Exercices de Cryptographie

ENSEIRB-MATMECA

3ème année filière Electronique,
Option Systèmes embarqués

Olivier VILLAIN

APICEM®

Exercices de cryptographie

Exercice 1 : Analyse de la sécurité web dans le navigateur

Nous allons étudier la sécurité procurée par les logiciels utilisateurs du grand public.
A l'aide du navigateur web installé sur la station, accédez au site de votre choix à condition qu'il utilise le protocole https.

1. **Comment êtes-vous sûr très rapidement que le site sécurise la connexion ?**
2. **Quel protocole de sécurité a été appliqué pour cette connexion ?**
3. **Quel type de chiffrement a été utilisé pour le trafic utilisateur ?**
4. **Expliquer tous les paramètres du chiffrement**
5. **Déterminez la chaîne de certificats qui a permis de vérifier l'authenticité du site**
6. **Le certificat a-t-il une période de validité ?**
7. **Vérifiez au moyen d'une autorité de certification (sur le site web de la CA) la validité du certificat du site que vous avez choisi. Pour cela utilisez l'URL de votre site pour la vérifier sur un site certifié.**
8. **Sur la station où sont stockés les certificats des autorités de certification racines de confiance**
9. **Sur le navigateur comment sont stockés les certificats obtenus sur les sites que vous visitez**
10. **Sur le navigateur, existe-t-il des exceptions accordées aux certificats non valides de certains site**

Exercice 2 : Analyse de la sécurité web avec des outils en ligne de commande

Nous allons ici découvrir les mécanismes employés par les applications afin de garantir la sécurité. Ceci sera fait au moyen de commandes système.

1. **A l'aide de commande dans un terminal, connectez-vous à un site utilisant https. Vous pouvez utiliser le site de l'exercice 1, ou un autre.**

2. **Vous pouvez essayer d'autres commandes pour initier une connexion à un site distant. Comparez les résultats obtenus**

3. **Retrouvez les informations que le navigateur a fournis à l'exercice 1.**

4. **Rappelez à quelle étape de la négociation TLS s'effectue l'échange des nombres aléatoires.**

5. **Quelle clé symétrique est utilisée pour les communications**

6. **Trouvez les nombres aléatoires générés par le client et le serveur**

7. **Au moyen d'une ligne de commande, affichez les informations contenues dans le certificat du site**

Exercice 3 : simulation d'attaque de type « Man In The Middle »

Nous allons ici voir comment les mécanismes de sécurité utilisant la cryptographie permettent de détecter une attaque classique.

1. **Décrire le principe de l'attaque « man-in-the-middle »**

2. **Faites un schéma expliquant un trafic « normal » et un trafic « intercepté par un tiers »**

3. **Comment s'effectue la négociation (« handshake ») TLS lorsque le trafic passe par un proxy. Combien de méthodes sont possibles**

4. **Installer l'outil MITM (https://www.mitmproxy.org/downloads/)**

5. **Démarrer le proxy**

6. **Paramétrer le navigateur sur la station pour rediriger le trafic par le proxy.**

7. **Accédez au site https://mitm.it/**

8. **Que se passe-t-il ? pourquoi ?**

9. **Décrivez la négociation TLS engendrée par cette situation**

10. **Quel est le danger ? est ce que l'usurpation peut être faite à notre insu ?**

11. **Comment peut-on savoir si nous sommes en face d'une usurpation ?**

12. **Si nous avons accepté la fausse identité, comment y remédier, et revenir à une situation sûre ?**

A la fin de cet exercice, rétablissez la situation « normale » de la station :

1. **Supprimez Mitm**
2. **Rétablissez le trafic direct sans proxy**
3. **Supprimez les certificats installés**
4. **Redémarrez le navigateur et vérifiez que la navigation est opérationnelle**

Exercice 4 : Chiffrement RSA

Nous allons étudier les mécanismes mathématiques de base qui sont utilisés dans le chiffrement et le déchiffrement. Nous choisirons RSA.

1. **Le nombre 18446744069414584321 est-il premier ?**

2. **J'ai un module 42A20022C0B89C280959. En déduire les facteurs premiers p et q de ce nombre**

3. **J'ai trouvé cette clé publique RSA :**

```
-----BEGIN PUBLIC KEY-----
MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBALAGlWM7J7dtIMqny+d365rEgYHtlhEU
a5Xpe7Svv1pNiXI14VK+YHiPdkoCW9fkZaIKLJmXT40E4hg76PlJpmkCAwEAAQ==
-----END PUBLIC KEY-----
```

   ⇨ **Quelle est la taille du chiffrement ?**

4. **Ecrire une fonction en C qui détermine aléatoirement 1 nombre premier de 64 bits maximum**

5. **Ecrire une fonction en C qui détermine si un nombre est premier (maximum 64 bits)**

6. **Ecrire une fonction en C qui décompose en facteurs premiers un nombre de 64 bits maximum**

7. **Mesurer le temps pris pour faire cette décomposition en utilisant une valeur proche de la limite maximale**

8. **Quelle taille devrait avoir le nombre à décomposer pour que cette opération prenne 10 jours**

9. **Ecrire 2 fonctions en C pour chiffrer et déchiffrer un caractère.**

Exercice 5 : Manipulation de BIGNUM

Nous allons étudier les performances des librairies utilisées en cryptographie pour gérer les nombres de très grande taille.
dans la suite de cette exercice, le terme « nombre » désignera toujours un Big Num.

1. **En utilisant la librairie BigNum d'OpenSSL, créer une fonction pour générer un nombre aléatoire de 2048 bits**

2. **Créez une fonction qui vérifie si un nombre est premier.**

3. **Créez une fonction qui génère un nombre premier.**

4. **Créez une fonction qui génère 2 facteurs premiers p et q**

5. **Créez une fonction qui factorise un nombre**

6. **Mesurez le temps de factorisation de N = p.q**

Exercice 6 : Chiffrement/déchiffrement en utilisant les API

Pour cet exercice, nous utiliserons les API fournies par Linux afin de gérer le chiffrement, le déchiffrement et la génération de clés.

1. **Quelles sont les API disponibles pour gérer la cryptographie sous Linux ?**

2. **Ecrire une fonction en C qui génère une paire de clés asymétriques.**
   ⇨ Les paramètres seront le type de chiffrement, la taille, et un mot de passe optionnel.
   ⇦ Les clefs sont enregistrées dans un fichier.

3. **Ecrire une fonction en C qui chiffre un tableau d'octets.**
   ⇨ L'entrée est le tableau contenant les informations en clair, la clé de chiffrement, le mot de passe optionnel.
   ⇦ En sortie le tableau chiffré.

4. **Ecrire une fonction en C qui déchiffre un tableau d'octets.**
   ⇨ L'entrée est le tableau chiffré, la clé de déchiffrement, le mot de passe optionnel.
   ⇦ En sortie le tableau en clair

Exercice 6 : Diffie-Hellman et TCP/IP

Nous allons écrire un projet qui aura pour but d'implémenter le partage de secret par l'algorithme de Diffie-Hellman.

1. **Ecrire un client TCP qui partagera un secret avec un serveur TCP**