


Labo Crypto (90 min)

1	Objectifs	<code>sudo ./c 2</code>
	<p>Ce travail de laboratoire de 90 minutes utilise l'excellent outil gratuit CrypTool 1.4.30 pour illustrer pratiquement :</p> <ul style="list-style-type: none">• le chiffrement symétrique (César, OU exclusif, DES, AES),• la compression,• la fonction de hachage,• l'algorithme RSA,• la signature numérique,• les systèmes hybrides	
Action	Ouvrir une session utilisateur Username= <code>albert</code> password= <code>admin</code> sous Windows7	
Lien	Documents utiles dont le corrigé dans le partage réseau \\10.2.1.1\doclabo\Secu\Crypto	
	Utiliser le raccourci du bureau 	
2	Chiffrements mono-alphabétique et poly-alphabétique	10'
But 2.1	Attaque basée sur la fréquence d'apparition des caractères	
Rappel	Jules César chiffrait ses messages en décalant chaque caractère de N positions dans l'alphabet ; N constituant la clé secrète.	
Action	Lancer CrypTool à partir du raccourci bureau Vous constatez que l'outil a ouvert le fichier startingexample-en.txt Sélectionner cette fenêtre startingexample-en.txt Chiffrer (Encrypt – Symmetric (classic) – Caesar) avec une clé "courte" comme C Observer le chiffrement mono-alphabétique affiché en bas de la nouvelle fenêtre Observer le <i>cyphertext</i> Déchiffrez-le (Decrypt – Symmetric (classic) – Caesar) puis entrer la clé Observer le <i>plaintext</i> Déterminer la fréquence d'apparition des caractères (Analysis – Tools for Analysis – Histogram) Prendre garde de sélectionner la bonne fenêtre <i>plaintext</i> ou <i>ciphertext</i> Superposer les 2 histogrammes pour casser la clé	
Question 2a	Obtient-on une meilleure protection avec une clé "longue" comme Z ?	
But 2.2	Enigma	
Action	Fermer toutes les fenêtres précédentes Dans CrypTool, sélectionner Individ. Procedures – Visualization of Algorithms – Enigma Suivre les informations affichées Observer les caractères chiffrés pour le plaintext = ssss	
Question 2b	Peut-on utiliser la méthode précédente (fréquence d'apparition des caractères) pour deviner le plaintext ?	

3	Chiffrement avec l'algorithme OU exclusif	10'
----------	--	------------

Introduction Le *ciphertext* est chiffré bit à bit à l'aide de la fonction OU exclusif et d'une clé cyclique
Le *plaintext* CrypTool.bmp se trouve dans C:\Program Files\CrypTool\examples

But 3.1 Analyse du *cleartext*

Question 3a Quelle est l'octet (8 bit) le plus fréquent dans cette image ?

Action Open – Files of type: All files – CrypTool.bmp
Analysis – Tools for Analysis – Histogram

Question 3b Quelle est la taille du fichier (données utiles) ?

But 3.2 Chiffrer ce document avec l'algorithme OU exclusif

Action Encrypt – Symmetric (classic) – XOR
Choisir une clé par exemple 12345678

Question 3c Quelle est la taille du document chiffré ?

But 3.3 La fonction d'autocorrélation permet de déterminer la longueur de la clé

Action Analysis – Symmetric Encryption (classic) – Ciphertext-Only – XOR
Entrer le caractère (8 bit) le plus fréquent puis Continue

Remarque On parle de *ciphertext only attack* car l'attaque ne porte que sur le document chiffré

Question 3d Expliquer la méthode utilisée par cette fonction d'autocorrélation pour deviner la clé

But 3.4 Montrer que la compression du fichier peut rendre cette attaque plus difficile

Action Indiv. Procedures – Tools – Compress – Zip

Question 3e Quelle est la taille du fichier compressé ?

Action Chiffrer ce document avec la même clé que précédemment
Utiliser la fonction d'autocorrélation et observer cette fois que l'attaque n'est plus si facile

Question 3f Expliquer pourquoi un texte compressé résiste mieux à une attaque traditionnelle

4	Entropie d'une source	10'
----------	------------------------------	------------

Rappel L'entropie permet de mesurer le degré d'aléa d'un message (en clair, compressé ou chiffré)
Entropie $H =$ valeur moyenne de la quantité d'information H_i portée par chaque symbole du message
 $H = - \sum p(i) \log_2 [p(i)]$ en bit

But 4.1 Mesurer l'entropie de divers fichiers

Action Ouvrir (File - Open) le fichier CrypTool.bmp
Sélectionner cette fenêtre puis Analysis – Tools for Analysis – Entropy

Question 4a Quelle valeur obtenez-vous pour l'image CrypTool.bmp ?

Question 4b Quelle valeur obtenez-vous pour l'image CrypTool.bmp chiffrée selon méthode du §3 ?

Question 4c Quelle valeur obtenez-vous pour l'image CrypTool.bmp compressée ?

Question 4d Quelle valeur obtenez-vous pour l'image CrypTool.bmp compressée puis chiffrée selon §3 ?

5	Longueur du <i>ciphertext</i>	10'
----------	--------------------------------------	------------

Rappel Pour le chiffrement à clé symétrique, la majorité des algorithmes (AES, ...) est basée sur un chiffrement par bloc produisant ainsi un *ciphertext* de longueur multiple de la taille du bloc (qui est souvent la taille de la clé)
 Parmi les algorithmes utilisés pratiquement, seul RC4 fonctionne sur le principe du chiffrement bit à bit

But 5.1 Déterminer la longueur minimale de divers *ciphertexts*

Action New pour créer un *cleartext* approprié
 Encrypt – Symmetric (modern) – DES (ECB) ; choisir une clé différente de celle proposée

Question 5a Quelle est la longueur du bloc DES (ECB) ?
 Utiliser une représentation hexadécimale avec View – Show as HexDump

Action Encrypt – Symmetric (modern) – Rijndael (AES)

Question 5b Quelle est la longueur du bloc AES ?

Action Encrypt – Symmetric (modern) – RC4

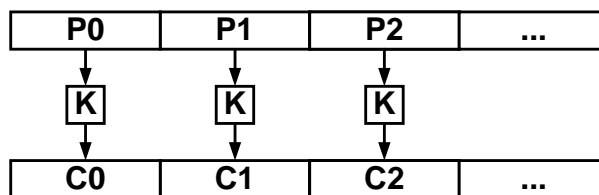
Question 5c Quelle est la longueur du *ciphertext* RC4 ?

Remarque Le prof Vaudenay et ses étudiants ont démontré la possibilité d'attaquer une implémentation naïve d'algorithme de chiffrement par blocs
 Voir SSL_etude_J_C_Asselborn.pdf dans <\\10.2.1.1\doclabo\Secu\Crypto>

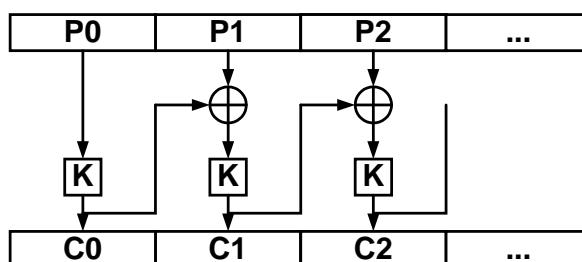
6	Modes ECB et CBC	5'
----------	-------------------------	-----------

Le mode ECB (*electronic codebook mode*) chiffre indépendamment chaque blocs (*plaintext*) alors que, dans le mode CBC (*cipher block chaining mode*), chaque bloc chiffré (*ciphertext*) opère une action sur les blocs en clair suivants

Mode ECB



Mode CBC



But 6.1 Proposer une méthode pour démontrer l'intérêt du mode CBC par rapport au mode ECB

Question 6 Expliquer votre méthode

But 7.1 Déterminer l'empreinte MD5 (hash) d'un texte

Action New puis entrer les chiffres 987654321
Indiv. Procédures – Hash – Hash Demonstration
Sélectionner MD5
Visualiser son empreinte → Hash value of the original file
Modifier légèrement le texte initial (par exemple le texte 987654320 ne diffère que d'un seul bit)
Observer les différences au niveau des 2 empreintes (nombre de bits différents)

Question 7a Quelle est la longueur d'une empreinte MD5 ?

Question 7b Est-elle fonction de la longueur du texte présent à l'entrée de la fonction de hachage ?

Question 7c Quel est l'intérêt de la fonction SHA-1 ?

But 7.2 Déterminer la valeur mémorisée par Windows (MD5) pour le mot de passe q2w3e4

Question 7d En répétant l'opération N fois, combien obtenez-vous d'empreintes différentes ?

But 7.3 Effectuer un contrôle d'intégrité de fichiers

Action Récupérer localement (bureau) les 2 fichiers original.txt et copy.txt situés dans le partage réseau
<\\10.2.1.1\doclabo\Secu\Crypto>

Question 7e Ces 2 fichiers sont-ils identiques ?
Expliquer la méthode utilisée

**But 7.4 Générer une "bonne" clé symétrique de 128 bit
"bonne" signifiant que les 2^N valeurs possibles sont équiprobables**

Question 7f Comment procédez-vous ?

Remarque Voir aussi PRNG = Pseudo Random Number Generator
<http://fr.wikipedia.org/wiki/PRNG>

Rappel Le crypto-système RSA (*Rivest, Shamir, Adleman* - 1977) continue d'être utilisé malgré son âge. Sa sécurité repose sur la difficulté de factoriser ...

Théorie		Illustration
1) Choisir aléatoirement 2 grands nombres premiers p et q		p=7 q=13
2) Calculer $n = p \cdot q$ $(p-1)(q-1)$		$n = 7 \cdot 13 = 91$ $6 \cdot 12 = 72$
3) Choisir e tel que $1 < e < (p-1)(q-1)$ $\text{pgcd}(72, e) = 1$ → clé publique <e,n>		$e = 5$ <5,91>
4) Calculer $d = e^{-1} \text{ mod } (p-1)(q-1)$ ou $d \cdot e = 1 \text{ mod } (p-1)(q-1)$ → $d = (X \cdot (p-1)(q-1) + 1) / e$ essayer $X = [1 \dots n]$ afin que d soit un entier $d = (2 \cdot (p-1)(q-1) + 1) / e$ → clé privée <d,n>		$d = 29$ <29,91>
5) Supprimer p, q, (p-1)(q-1)		

Action Fermer toutes les fenêtres
Sélectionner Indiv. Procédures – RSA Cryptosystem – RSA Demonstration
Entrer les 3 valeurs de l'illustration ci-dessus puis Update parameters

Entrer le texte HELLO puis *Update parameters*
Le logiciel vous demande de réduire la taille du jeu de caractères à moins de $n = 91$
Sélectionner Alphabet and number system ...
Choisir Specify alphabet (27 caractères majuscules) puis OK

Encrypt pour chiffrer votre texte
Copier (CTRL-C) le ciphertext dans le champ Input text
Decrypt pour déchiffrer le ciphertext

Question 8a Quelle clé est utilisée pour chiffrer ?

Question 8b Quelle clé est utilisée pour déchiffrer ?

Remarque La notation en base 10 indique la position du caractère dans l'alphabet

Rappel Grâce à la signature numérique, l'email ou le certificat que Bob reçoit est authentique, car seule Alice possède la clé privée appairée à la clé publique du certificat.

Théorie	Illustration
1) Alice signe le message M $C = M^d \text{ mod } n$	M = 17 $C = 17^{29} \text{ mod } 91 = 75$
2) Bob ... $C^e \text{ mod } n = M ?$	$75^5 \text{ mod } 91 = 17 = M$

But 9.1 **Générer une paire de clés asymétriques**
Action Fermer toutes les fenêtres
 Sélectionner Digital Signatures – PKI – Generate pour générer une paire de clés RSA de 1024 bits
 Remplir les 4 champs obligatoires (Last name, ...)
 Generate new key pair
 OK pour mémoriser clé privée et certificat

But 9.2 **Afficher le certificat numérique**
Action Sélectionner Digital Signatures – PKI – Display pour visualiser les clés RSA disponibles
 Sélectionnez la ligne correspondante puis Show certificate pour accéder aux divers champs du certificat X.509

But 9.3 **Signer un document**
Action New puis entrer le texte qu'Alice veut envoyer à Bob
 Digital Signatures – Sign Document avec les options par défaut
 Sélectionner la paire de clé
 Entrer le code PIN
 Observer le résultat : Signature au début du document puis votre texte à la fin

Remarque Digital Signatures – Extract Signature donne un affichage plus explicite

Question 9a Pourquoi avez-vous dû entrer votre code PIN ?

But 9.3 **Contrôler la signature du document**
Action Digital Signatures – Verify Signature
 Sélectionner la paire de clé

Modifier le document et revérifier la signature.
 Ne pas utiliser de *backspace* mais écrire la nouvelle valeur

Question 9b A quoi sert la signature ?

Question 9c Comment Bob contrôle-t-il cette signature ?

Question 9d Quels caractères du fichier signé sont protégés ?

10	Système hybride	10'
-----------	------------------------	------------

Objectif Illustrer le principe de fonctionnement des systèmes tels que SSL, EFS, IPSec, ... qui protègent les données en utilisant un chiffrement symétrique (performance) et transfèrent le secret partagé dans un canal sécurisé par un chiffrement asymétrique

But 10.1 Chiffrer le document avec une clé sym

Action
 Fermer toutes les fenêtres
 Encrypt – Hybrid – RSA-AES Encryption
 Créer un document texte sur le bureau
 Chaque élément visuel rouge doit être sélectionné pour passer au vert
 Open document : choisir un fichier
 Generate session key
 Encrypt document symmetr.

But 10.2 Chiffrer la clé sym (secret à partager)

Select asymmetr. key : utiliser la clé générée précédemment
 Encrypt session key asymmetr. (chiffrer la clé de session avec la clé publique)
 Les étapes intermédiaires sont visibles à partir des objets bleus
 Save

But 10.3 Effectuer les opérations inverses

Decrypt – Hybrid – RSA-AES Decryption
 ...

Question 10 Quand et pourquoi devez-vous entrer un code PIN ?

Remarque Dans le cas d'un échange SSL, le serveur transmet son certificat qui permet ainsi au client de transmettre la clé de session chiffrée

11	Pour ceux qui ont terminé tous les points précédents
-----------	---

11a Indiv. Procedures – RSA Cryptosystem – Prime Number Test
 Generate Prime Number
 Factorization of a Number
 Signature Demo

11b Indiv. Procedures – Protocols – Secure E-Mail with S/MIME

11c Indiv. Procedures – Hash – Generation of HMACs

11d Indiv. Procedures – Tools – Generate Random Numbers
 Password Quality Meter
 Password Entropy

11e Analysis – Tools for Analysis – N-Gram