

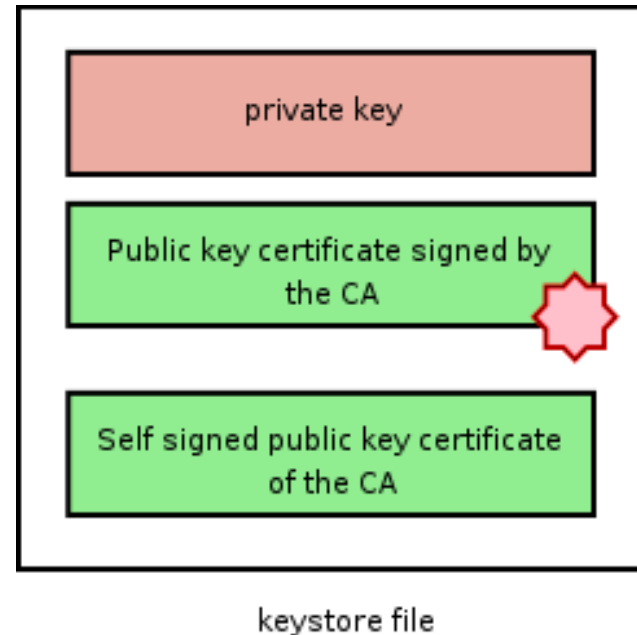
Key Management

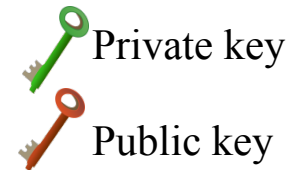
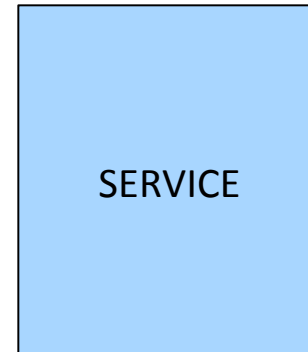


© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See <http://creativecommons.org/licenses/by-sa/3.0/>

Key manipulation

- JDK keytool
- OpenSSL
- A keystore
 - Private Keys
 - Certificates
- Types
 - PKCS12
 - JKS





- Client and service should share each other's public keys



Requirement

- Two keystores for the service and client
- Signed by a CA
- Make sure you have
 - OpenSSL
 - JDK



Simple Certificate Authority

- `$ openssl req -x509 -newkey rsa:1024 \`
- `-keyout cakey.pem -out cacert.pem -config openssl.cnf`

- This creates



cakey.pem



cacert.pem



© Paul Fremantle 2012. Portion

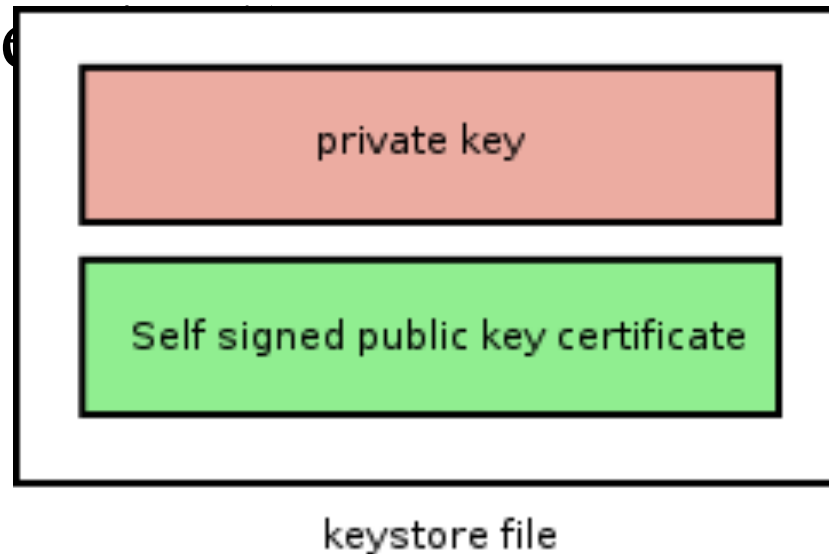
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.

See <http://creativecommons.org/licenses/by-sa/3.0/>

mission of the author(s).

Client and Service Keys

- `$ keytool -genkey -alias client -keyalg RSA -keystore client.jks`
- `$ keytool -genkey -alias service -keyalg RSA -keystore service.jks`



Now we need our CA to sign the public keys



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See <http://creativecommons.org/licenses/by-sa/3.0/>

Certificate Signing Request (CSR)

- `$ keytool -certreq -keystore client.jks -storepass changeme -alias client -file client.cert.req`
- `$ keytool -certreq -keystore service.jks -storepass changeme -alias service -file service.cert.req`



Sign the CSRs

- `$ openssl ca -config openssl.cnf -out client.pem -infiles client.cert.req`
- `$ openssl ca -config openssl.cnf -out service.pem -infiles service.cert.req`
- And convert the signed certificates and CA cert to DER
- `$ openssl x509 -outform DER -in client.pem -out client.cert`
- `$ openssl x509 -outform DER -in service.pem -out service.cert`
- `$ openssl x509 -outform DER -in cacert.pem -out cacert.cert`



Import Certificates

- Import CA certificate
- `$ keytool -import -file cacert.cert -keystore service.jks -storepass changeme -alias ca`
- `$ keytool -import -file cacert.cert -keystore client.jks -storepass changeme -alias ca`
- Import signed certificates
- `$ keytool -import -file client.cert -keystore client.jks -storepass changeme -alias client`
- `$ keytool -import -file service.cert -keystore service.jks -storepass changeme -alias service`

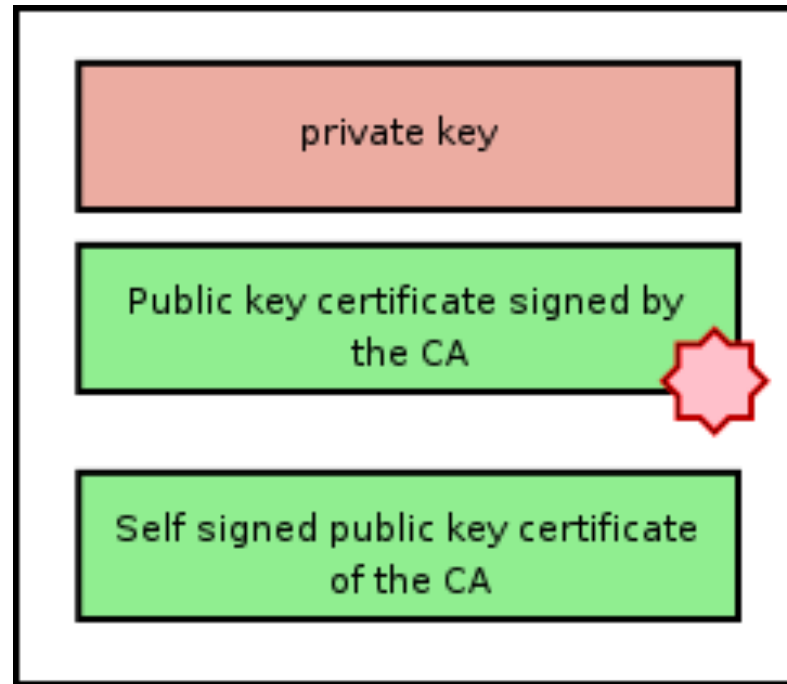


Importing Peer Certificates

- To be able to directly trust the other
- `$ keytool -import -file client.cert -keystore service.jks -storepass changeme -alias client`
- `$ keytool -import -file service.cert -keystore client.jks -storepass changeme -alias service`



Now our service and client keystores are ready!



keystore file



© Paul Fremantle 2012. Portion

Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.

See <http://creativecommons.org/licenses/by-sa/3.0/>

of the author(s).