DÉPARTEMENT
Sciences Informatiques

Licence Pro R&S
2012–2013
Chiffrement et authentification
T.T. Dang Ngoc
dntt@u-cergy.fr

TD 2 - Bases de la cryptographie moderne

1 Chiffrement par bloc

L'idée générale du chiffrement par blocs est la suivante :

1. Remplacer les caractères par un code binaire (ASCII en base 2). On obtient ainsi une longue chaîne de 0 et de 1.
2. Découper cette chaîne en blocs de longueur donnée, par exemple 12 bits.
3. Déplacer certains bits du bloc.
4. Chiffrer un bloc en l'"additionnant" bit par bit à une sous-clef dérivée d'une clef (la dérivation des sous-clef s'appelle key schedule). Le key schedule (préparation des clés) consiste à générer des sous-clés à partir de la clé principale pour un algorithme de chiffrement par bloc.
5. Recommencer éventuellement un certain nombre de fois l'opération 3. On appelle cela une ronde.
6. Passer au bloc suivant et retourner au point 3 jusqu'à ce que tout le message soit chiffré.

Soit le message "salut" que vous voulez coder à l'aide de la clef de 24 bits "waf" en utilisant une taille de bloc de 12 bits.

1. Remplacer les caractères par un code binaire

Codez le message "salut" en ASCII 8 bits

-------- -------- -------- -------- --------

2. Découpage en blocs

Découpez cette chaîne en blocs de longueur 16 bits. Le remplisage se fera avec des 0.

Bloc 1 : ----------------
Bloc 2 : ----------------
Bloc 3 : ----------------
Bloc 4 : ----------------

3. Ajout de la clef

Nous utiliserons un algorithme simple se contentant de découper la clé de 24 bits en trois morceaux de 8 bits qui sont utilisés dans les différentes rondes. (on utilisera la sous-clef 1 dans la ronde 1, la 2 dans la ronde 2, la 3 dans la ronde 3, la 1 à nouveau dans la ronde 4, la 2 dans la ronde 5, etc.)

On utilisera des 0 comme remplissage de blocs.

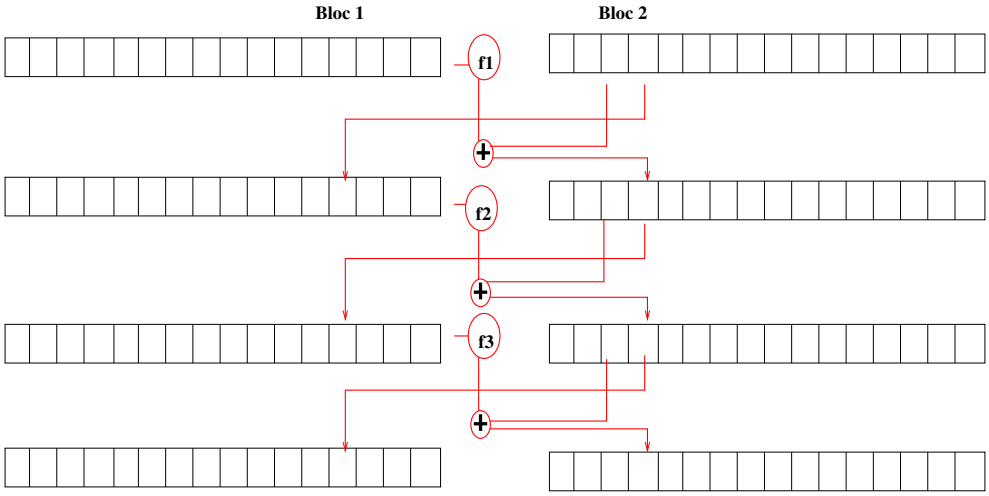
Chiffrement de la clef "waf" en ASCII 8 bits :

`-------- -------- --------`

Sous-clef 1 : `--------`
Sous-clef 2 : `--------`
Sous-clef 3 : `--------`

On utilisera un réseau de Feistel sur 3 rondes. La fonction f_i utilisée ici à chaque ronde $i(mod 3)$ sera simplement un XOR avec la sous-clef i suivi d'une rotation de 1 bit vers la droite.

On réalisera l'opération sur les deux premiers blocs seulement.



Utilisez l'annexe pour effectuer les calculs intermédiaires.

Une fois les 2 premiers blocs ainsi chiffrés au bout des 3 rondes de Feistel obtenus. Réfléchissez comment votre correspondant pourra déchiffrer (c'est à dire reconstituer les 2 blocs initiaux) en connaissant la clef. Effectuez les opérations pour retrouver les 2 blocs initiaux.

2 Diffie-Hellman-Merkle

Soit la fonction $f(x) = Y^x(modP)$ avec P premier et $Y < P$. Soit $Y = 7$ et $P = 11$. On considère donc dans cet exercice, la fonction suivante :

$$f(x) = 7^x(mod11)$$

1. Choisissez chacun un nombre secret A et calculez $f(A)$.
2. Choisissez un camarade avec qui vous allez communiquer, envoyez-lui votre $f(A)$.
3. De la même manière, lui même choisira un nombre secret B et calculera $f(B)$ qu'il vous communiquera.
4. Calculez $(f(B))^A(mod(P)$ (et votre camarade $(f(A))^B(mod(P))$

Constatez que vous obtenez le même nombre. Ce nombre est la clef de session.

Remarquez que si quelqu'un avait intercepté vos échanges et avait capté $f(x) = 2^x(mod11)$, $f(A)$ et $f(B)$, il aurait été très difficile pour lui de deviner A, B ou la clef de session.

3 Rivest-Shamir-Adleman (RSA)

3.1 Calcul d'une paire de clef publique/privée

1. Choisissez deux nombres p et q premiers (une liste des 147121206 premiers nombres premiers est disponible sur http ://www.bigprimes.net/archive/prime
2. Prendre un nombre e qui n'a aucun facteur en commun avec $(p-1)(q-1)$.
3. Calculer d tel que $e \times dmod(p-1)(q-1) = 1$, c-à-d d est le modulo inverse de e dans $Z/(p-1)(q-1)Z$: $d = e^{-1}mod((p-1)(q-1))$. Vous vous aiderez pour cela du programme `Inverse.class` que vous pouvez récupérer sur

   `http://depinfo.u-cergy.fr/~dntt/supports/crypto-applique/Inverse.class`

   se lance de la manière suivante :
   java Inverse *la_valeur_de_e la_valeur_de_(p-1)(q-1)*
   et permet de calculer $d = e^{-1}mod(p-1)(q-1)$
4. Calculez $n = p \times q$
5. (e, n) sera la clef publique et (d, n) la clef privée.
6. Publiez votre clef publique (Donnez la à votre voisin).

p = [ ] q = [ ]

Clef privée :
n = [ ] d = [ ]

Clef publique :
n = [ ] e = [ ]

3.2 Chiffrement

Récupérez la clef publique de votre voisin, puis choisissez un mot (6 lettres max !!!). Puis

1. Codez le mot sous forme d'une séquence de nombre :
   (a) en le codant en nombre à l'aide de la table ASCII
   (b) en le convertissant en binaire
   (c) en le découpant par bloc de 6 (remplir avec des 0 finaux si nécessaire)
   (d) convertir en décimal
2. chiffrez avec la clef publique du destinataire en faisant pour chaque nombre $C = M^e(modN)$. Vous utiliserez un calculateur à entier long, par exemple celui sur :
   `http://www.jpvweb.com/cgi-bin/calculextcgi.py`

Ecrivze le message chiffré et transmettez le à votre voisin.

3.3 Déchiffrement

Le destinataire d'un message chiffré (votre voisin) le déchiffrera en :

1. Déchiffrant la séquence en calculant pour chacun des nombres : $M = C^d(modN)$
2. Décodera la séquence obtenue en :
   (a) convertissant en binaire
   (b) en prenant par bloc de 7
   (c) en convertissant en caractère d'après la table ASCII

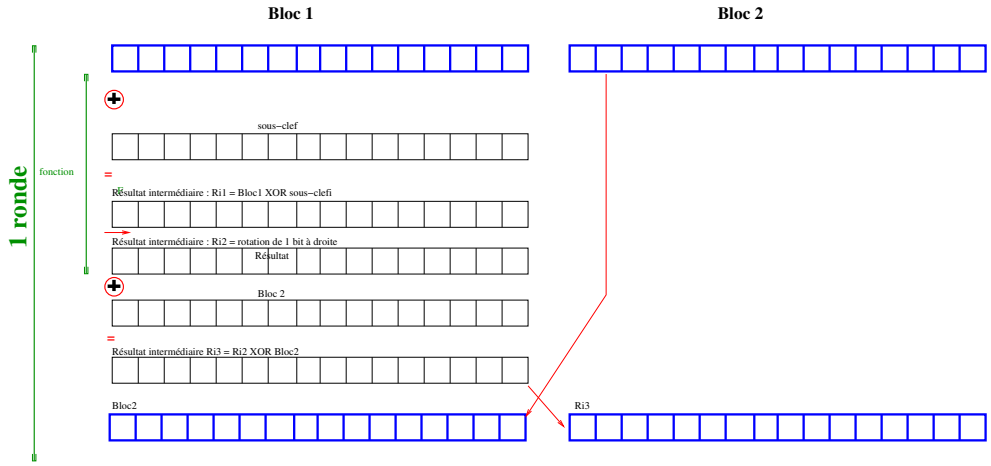
4 Annexes

4.1 Table ASCII 8 bits

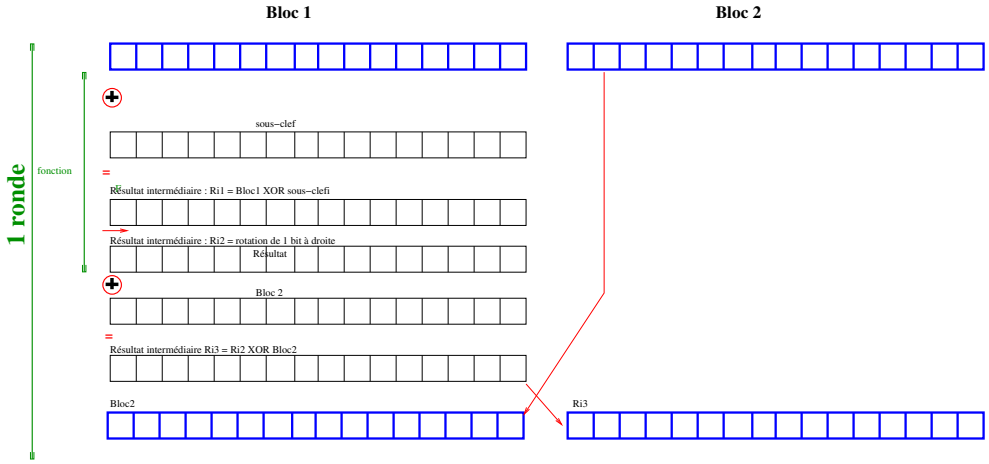
Code dec.	Code bin.	Signif.
32	00100000	ESPACE
65	01000001	A
66	01000010	B
67	01000011	C
68	01000100	D
69	01000101	E
70	01000110	F
71	01000111	G
72	01001000	H
73	01001001	I
74	01001010	J
75	01001011	K
76	01001100	L
77	01001101	M
78	01001110	N
79	01001111	O
80	01010000	P
81	01010001	Q
82	01010010	R
83	01010011	S
84	01010100	T
85	01010101	U
86	01010110	V
87	01010111	W
88	01011000	X
89	01011001	Y
90	01011010	Z

4.2 Calculs intermédiaires pour les 3 rondes

Ronde 1



Ronde 2



Ronde 3