Krótka instrukcja na temat tego jak używać protokołu SSL w Javie

Java dostarcza protokół SSL za pośrednictwem biblioteki Jsse (java security socket extensions). Razem z Jdk/Jre dostarczane jest narzędzie **keytool** do zarządzania kluczami w systemie. Żeby używać SSL musimy wygenerować parę kluczy wg. algorytmu RSA (bądź innego, mniej znanego i nie zawsze implementowanego). Możemy to zrobić np. tak:

C:\Users\Tomek\Desktop\Internet Voting System\System\SSLKeys>keytool -genkeypair -keyalg RSA -keystore MojMagazyn -alias MojaPara -keypass 123456
Enter keystore password: 123456 #haslo dla magazynu aby nawet jesli ktos wykradnie go nie mial #latwego dostepu do klucza prywatnego
Re-enter new password: 123456 #haslo dla magazynu
What is your first and last name?
[Unknown]: Maciek i Tomek
What is the name of your organizational unit?
[Unknown]: Instytut Informatyki UJ
What is the name of your organization?
[Unknown]: Uniwersytet Jagiellonski
What is the name of your City or Locality?
[Unknown]: Krakow
What is the name of your State or Province?
[Unknown]: Malopolskie
What is the two-letter country code for this unit?
[Unknown]: MP
Is CN=Maciek i Tomek, OU=Instytut Informatyki UJ, O=Uniwersytet Jagiellonski, L=Krakow, ST=Malopolskie, C=MP correct?
[no]: yes

Plik mozna odnalezc w katalogu roboczym - tutaj jest to: C:\Users\Tomek\Desktop\Internet Voting System\System\SSLKeys.

Można to było zrobić i prościej – nie podając parametru alias - wtedy wpis w magazynie (para kluczy) przyjmie domyślna nazwę mykey, jednak musimy brać pod uwagę ewentualny konflikt jeśli magazyn będzie zawierał więcej wpisów które nie specyfikują tego parametru.

Po stronie klienta wymagany jest tylko certyfikat. Można go uzyskać z pary kluczy poleceniem keytool -exportcert jak nizej:

C:\Users\Tomek\Desktop\Internet Voting System\System\SSLKeys>keytool -exportcert -keystore MojMagazyn -file mojcert.cert -alias mojapara
Enter keystore password:
Certificate stored in file <mojcert.cert>

Tak uzyskany certyfikat należy zaimportować do magazynu, który nie będzie zawierał pary kluczy i certyfikatu ale jedynie sam certyfikat (tak należy to robić w rzeczywistości gdyż inaczej SSL nie różniłby się niczym od protokołu wykorzystującego jedynie klucze sekretne/symetryczne). Można to zrobić tak:

C:\Users\Tomek\Desktop\Internet Voting System\System\SSLKeys>keytool -importcert -keystore mojmagcertyfikatow -file mojcert.cert

Enter keystore password: 123456
Re-enter new password:
Owner: CN=Maciek i Tomek, OU=Instytut Informatyki UJ, O=Uniwersytet Jagiellonski
, L=Krakow, ST=Malopolskie, C=MP
Issuer: CN=Maciek i Tomek, OU=Instytut Informatyki UJ, O=Uniwersytet Jagiellonsk
i, L=Krakow, ST=Malopolskie, C=MP
Serial number: 48f857c2
Valid from: Tue Jan 15 23:20:00 CET 2013 until: Tue Apr 16 00:20:00 CEST 2013
Certificate fingerprints:
MD5: 5E:87:E7:35:0C:B2:CC:F0:50:FF:BA:12:41:09:0F:30
SHA1: 59:52:55:56:65:79:93:F4:83:AB:39:4D:BA:35:00:50:EB:89:F2:33
SHA256:
0C:B3:C5:DC:33:53:2A:B3:14:6F:40:31:E7:58:E7:70:13:1F:71:F5:0E:5A:38:F2:14:CF:52:F5:EB:
E0:94:9E
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: D5 0D 76 AA 51 B7 B2 10 73 D2 B2 71 15 79 A6 BD ..v.Q...s..q.y..
0010: EB 44 11 E2 .D..
]
]

Trust this certificate? [no]: yes
Certificate was added to keystore

Mając osobny magazyn certyfikatów nie zawierający klucza prywatnego, a taki należy z ideologicznego punktu widzenia użyć dla klienta, pomimo tego że u nas ten z kluczem prywatnym, który używamy dla serwera także zadziałałby gdyż zawiera on wszystkie informacje łącznie z certyfikatem – gdyby ktoś chciał użyć magazynu certyfikatów dla serwera również dla klienta, to Jsse samo wyodrębni z niego potrzebny mu certyfikat, natomiast w praktyce powinien być to sam certyfikat.

Oto jak postępujemy w przypadku naszego programu:

C:\Users\Tomek\Desktop\Internet Voting System\System\SSLKeys>keytool -genkeypair -keyalg RSA -keystore CLAKeyStore
Enter keystore password: 123456
Re-enter new password:
What is your first and last name?
[Unknown]: CLA
What is the name of your organizational unit?
[Unknown]: Voting Commision Of Future Poland
What is the name of your organization?
[Unknown]: Government
What is the name of your City or Locality?
[Unknown]: Warsaw
What is the name of your State or Province?

[Unknown]: PL
What is the two-letter country code for this unit?
[Unknown]: PL
Is CN=CLA, OU=Voting Commision Of Future Poland, O=Government, L=Warsaw, ST=PL, C=PL correct?
[no]: yes

Enter key password for <mykey>
(RETURN if same as keystore password):

C:\Users\Tomek\Desktop\Internet Voting System\System\SSLKeys>keytool -genkeypair -keyalg RSA -keystore CTFKeyStore
Enter keystore password: 123456
Re-enter new password:
What is your first and last name?
[Unknown]: CTF
What is the name of your organizational unit?
[Unknown]: Voting Commision Of The Future Poland
What is the name of your organization?
[Unknown]: Government
What is the name of your City or Locality?
[Unknown]: Warsaw
What is the name of your State or Province?
[Unknown]: Warsaw
What is the two-letter country code for this unit?
[Unknown]: PL
Is CN=CTF, OU=Voting Commision Of The Future Poland, O=Government, L=Warsaw, ST=Warsaw, C=PL correct?
[no]: yes

Enter key password for <mykey>
(RETURN if same as keystore password):

Tych kluczy uzywamy w projekcie

Podobnie generujemy magazyn zawierający oba powyższe certyfikaty

C:\Users\Tomek\Desktop\Internet Voting System\System\SSLKeys>keytool -exportcert -keystore CTFKeyStore -file CTF.cert
Enter keystore password:
Certificate stored in file <CTF.cert>

C:\Users\Tomek\Desktop\Internet Voting System\System\SSLKeys>keytool -exportcert -keystore CLAKeyStore -file CLA.cert
Enter keystore password:
Certificate stored in file <CLA.cert>

C:\Users\Tomek\Desktop\Internet Voting System\System\SSLKeys>keytool -importcert -keystore VotComCertMag -file CTF.cert

Enter keystore password:
Re-enter new password:
Owner: CN=CTF, OU=Voting Commision Of The Future Poland, O=Government, L=Warsaw, ST=Warsaw, C=PL
Issuer: CN=CTF, OU=Voting Commision Of The Future Poland, O=Government, L=Warsaw, ST=Warsaw, C=PL
Serial number: 35928338
Valid from: Tue Jan 15 23:57:58 CET 2013 until: Tue Apr 16 00:57:58 CEST 2013
Certificate fingerprints:
MD5: 5E:E6:6B:74:16:02:A6:3D:E3:48:7C:C1:12:C7:31:DE
SHA1: A6:E9:B3:7C:70:27:6C:E0:DC:34:E6:FA:34:14:8A:D1:10:82:62:46
SHA256:
75:16:DD:40:25:7C:9F:57:2C:87:D1:17:71:A1:A0:87:42:DF:5A:1A:85:3D:99:E0:78:DE:4F:EA:A1:0E:C4:21
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: B8 93 F3 6C C4 6B AA 84 ED CE BD C7 A9 68 CE FB ...l.k.....h..
0010: 63 E8 60 40 c.`@
]
]

Trust this certificate? [no]: yes
Certificate was added to keystore

Jako że domyślna nazwa wpisu (alias) w magazynie od początku była równa mykey, w obu przypadkach wiec chcąc dodać każdy do tego samego magazynu trzeba nadać im wcześniej rożne nazwy (aliasy).

C:\Users\Tomek\Desktop\Internet Voting System\System\SSLKeys>keytool -changealias -keystore VotComCertMag -alias mykey -destalias ctfcert
Enter keystore password: 123456

C:\Users\Tomek\Desktop\Internet Voting System\System\SSLKeys>keytool -importcert -keystore VotComCertMag -file CLA.cert
Enter keystore password: 123456
Owner: CN=CLA, OU=Voting Commision Of Future Poland, O=Government, L=Warsaw, ST=PL, C=PL
Issuer: CN=CLA, OU=Voting Commision Of Future Poland, O=Government, L=Warsaw, ST=PL, C=PL
Serial number: 1f7d1a35
Valid from: Tue Jan 15 23:56:25 CET 2013 until: Tue Apr 16 00:56:25 CEST 2013
Certificate fingerprints:
MD5: 6B:3E:87:7D:FF:EC:C1:CC:E2:3F:9B:DD:1D:BA:44:27
SHA1: 9A:89:97:52:7B:CA:C1:FC:E4:6F:1F:5B:91:D4:8F:5C:74:1C:06:D0
SHA256:

19:42:9F:96:15:2E:AE:DA:32:F3:CD:B5:57:D8:47:64:29:FB:3C:53:AE:33:24:C5:A1:E8:3D:77:93
:3E:EF:8D
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 07 A3 50 CC 71 37 83 6E A8 CF 68 0B 78 37 15 0E ..P.q7.n..h.x7..
0010: 57 9E 50 2D W.P-
]
]

Trust this certificate? [no]: yes
Certificate was added to keystore

C:\Users\Tomek\Desktop\Internet Voting System\System\SSLKeys>keytool -changealias -keystore VotComCertMag -alias mykey -destalias clacert
Enter keystore password:

a teraz wyswietlmy zawartosc naszego magazynu ktory powinien miec zaimportowany kazdy klient:

C:\Users\Tomek\Desktop\Internet Voting System\System\SSLKeys>keytool -list -keystore VotComCertMag
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 2 entries

ctfcert, 2013-01-16, trustedCertEntry,
Certificate fingerprint (SHA1): A6:E9:B3:7C:70:27:6C:E0:DC:34:E6:FA:34:14:8A:D1:
10:82:62:46
clacert, 2013-01-16, trustedCertEntry,
Certificate fingerprint (SHA1): 9A:89:97:52:7B:CA:C1:FC:E4:6F:1F:5B:91:D4:8F:5C:
74:1C:06:D0

Czyli widać ze ok.

Na tę chwilę powinniśmy mieć już magazyn certyfikatów wygenerowany z magazynów kluczy odpowiednich serwerów. Taki magazyn, lub jego nadzbiór musi znaleźć się u każdego klienta aby połączenia z serwerami poprzez protokół SSL zostały uwierzytelnione. W samym programie Javowym biblioteka Jsse wykorzystuje jednak jeszcze pewne zmienne systemowe, które umożliwiają jej znalezienie rzeczonych certyfikatów. Trzeba rozważyć zmienne dla klienta i serwera.

1) **SERWER** – jego gniazdo będzie szukać pary kluczy w magazynie - trzeba mu ja podać.

Możliwości są dwie:

1) modyfikujac zmienne systemowe w kodzie programu np jak tutaj:

```
System.setProperty("javax.net.ssl.keyStore", "..\\..\\..\\System\\SSLKeys\\MojMagazyn");
System.setProperty("javax.net.ssl.keyStorePassword", "123456");
```

2) podajac zmienne systemowe przy uruchamianiu programu za pomocą parametru -D:

```
-Djavax.net.ssl.keyStore=..\\..\\..\\mySrvKeystore -D...
```

UWAGA: Trzeba wziąć pod uwagę, że w obu przypadkach katalogi robocze mogą się różnic, co może prowadzić do nieznalezienia magazynu i błędu podczas próby użycia gniazd.

2) **KLIENT** - potrzebuje bazy certyfikatów którym może zaufać. W naszym przypadku jest to samopodpisujący się certyfikat, który jest generowany w tle z pary kluczy. Odkąd zostanie wskazany jako tzw. magazyn zaufania (zaufanych certyfikatów) rogram traktuje go jakby był wystawiony przez urząd wystawiający certyfikaty. Podobnie jak poprzednio mamy dwie możliwości:

1) wewnątrz programu:

```
System.setProperty("javax.net.ssl.trustStore", "..\\..\\..\\System\\SSLKeys\\MojMagazyn");
System.setProperty("javax.net.ssl.trustStorePassword", "123456");
```

2) podczas jego uruchamiania

```
java -Djavax.net.ssl.trustStore=..\\..\\..\\System\\SSLKeys\\MojMagazyn
-Djavax.net.ssl.trustStorePassword=123456 KlasaProgramu
```

Trzeba jednak liczyć się z tym, że w obu przypadkach inny będzie domyślny katalog roboczy - dlatego nie należy powyższych ścieżek brać jako wykładni!!