programme doit ensuite le guider, lui demander de choisir une clé, un fichier, etc. ...

Les trois algorithmes qui vous sont demandés sont décrits ci-dessous, respectivement dans les sections 2 et 3.

Il est conseillé de ré-utilisé les fonctions données en TP pour la lecture et l'écriture des fichiers ainsi que les fonctions que vous avez pu écrire durant les séances de TP.

Enfin, le choix du langage de programmation vous appartient, néanmoins votre enseignant n'étant pas omniscient, un soutient n'est assuré que pour les langages Matlab/GMPint et C/GMP. La seule contrainte **obligatoire** est seulement de respecter les consignes données dans la section 4 du présent document.

Vous devrez rendre un rapport ainsi que votre code (commenté !).

Date limite de restitution : **Dimanche 8 janvier à 23h59** (au délà, un point sera enlevé par minute de retard).

Une soutenance est prévue la semaine précédent les examens finaux, vous devrez vous inscrire pour “réserver” un horaire pour votre présentation.

---

# 2 DES+AES=VCES

Dans cette partie, le but du projet est de proposer une méthode de chiffrement mixant AES et DES. Ce schéma est appelé le VCES (**V**ery **C**omplex **E**ncryption **S**cheme), il utilise des blocs de 128 bits et des clés

de 128 bits et est définit ci-dessous. Les "détails" qui ne sont pas spécifiés dans la définintion doivent être choisis par vos soins (vous pourrez discuter ces choix, section 4) .
Il vous ait demandé de d'écrire une méthode de chiffrement symétrique dans lesquelles les itérations de DES et de AES s'enchaînent de façon alterné.

On rappelera que DES est basé sur les schémas de Feistel et utilise donc des itérations définies par :

$$\begin{cases} L_{i+1} = R_i \\ R_{i+1} = L_i \oplus F(K_i, R_i) \end{cases} \tag{1}$$

où $L_i$ et $R_i$ représentent les deux moitié d'un bloc de $64$ après application de la $i$-ième itération; $K_i$ représente la clé de la $i$-ème itérations et $F$ une fonction vérifiant $F : 0,1^{32} \times 0,1^{48} \mapsto 0,1^{32}$.
On rappelera également que la fonction $F$ est définie par les étapes suivantes :

1. Extension du bloc de $32$ à $48$ bits (permutation initiale) ;
2. XOR entre la clé de l'itération (de $48$ bits) et le bloc étendu ;
3. Application des S-boxes par bloc de $6$ bits avec résultat de $4$ bits pour chacune ;
4. Applications de la permutation finale (inverse de la permutation initiale).

Le chiffrement symétrique "AES" est quant à lui un schéma de chiffrement symétrique par réseaux de substitution - permutation. Ce schéma utilise des clés de $128$, $192$ et $256$ bits et des blocs de taille fixée à $128$ bits. Dans le schéma AES le bloc de données à chiffrer est défini pa run matrice et chaque itération est définie par les opérations suivantes :

1. SubBytes : inversion de chaque élément (octet) dans le corps $G\mathbb{F}_{2^8}$. Vous pourrez implémenter cette fonction en refaisant les calculs d'inversion dans ce corps particulier ou en tabulant les opération une fois pour toutes (si votre implémentation utilise le calcul vous pourrez aussi vous amuser à changer la matrice $\mathbf{A}$ et le vecteur $\mathbf{c}$ de l'application affine.
2. ShiftRows opérant un décalage des lignes des états.
3. MixColumns qui effectue une substitution des colonnes par multiplication matricielle. Là encore, vous pourrez si vous le souhaitez tabuler les opérations (de multiplication) ou bien effectuer les calculs.
4. AddRoundKey combinant par $\oplus$ / XOR le bloc de données avec la clé de l'itération.

Contrairement à DES, il est important de noter que ces fonctions doivent être inversibles et sont inversé une à une lors d'une itération de déchiffrement.

Votre algorithme de chiffrement doit chiffrer et déchiffrer des blocs de $128$ bits de la façon suivantes :

1. Chaque bloc est découpé en deux blocs de $64$ bits. Chacun de ces blocs est utilisé dans une itération (de Feistel) du chiffrement DES.

2. Ensuite, le résultat des deux blocs est concaténé est utilisé dans une itération du chiffrement AES (substitution-permutation).

3. Vous devrez donc générer des clés pour chacune des itération pour le réseau de Feistel et le réseau SP ; la génération des sous-clés est laissé à votre choix.

En bref : votre schéma doit réaliser un chiffrement en appliquant une itération de DES puis une itération de AES. Le nombre d'itération total est (minimum) de 10 !
(un bonus est attribué si votre code peut chiffrer avec un débit supérieur à 1Gbps).

# 3 Chiffrement RSA avec modules multiples

Dans cette partie il vous ait demandé d'implémenter le système de chiffrement RSA avec modulo multiples. Autant que faire se peut, il vous est demandé d'implémenter :

1. La génération des clés (ce qui inclut la génération des nombres premiers $p_1, p_2, \ldots, p_N$) ;

2. Un système pour écrire les clés publiques et privés dans un fichier ;

3. Le code permettant de chiffrer un message en utilisant le fichier contenant la clé publique ;

4. Le code permettant de déchiffrer en utilisant le fichier de clé privée ;

5. Le code permettant de générer une signature en utilisant la clé privée ;

6. Le code permettant de vérifier une signature en utilisant la clé publique.

7. Enfin, il est également demander d'utiliser le RSA avec le padding type PKCS vu en cours.

Naturellement, il est ici impératif d'utiliser des grands entiers (au moins quelques centaines de bits) et donc l'utilisation de la toolbox Matlab GMPint est nécessaire.

Enfin, le choix de la fonction de hashage est laissé à votre discretion, une fonction "un peu tordue quand même, mais pas trop compliqué à implémenter" est recommandée.

# 4 Documents à fournir et autres détails

Il est impératif que ce projet soit réalisé en binôme. Tout trinôme obtiendra une note divisé en conséquence (par $3/2$, soit une note maximale de $13,5$).
Encore une fois votre enseignant n'étant pas omniscient et ne connaissant pas tous les langages informatique du monde, l'aide pour la programmation ne sera assuré que pour Matlab/GMPint et C/GMP. Par ailleurs, votre code devra être commenté (succinctement, de façon à comprendre les étapes de calculs, pas plus).
Votre code doit être a minima capable de prendre en entré un texte (vous pouvez aussi vous amuser à assurer la prise en charge d'image pgm comme en TP, de fichiers binaires, etc .... mais la prise en charge des textes est le minimum souhaité).

Un court rapport est également attendu; ce dernier devra argumenter les choix que vous avez fait notamment en ce qui concerne les implémentations que vous avez fait (par exemple, comment générer une clé pour chaque tournée, comment faire les calculs, etc. ... ).
La présentation est très informelle, c'est en fait plutôt une discussion autour des choix discuter dans votre rapport avec démonstration du fonctionnement de votre programme.

Vous avez le droit de chercher des solutions sur le net (ou bien où vous voulez), par contre, essayez autant que possible de comprendre les éléments techniques trouvés (voire les présenter dans votre rapport s'ils sont intéressants, par exemple comment trouver un entier premier sécurisé, comment utiliser RSA avec plusieurs modules, etc. . . . ).

Enfin, vous pouvez vous amuser à faire plus que ce qui est présenter dans ce projet ; par exemple proposer un protocole d'échange de clé basé sur votre chiffrement RSA pour votre VCES, etc. . . .

Je réponds volontiers aux questions (surtout en cours / TD) mais ne ferais pas le projet à votre place ... bon courage !