Document d'audit du protocole SSL

Version	0.1
Date	11/02/2014
Rédigé par	Julien Legras
Relu par	
Approuvé par	

MISES À JOUR

Version	Date	Modifications réalisées
0.1	12/12/2013	Création du document

Table des matières

1 Objet

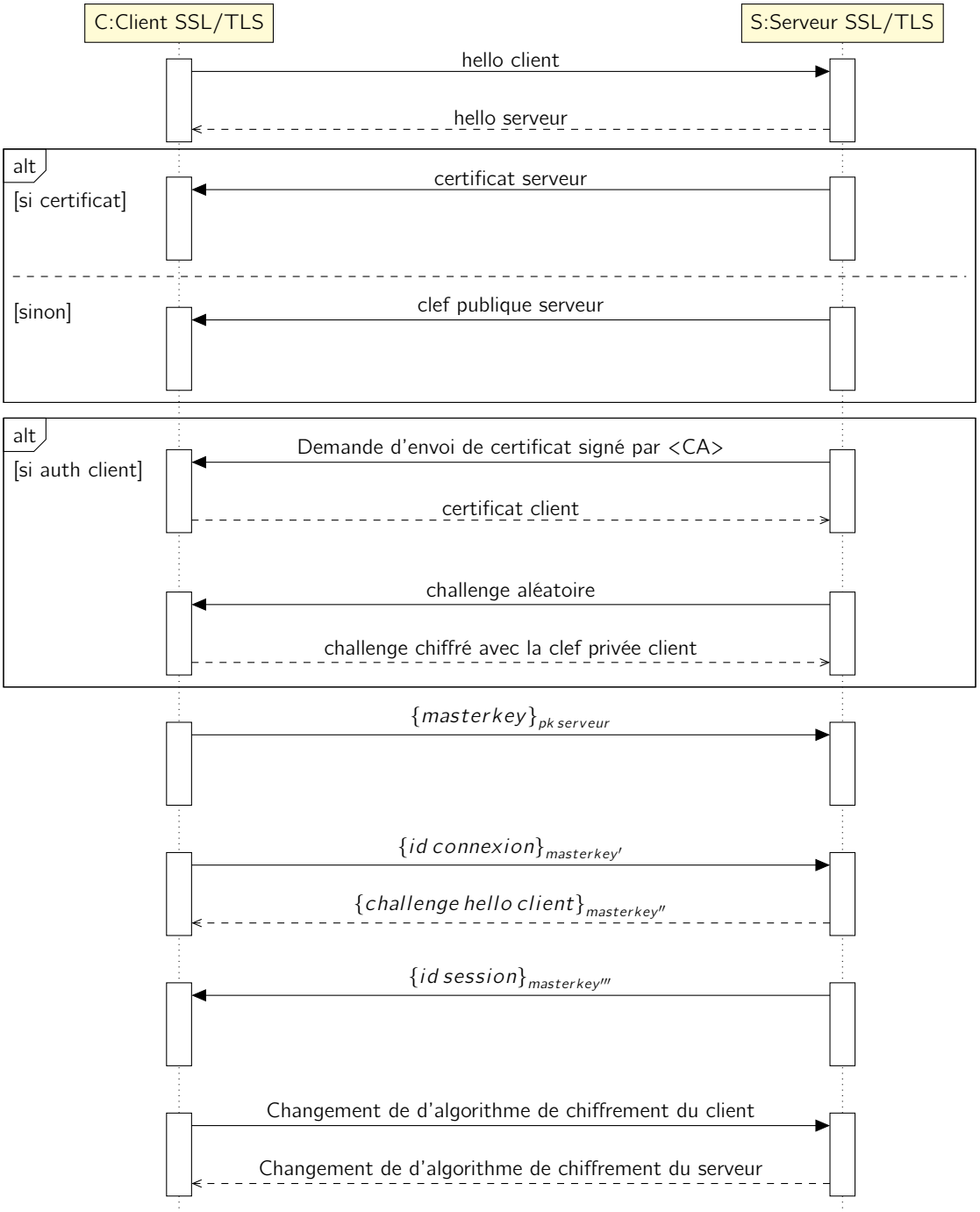
Ce document présente l'audit réalisé sur la partie SSL/TLS d'OpenSSL. Pour chaque partie, il y aura une présentation des recommandations/spécifications du protocole puis l'analyse du code associé dans OpenSSL.

2 Terminologie et sigles utilisés

— **RFC** : Les RFC (Request For Comments) sont un ensemble de documents qui font référence auprès de la Communauté Internet et qui décrivent, spécifient, aident à l'implémentation, standardisent et débattent de la majorité des normes, standards, technologies et protocoles liés à Internet et aux réseaux en général.
— **SSL** : Secure Sockets Layer
— **TLS** : Transport Layer Security
— **IETF** : The Internet Engineering Task Force est un groupe informel, international, ouvert à tout individu, qui participe à l'élaboration de standards Internet.
— **KeyExch** : Échange de clef
— **Authn** : Authentification
— **Enc** : Chiffrement
— **MAC** : Message Authentication Code

3 Schéma global d'une connexion SSL/TLS

Pour simplifier la compréhension des parties suivantes, voici un schéma qui représente de manière large l'établissement d'une connexion SSL/TLS. Les données entre accolades sont chiffrées avec la clef indiquée en indice. La *masterkey* est la clef principale qui sera dérivée pour chiffrer chaque message.



hello client Version du protocole SSL avec laquelle le client souhaite communiquer, challenge, algorithmes de chiffrement supportés par le client, méthodes de compressiosn supportées par le client.

hello serveur Version du protocole SSL calculée par le serveur (plus haute version du serveur supportée également par le client), challenge, id de session, algorithmes de chiffrement supportés par le serveur,

méthodes de compressiosn supportées par le serveur.

Sources = `http://www.symantec.com/connect/articles/apache-2-ssltls-step-step-part-1`, `http://datatracker.ietf.org/doc/rfc6101/?include_text=1`

4 SSL version 2

4.1 Spécifications

Il n'existe pas de RFC pour SSL version 2. En effet, ce protocole a été pensé et développé par la société Netscape Communications. Cette version est sortie en 1994. Toutefois, on trouve des morceaux d'informations dans certaines RFC (6176) et le draft de Hickman (`http://tools.ietf.org/html/draft-hickman-netscape-ssl-00`).

Algorithmes supportés

Identifiant	KeyExch	Authn	Enc	MAC
SSL_CK_RC2_128_CBC_WITH_MD5	RSA	RSA	RC2.128 CBC	MD5
SSL_CK_RC2_128_CBC_EXPORT40_WITH_MD5	RSA.512	RSA	RC4.40 CBC	MD5
SSL_CK_IDEA_128_CBC_WITH_MD5	RSA	RSA	IDEA.128 CBC	MD5
SSL_CK_DES_64_CBC_WITH_MD5	RSA	RSA	DES.56 CBC	MD5
SSL_CK_DES_192_EDE3_CBC_WITH_MD5	RSA	RSA	3DES.168 CBC	MD5
SSL_CK_RC4_128_WITH_MD5	RSA	RSA	RC4.128	MD5
SSL_CK_RC4_128_EXPORT40_WITH_MD5	RSA.512	RSA	RC4.40	MD5

Remarque Le `CK` signifie `CIPHER-KIND`.

4.2 Implémentation

Dans le code d'OpenSSL, cette version du protocole SSL se trouve dans les fichiers commençant pas `s2_` du répertoire `ssl/`. Les constantes sont déclarées dans le fichier `ssl2.h`, on retrouve bien les algorithmes du draft :

Identifiant	Constante OpenSSL
SSL_CK_RC2_128_CBC_WITH_MD5	SSL2_CK_RC2_128_CBC_WITH_MD5
SSL_CK_RC2_128_CBC_EXPORT40_WITH_MD5	SSL2_CK_RC2_128_CBC_EXPORT40_WITH_MD5
SSL_CK_IDEA_128_CBC_WITH_MD5	SSL2_CK_IDEA_128_CBC_WITH_MD5
SSL_CK_DES_64_CBC_WITH_MD5	SSL2_CK_DES_64_CBC_WITH_MD5
SSL_CK_DES_192_EDE3_CBC_WITH_MD5	SSL2_CK_DES_192_EDE3_CBC_WITH_MD5
SSL_CK_RC4_128_WITH_MD5	SSL2_CK_RC4_128_WITH_MD5
SSL_CK_RC4_128_EXPORT40_WITH_MD5	SSL2_CK_RC4_128_EXPORT40_WITH_MD5

On y trouve également des constantes non définies dans le draft avec des commentaires très succincts :

- `SSL2_CK_NULL_WITH_MD5 /* v3 */`
- `SSL2_CK_DES_64_CBC_WITH_SHA /* v3 */`
- `SSL2_CK_DES_192_EDE3_CBC_WITH_SHA /* v3 */`
- `SSL2_CK_RC4_64_WITH_MD5 /* MS hack */`
- `SSL2_CK_DES_64_CFB64_WITH_MD5_1 /* SSLeay */`
- `SSL2_CK_NULL /* SSLeay */`

Les constantes commentées avec `v3` sont présentes pour des raisons de rétro-compatilibité depuis SSL v3. Celles commentées par `SSLeay` sont des vestiges de l'ancêtre d'OpenSSL : SSLeay. Elles sont sûrement conservées pour la rétro-compatibilité avec des vieux logiciels utilisant SSLeay. La `MS hack` est spécifique à Windows

5 SSL version 3

5.1 Spécifications

La version 3 du protocole SSL est décrite dans la RFC 6101. On y trouve notamment en section A.6 la liste des algorithmes de chiffrement pouvant être utilisés avec cette version :

Algorithmes supportés

Identifiant	KeyExch	Authn	Enc	MAC
SSL_NULL_WITH_NULL_NULL	NULL	NULL	NULL	NULL
SSL_RSA_WITH_NULL_MD5	RSA	RSA	NULL	MD5
SSL_RSA_WITH_NULL_SHA	RSA	RSA	NULL	SHA1
SSL_RSA_EXPORT_WITH_RC4_40_MD5	RSAex	RSAex	RC4.40	MD5
SSL_RSA_WITH_RC4_128_MD5	RSA	RSA	RC4.128	MD5
SSL_RSA_WITH_RC4_128_SHA	RSA	RSA	IDEA.128	SHA1
SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5	RSAex	RSAex	RC2.40 CBC	MD5
SSL_RSA_WITH_IDEA_CBC_SHA	RSA	RSA	IDEA.128 CBC	SHA1
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	RSAex	RSAex	DES.40	SHA1
SSL_RSA_WITH_DES_CBC_SHA	RSA	RSA	DES.56 CBC	SHA1
SSL_RSA_WITH_3DES_EDE_CBC_SHA	RSA	RSA	3DES.168 CBC	SHA1
SSL_DH_DSS_EXPORT_WITH_DES40_CBC_SHA	DH	DSS	DES.40 CBC	SHA1
SSL_DH_DSS_WITH_DES_CBC_SHA	DH	DSS	DES.56 CBC	SHA1
SSL_DH_DSS_WITH_3DES_EDE_CBC_SHA	DH	DSS	3DES.168 CBC	SHA1
SSL_DH_RSA_EXPORT_WITH_DES40_CBC_SHA	DH	RSA	DES.40 CBC	SHA1
SSL_DH_RSA_WITH_DES_CBC_SHA	DH	RSA	DES.56 CBC	SHA1
SSL_DH_RSA_WITH_3DES_EDE_CBC_SHA	DH	RSA	3DES.168 CBC	SHA1
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	DHE.512	DSS	DES.40 CBC	SHA1
SSL_DHE_DSS_WITH_DES_CBC_SHA	DHE	DSS	DES.56 CBC	SHA1
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA	DHE	DSS	3DES.168 CBC	SHA1
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	DHE.512	RSA	DES.40CBC	SHA1
SSL_DHE_RSA_WITH_DES_CBC_SHA	DHE	RSA	DES.56 CBC	SHA1
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	DHE	RSA	3DES.168 CBC	SHA1
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5	DH.512	None	RC4.40	MD5
SSL_DH_anon_WITH_RC4_128_MD5	DH	None	RC4.128	MD5
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA	DH.512	None	DES.40 CBC	SHA1
SSL_DH_anon_WITH_DES_CBC_SHA	DH	None	DES.56 CBC	SHA1
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA	DH	None	3DES.168 CBC	SHA1
SSL_FORTEZZA_KEA_WITH_NULL_SHA	FRTZA	KEA	None	SHA1
SSL_FORTEZZA_KEA_WITH_FORTEZZA_CBC_SHA	FRTZA	KEA	FRTZA	SHA1
SSL_FORTEZZA_KEA_WITH_RC4_128_SHA	FRTZA	KEA	RC4.128	SHA1

5.2 Implémentation

Dans le code d'OpenSSL, cette version du protocole SSL se trouve dans les fichiers commençant pas `s3_` du répertoire `ssl/`. Les constantes sont déclarées dans le fichier `ssl3.h`, on y retrouve les algorithmes de la RFC :

Identifiant	Constante OpenSSL
SSL_NULL_WITH_NULL_NULL	
SSL_RSA_WITH_NULL_MD5	SSL3_CK_RSA_NULL_MD5
SSL_RSA_WITH_NULL_SHA	SSL3_CK_RSA_NULL_SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5	SSL3_CK_RSA_RC4_40_MD5
SSL_RSA_WITH_RC4_128_MD5	SSL3_CK_RSA_RC4_128_MD5
SSL_RSA_WITH_RC4_128_SHA	SSL3_CK_RSA_RC4_128_SHA
SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5	SSL3_CK_RSA_RC2_40_MD5
SSL_RSA_WITH_IDEA_CBC_SHA	SSL3_CK_RSA_IDEA_128_SHA
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	SSL3_CK_RSA_DES_40_CBC_SHA
SSL_RSA_WITH_DES_CBC_SHA	SSL3_CK_RSA_DES_64_CBC_SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA	SSL3_CK_RSA_DES_192_CBC3_SHA
SSL_DH_DSS_EXPORT_WITH_DES40_CBC_SHA	SSL3_CK_DH_DSS_DES_40_CBC_SHA
SSL_DH_DSS_WITH_DES_CBC_SHA	SSL3_CK_DH_DSS_DES_64_CBC_SHA
SSL_DH_DSS_WITH_3DES_EDE_CBC_SHA	SSL3_CK_DH_DSS_DES_192_CBC3_SHA
SSL_DH_RSA_EXPORT_WITH_DES40_CBC_SHA	SSL3_CK_DH_RSA_DES_40_CBC_SHA
SSL_DH_RSA_WITH_DES_CBC_SHA	SSL3_CK_DH_RSA_DES_64_CBC_SHA
SSL_DH_RSA_WITH_3DES_EDE_CBC_SHA	SSL3_CK_DH_RSA_DES_192_CBC3_SHA
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	SSL3_CK_DHE_DSS_DES_40_CBC_SHA
SSL_DHE_DSS_WITH_DES_CBC_SHA	SSL3_CK_DHE_DSS_DES_64_CBC_SHA
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA	SSL3_CK_DHE_DSS_DES_192_CBC3_SHA
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	SSL3_CK_DHE_RSA_DES_40_CBC_SHA
SSL_DHE_RSA_WITH_DES_CBC_SHA	SSL3_CK_DHE_RSA_DES_64_CBC_SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	SSL3_CK_DHE_RSA_DES_192_CBC3_SHA
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5	SSL3_CK_ADH_RC4_40_MD5
SSL_DH_anon_WITH_RC4_128_MD5	SSL3_CK_ADH_RC4_128_MD5
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA	SSL3_CK_ADH_DES_40_CBC_SHA
SSL_DH_anon_WITH_DES_CBC_SHA	SSL3_CK_ADH_DES_64_CBC_SHA
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA	SSL3_CK_ADH_DES_192_CBC_SHA
SSL_FORTEZZA_KEA_WITH_NULL_SHA	SSL3_CK_FZA_DMS_NULL_SHA
SSL_FORTEZZA_KEA_WITH_FORTEZZA_CBC_SHA	SSL3_CK_FZA_DMS_FZA_SHA
SSL_FORTEZZA_KEA_WITH_RC4_128_SHA	SSL3_CK_FZA_DMS_RC4_SHA

Attention Les 3 algorithmes FORTEZZA sont commentés dans OpenSSL depuis le commit 89bbe14c506b9bd2fd00e6bae22a99ef1ee7ad19 de 2006.

Remarque OpenSSL déclare d'autre constantes pour utiliser SSL 3 avec Kerberos 5 :

- SSL3_CK_KRB5_DES_64_CBC_SHA
- SSL3_CK_KRB5_DES_192_CBC3_SHA
- SSL3_CK_KRB5_RC4_128_SHA
- SSL3_CK_KRB5_IDEA_128_CBC_SHA
- SSL3_CK_KRB5_DES_64_CBC_MD5
- SSL3_CK_KRB5_DES_192_CBC3_MD5
- SSL3_CK_KRB5_RC4_128_MD5
- SSL3_CK_KRB5_IDEA_128_CBC_MD5

- SSL3_CK_KRB5_DES_40_CBC_SHA
- SSL3_CK_KRB5_RC2_40_CBC_SHA
- SSL3_CK_KRB5_RC4_40_SHA
- SSL3_CK_KRB5_DES_40_CBC_MD5
- SSL3_CK_KRB5_RC2_40_CBC_MD5
- SSL3_CK_KRB5_RC4_40_MD5

6 TLS version 1

6.1 Spécifications

6.2 Implémentation

7 TLS version 1.1

7.1 Spécifications

7.2 Implémentation

8 TLS version 1.2

8.1 Spécifications

8.2 Implémentation