	Lycée de l'Hyrôme - Chemillé	2012 - 2013
	TP Cryptographie APACHE HTTPS	Réseaux informatiques
	BTS IRIS	TP

Objectif:

Les objectifs du TP sont :

- Utilisation de l'outil openssl
 - Chiffrement symétrique
 - Chiffrement asymétrique
 - Hachage
- Initiation à la cryptographie SSL et mise en place d'une connexion https
 - Avec certificat auto signé
 - Avec certificat signé par une autorité

Partie 1 : Découverte de l'outil OPENSSL

Introduction (wikipédia) :

OpenSSL est une boîte à outils de chiffrement comportant deux bibliothèques (une de cryptographie générale et une implémentant le protocole SSL), ainsi qu'une commande en ligne.

Les bibliothèques (qui sont écrites en langage C) implémentent les fonctions basiques de cryptographie et fournissent un certain nombre de fonctions utiles.

Les paramètres de la commande en ligne OpenSSL sont très nombreux. Ils permettent d'indiquer entre autres l'un des nombreux types de chiffrement (exemple : blowfish, DES ou Triple DES, DSA, RC4, RC5, RSA...), d'encodage (base64 ou autres) et de hachage (MD5, SHA-1...).

Cet utilitaire et les bibliothèques associées sont disponibles pour la plupart des Unix, mais aussi pour Microsoft

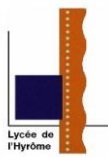
La page de manuel de openssl permet de visualiser l'ensemble des fonctionnalités et des sous-commandes associées (chaque sous commande dispose de sa propre page de manuel)

Fonction de hachage (sous-commande dgst) :

Tester la commande suivante et interpréter le résultat:
openssl dgst /etc/passwd

Modifier la commande en utilisant l'algorithme SHA1.

Modifier la commande en utilisant l'option -out pour stocker le résultat dans un fichier texte.

	Lycée de l'Hyrôme - Chemillé	2012 - 2013
	TP Cryptographie APACHE HTTPS	Réseaux informatiques
	BTS IRIS	TP

Chiffrement symétrique (sous-commande enc) :

Tester et interpréter la commande suivante :

```
openssl enc -bf -in /etc/passwd -out passwd.enc -k maclef
```

L'affichage du fichier chiffré peut poser des problèmes puisque ce n'est pas un fichier ASCII. La commande « hd fichier » permet de l'afficher en hexadécimal.

Modifier la commande pour utiliser l'algorithme DES

Trouver la commande permettant de déchiffrer le fichier passwd.enc

Dans les commandes précédentes la clef est directement utilisée sur la ligne de commande. Il est plus sécurisant d'utiliser une clef stockée dans un fichier, grâce à l'option « -pass file : » qui remplacera l'option -k

Créer un fichier texte protégé par des droits d'accès judicieusement choisis, y stocker une clef mélangeant les caractères spéciaux et alphanumérique avec une longueur d'au moins 8 caractères.

Reprendre les commandes précédentes en utilisant le fichier contenant votre clef.

Il est également possible de demander à openssl de générer une clef aléatoire avec la sous-commande rand : `openssl rand 32 -out key.temp`

Tester et interpréter la commande précédente

Chiffrement asymétrique RSA (sous-commandes genrsa, rsa, rsautl)

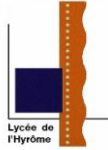
A l'aide de la page de manuel de genrsa, générer une clé privée sur 1024 bits en la stockant dans un fichier nommé private.key.clr. Observer le fichier généré.

Il est possible, à la création, de protéger la clef privée par un chiffrement DES.

Recommencer l'opération précédente en protégeant votre clef privée par un mot de passe DES3, en la stockant dans un fichier nommé private.key.enc. Observer le fichier généré..

A partir de la clé privée, il est possible de générer la clé publique correspondante avec l'option rsa.

Générer la clé publique public.key au format PEM à partir de la clé privée non protégée par un mot de passe.

	Lycée de l'Hyrôme - Chemillé	2012 - 2013
	TP Cryptographie APACHE HTTPS	Réseaux informatiques TP
BTS IRIS		

Le chiffrement et de déchiffrement se fait avec l'option rsautl de openssl :
Pour chiffrer, on utilisera aussi les options suivantes : -encrypt, -in, -out, -pubin, -inkey
Pour déchiffrer, on utilisera aussi les options suivantes : -decrypt, -in, -out, -inkey.

Réaliser un chiffrement/déchiffrement d'un fichier avec votre couple clé publique / clé privée.

Partie 2 : Mise en œuvre d'une connexion WEB sécurisée (SSL)

Les objectifs de cette partie sont :

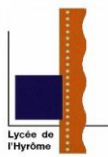
- Configurer apache pour le protocole https
- Génération de certificats auto-signés
- Génération de certificats signés par une autorité de certification
- Création de sa propre autorité de certification

1. Présentation:

SSL (**Secure Sockets Layers**, que l'on pourrait traduire par *couche de sockets sécurisée*) est un procédé de sécurisation des transactions effectuées via Internet. Le standard SSL a été mis au point par *Netscape*, en collaboration avec *Mastercard*, *Bank of America*, *MCI* et *Silicon Graphics*. Il repose sur un procédé de cryptographie par clef publique afin de garantir la sécurité de la transmission de données sur internet. Son principe consiste à établir un canal de communication sécurisé (chiffré) entre deux machines (un client et un serveur) après une étape d'authentification.

Le système SSL est indépendant du protocole utilisé, ce qui signifie qu'il peut aussi bien sécuriser des transactions faites sur le Web par le protocole HTTP que des connexions via le protocole FTP, POP ou IMAP. En effet, SSL agit telle une couche supplémentaire, permettant d'assurer la sécurité des données, située entre la couche application et la couche transport (protocole TCP par exemple).

De cette manière, SSL est transparent pour l'utilisateur (entendez par là qu'il peut ignorer qu'il utilise SSL). Par exemple un utilisateur utilisant un navigateur internet pour se connecter à un site de commerce électronique sécurisé par SSL enverra des données chiffrées sans aucune manipulation nécessaire de sa part.

	Lycée de l'Hyrôme - Chemillé	2012 - 2013
	TP Cryptographie APACHE HTTPS	Réseaux informatiques
	BTS IRIS	TP

Un serveur web sécurisé par SSL possède une URL commençant par *https://*, où le "s" signifie bien évidemment *secured* (*sécurisé*). Le port par défaut utilisé par le protocole https est 443.

2. Configuration du serveur Apache

- Visualiser le contenu du répertoire `/etc/apache2/mods-enabled/`

Par défaut le module SSL n'est pas activé pour Apache2, pour l'activer :

- taper la commande en étant root : `a2enmod ssl`
- Visualiser à nouveau le contenu du répertoire `/etc/apache2/mods-enabled/`

Afin de fabriquer les clefs publique et privé ainsi que le certificat numérique du serveur WEB, nous allons utiliser un outil (`make-ssl-cert`) disponible en installant le paquetage `ssl-cert` :

- Installer le paquetage `ssl-cert` (`apt-get ...`)

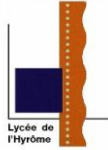
Fabrication du certificat autosigné:

Les fichiers de configuration d'Apache2 s'attendent à trouver la clé privée dans le répertoire : `/etc/apache2/` .

- Dans une console, en étant root, taper la commande qui génère le certificat (fichier `apache.pem`)
- `make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/apache2/apache.pem`

Remplir le formulaire, en ne mettant pas d'accent dans vos réponses. Le `commonName` est le nom de votre machine.

- Grâce à Dolphin (en root) visualiser le certificat et commenter les informations visualisées, on notera
 - l'algorithme utilisé pour la clé public du certificat ainsi que celui utilisé pour l'empreinte.
 - La période de validité

	Lycée de l'Hyrôme - Chemillé	2012 - 2013
	TP Cryptographie APACHE HTTPS	Réseaux informatiques
	BTS IRIS	TP

Il faut alors adapter la configuration d'Apache en fonction des noms donnés à la clé privée et au certificat.

- Un exemple de configuration ssl est fourni de le fichier `/etc/apache2/sites-available/default-ssl`
 - modifier ce fichier pour renseigner votre certificat.
- Activer le site https fournit par défaut avec la commande `a2ensite default-ssl`
- Tester la connexion sécurisée https à l'aide d'un navigateur sous Linux et sous Windows. Observer dans les détails les messages qui s'affichent. Les justifier.
 - Il ne devrait avoir qu'un seul avertissement concernant l'autosignature

On souhaite que l'accès en https sur votre serveur nous délivre une autre page web que celle délivrée par un accès en http.

- Créez un répertoire ssl dans le répertoire `/var/www/` et dans ce répertoire ssl un fichier `index.htm` (contenant votre nom par exemple)
- Modifier le fichier de configuration d'apache concernant SSL afin que la directive `DocumentRoot` corresponde au répertoire de votre site sécurisé.
- Tester la connexion à votre serveur en https.

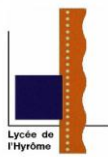
3. Les certificats signés par une autorité :

L'objectif de cette partie est de générer une clef privée et une demande de certification afin d'obtenir un certificat signé par une autorité certifiée.

- Suivre les étapes détaillées du document annexe 1 afin de fabriquer une clé privée sans protection par mot de passe, et une demande de certificat (fichier `csr`) associé à la clé privée.
- Installer la clé privée dans le répertoire `/etc/apache2/`

Pour obtenir un certificat signé, il faut souscrire un abonnement (payant) auprès d'une autorité certifiée et reconnue par les navigateurs. Toutefois, il est possible d'obtenir un certificat signé (par une autorité de test) temporaire pour des tests.

- Aller sur le site, <https://www.thawte.com/>, pour obtenir un certificat de test signé par

	Lycée de l'Hyrôme - Chemillé	2012 - 2013
	TP Cryptographie APACHE HTTPS	Réseaux informatiques TP
BTS IRIS		

cet organisme (fichier crt). Ce certificat de test nécessite l'importation au niveau du navigateur d'une autorité de certification de test.

- Installer ce certificat dans le répertoire /etc/apache2/ et modifier la configuration d'apache pour prendre en compte ce certificat et la clef privée associée.
- Relancer votre serveur WEB et tester la connexion https en utilisant un navigateur Windows.
- Afin d'avoir aucun message d'avertissement coté client, intégrer l'autorité de certification de test à votre navigateur (si elle est disponible)

4. Créer sa propre autorité de certification :

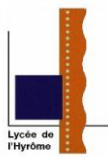
Les outils openssl permettent de fabriquer une CA pour une utilisation en Intranet. Cependant l'utilisation de ces outils peut s'avérer fastidieuse, c'est pourquoi il existe des outils graphiques pour linux : comme TINYCA

- Installer tinyca : apt-get install tinyca
- Lancer tinyca : \$kdesu tinyca2

- Via l'interface de TinyCA :

- Générer une nouvelle CA
- Remplir le formulaire
- Valider la création avec les caractéristiques par défaut
- Exporter le certificat de la CA dans un fichier
- Importer votre requête de certification (fichier csr de la question précédente)
- Signer la requête de certification avec votre CA pour un certificat serveur
- Exporter votre certificat serveur dans un fichier PEM
- Visualiser les différentes informations présentes sur l'IHM de TinyCA

- Importer le certificat de la CA dans votre navigateur WEB
- Reconfigurer Apache si nécessaire avec les nouveaux noms des fichiers (clé privé et certificat)
- Tester l'accès à votre site https, votre navigateur ne devrait afficher aucun message d'avertissement concernant la sécurité

	Lycée de l'Hyrôme - Chemillé	2012 - 2013
	TP Cryptographie APACHE HTTPS	Réseaux informatiques
	BTS IRIS	TP

Partie 2 : Bonus : Installation PHP, MySql, Phpmyadmin

1. Installation de php5

Installer le paquet Debian PHP5 : `apt-get install php5`

Vérifier que le module php pour apache est activé : lister le répertoire `/etc/apache2/mods-enabled/`

Redémarrer apache2

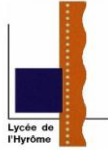
Créer un fichier `index.php` dans `/var/www/` avec le contenu suivant :

```
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <?php
      phpinfo();
    ?>
  </body>
</html>
```

Tester la page php avec votre navigateur

2. Installation de la base de donnée (Mysql)

PHP est très très souvent couplé à un système de base de données : Mysql. Nous installons ici Mysql-server version 5. Vous verrez plus bas que nous allons également installer phpmyadmin. Il s'agit d'un script php qui permet de gérer ses bases de données Mysql de façon très simple.

	Lycée de l'Hyrôme - Chemillé	2012 - 2013
	TP Cryptographie APACHE HTTPS	Réseaux informatiques TP
BTS IRIS		

Installer le paquet mysql-server : `apt-get install mysql-server`

Définir le mot de passe root de Mysql (« hyrome» par exemple).

Vérifier que Mysql est opérationnel : `# mysql -p`

entrer le mot de passe

>Exit;

Installer les librairies php5-mysql : `apt-get install php5-mysql`

Redémarrer apache2

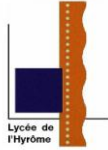
Vérifier qu'un bloc mysql apparaît avec index.php

Installer PhpMyAdmin : `apt-get install phpmyadmin`

Redémarrer apache2

Observer le contenu du fichier de configuration de phpmyadmin (/etc/apache2/conf.d) pour déterminer l'URL à utiliser pour accéder à l'interface Web de phpmyadmin.

Tester la connexion à phpmyadmin avec un navigateur

	Lycée de l'Hyrôme - Chemillé	2009 – 2010
	Annexe 1 Création de certificat autosigné	Réseaux informatiques TP 3
Licence RII		

SSL/TLS Strong Encryption: FAQ

About Certificates

What are RSA Private Keys, CSRs and Certificates?

The RSA private key file is a digital file that you can use to decrypt messages sent to you. It has a public component which you distribute (via your Certificate file) which allows people to encrypt those messages to you. A Certificate Signing Request (CSR) is a digital file which contains your public key and your name. You send the CSR to a Certifying Authority (CA) to be converted into a real Certificate. A Certificate contains your RSA public key, your name, the name of the CA, and is digitally signed by your CA. Browsers that know the CA can verify the signature on that Certificate, thereby obtaining your RSA public key. That enables them to send messages which only you can decrypt. See the [Introduction](#) chapter for a general description of the SSL protocol.

Ok, I've got my server installed and want to create a real SSL server Certificate for it. How do I do it?

Here is a step-by-step description:

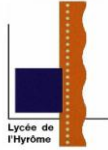
- ❖ Make sure OpenSSL is really installed and in your PATH. But some commands even work ok when you just run the ``openssl'' program from within the OpenSSL source tree as ``./apps/openssl''.
- ❖ Create a RSA private key for your Apache server (will be Triple-DES encrypted and PEM formatted):

```
$ openssl genrsa -des3 -out server.key 1024
```

Please backup this server.key file and remember the pass-phrase you had to enter at a secure location. You can see the details of this RSA private key via the command:

```
$ openssl rsa -noout -text -in server.key
```

And you could create a decrypted PEM version (not recommended) of this RSA private

	Lycée de l'Hyrôme - Chemillé	2009 – 2010
	Annexe 1 Création de certificat autosigné	Réseaux informatiques TP 3
Licence RII		

key via:

```
$ openssl rsa -in server.key -out server.key.unsecure
```

- ❖ Create a Certificate Signing Request (CSR) with the server RSA private key (output will be PEM formatted):

```
$ openssl req -new -key server.key -out server.csr
```

Make sure you enter the FQDN ("Fully Qualified Domain Name") of the server when OpenSSL prompts you for the "CommonName", i.e. when you generate a CSR for a website which will be later accessed via `https://www.foo.dom/`, enter "www.foo.dom" here. You can see the details of this CSR via the command

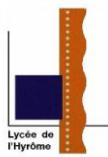
```
$ openssl req -noout -text -in server.csr
```

- ❖ You now have to send this Certificate Signing Request (CSR) to a Certifying Authority (CA) for signing. The result is then a real Certificate which can be used for Apache. Here you have two options: First you can let the CSR sign by a commercial CA like Verisign or Thawte. Then you usually have to post the CSR into a web form, pay for the signing and await the signed Certificate you then can store into a `server.crt` file. For more information about commercial CAs have a look at the following locations:

- Verisign
<http://digitalid.verisign.com/server/apacheNotice.htm>
- Thawte Consulting
<http://www.thawte.com/certs/server/request.html>
- CertiSign Certificadora Digital Ltda.
<http://www.certsign.com.br>
- IKS GmbH
<http://www.iks-jena.de/produkte/ca/>
- Uptime Commerce Ltd.
<http://www.uptimecommerce.com>
- BelSign NV/SA
<http://www.belsign.be>

Second you can use your own CA and now have to sign the CSR yourself by this CA. Read the next answer in this FAQ on how to sign a CSR with your CA yourself. You can see the details of the received Certificate via the command:

```
$ openssl x509 -noout -text -in server.crt
```

 Lycée de l'Hyrôme	Lycée de l'Hyrôme - Chemillé	2009 – 2010
	Annexe 1 Création de certificat autosigné	Réseaux informatiques TP 3
Licence RII		

- ❖ Now you have two files: server.key and server.crt. These now can be used as following inside your Apache's httpd.conf file:

```
SSLCertificateFile /path/to/this/server.crt
SSLCertificateKeyFile /path/to/this/server.key
```

The server.csr file is no longer needed.