NAME

clkeys – public/private key management tool interoperating with Thunderbird and OpenSSL

SYNOPSIS

clkeys generate [-DSA] KeysetName [-SIZE RSABits] [-CN YourName]

clkeys request KeysetName [-CN YourName]

clkeys import KeysetName CertFile

clkeys casign CA-KeysetName RequestFile

clkeys list KeysetName

DESCRIPTION

clkeys generates RSA and DSA asymmetric keys and stores keys in a file in the PKCS#15 token format.

As both OpenSSL and standard E-mail clients expect private RSA keys in a file based on the PKCS#12 format, keys generated by Cryptlib cannot be exchanged between **clkeys** and those programs. Many will see this as a disadvantage or detriment and maybe even as a reason not to use Cryptlib, but this assessment is totally incorrect.

There is a reason why Cryptlib does not provide any other file storage of private keys and a very important one. Storing private keys in the PKCS#12 format exposes the key material to a number of weaknesses, described in a paper here:

<https://www.usenix.org/legacy/event/sec02/full_papers/gutmann/gutmann.pdf>

In contrast, one of the principal design features of Cryptlib is that it never exposes private keys to outside access. So do not wait (or even wish) for a conversion tool that makes Cryptlib-generated keys available in a p12 file. It makes no sense to deliberately reduce the security of your private keys just because some E-mail client wants them in a weak format only.

However, even if standard E-mail programs cannot import Cryptlib-generated keys (which can be seen as a missing feature) a comprehensive exchange of SMIME messages is nevertheless possible. Certificates based on requests created by **clkeys** can be imported into standard E-mail programs directly. And any messages in SMIME format can be decrypted and signed with clkeys-generated keys using **clsmime**.

In addition to generating RSA and DSA public key pairs, clkeys can generate certificate signing requests for a public key and also import self-signed certificates or certificates signed by a CA into the p15 keyset file if a corresponding private key is found in the keyset file.

When **clkeys** generates a key pair, a self-signed certificate is created that can be used for all kinds of purposes (Digital Signature, Non Repudiation, Key Encipherment, Certificate Sign, CRL Sign) and has a very long lifespan (20 years). Any of these self-signed certificates can be used together with the corresponding private key to sign certificate requests for other keys used by PKI users for email or other purposes.

These self-signed certificates can also be imported in the CA section of E-mail clients like Thunderbird, but they are not fit for identifying persons, as their Basic Constraint is set to CA:TRUE and they don't have an email address included.

The process of producing valid email certificates for persons is described below.

OPTIONS

−DSA generate a DSA public key pair instead of the (default) RSA keys

−RSABits number
specify the number of bits for the RSA modulus (between 1536 and 4096 bits)

−CN common name
without this option the generated key is stored under the KEYID identical to the name of the keyset file. To store the keys (private and public) under a different Name comprised of several distinct name components (like James O´Connor Jr) the option -CN can be used. This common name is also necessary to be included in a certificate signing request. The character ’ must be escaped on the command line.

NOTES

Full documentation <https://senderek.ie/cryptlib/tools>

This program depends on two packages providing the cryptlib shared object library and the python3-bindings to this library.

You can download both packages in RPM or DEB format at https://senderek.ie/cryptlib/downloads

Using FEDORA you can install the packages cryptlib and cryptlib-python3 directly from the repository.

In addition the program /bin/systemd-ask-password is needed to read sensible data from stdin. This program is part of the systemd package. And to be able to list the ASN1 structure of p15 keyset files, the program /bin/dumpasn1 is needed.

INTEROPERABILITY

Thunderbird E-mail client (user certificates)

Thunderbird does accept certificates only when an email address is present as a part of the CN (common name) or stored in the SAN (subject alternative name). As **clkeys** always uses the SAN to store the email address in certificate signing requests (CSR), the CSR should normally be signed by a CA key made by **clkeys**. This will ensure, that the user can import the resulting certificate directly into an E-mail client. The user’s certificate will contain the email address in the SAN field.

To sign a CSR with the CA signing key generated by clkeys in a p15 keyset you will use the following steps:

```
1) clkeys generate CAkey −SIZE 4096 −CN Your CAname

2) clkeys import CAkey CAkey.cert.pem
   # without importing the self−signed cert the CA private key cannot sign

3) clkeys generate Joe −SIZE 2678 −CN Joe Doe Sen

4) clkeys request Joe −CN Joe Doe Sen
   # here some additional information about the certificate must be entered

5) clkeys casign CAkey Joe.CSR.der
```

The resulting file Joe.newcert.pem can then be used to be imported into Thunderbird in the persons section. Make sure the file ’CAkey.cert.pem’ is already imported in the CA section and proper permissions are being set.

On the other hand, if the user's certificate is being produced by an external CA that does accept only fields of the DN in the request (and not the SAN), the resulting certificate will not have an email address and is useless for import into E-mail clients. The external CA should therefore use the option "subjectAltName=email:${ENV::SAN}" in the [usr_cert] section of its config file and provide the email address via the environment variable SAN.

To extract the *subjectAltName* from a request file (in DER format), you can use the program here: https://senderek.ie/cryptlib/tools/selectSAN

All SMIME encrypted or signed messages produced by Thunderbird can be handled by a separate program **clsmime** that uses keyset files generated by **clkeys**.

OpenSSL

clkeys generates self-signed certificates for RSA and DSA keys and certificate signing requests that can be processed by OpenSSL in PEM and DER format in the usual way.

FILES

/usr/bin/systemd-ask-password
: This program is used to provide the passphrase based on a user's input.

/usr/bin/dumpasn1
: This program is used to list the ASN1 structure of p15 keyset files.

/lib64/libcl.so.3.4.8
: The cryptlib library.

/usr/lib/python3.1x/site-packages/cryptlib_py.so
: Bindings to the cryptlib library used by python3.

BUGS

Please report bugs to innovation@senderek.ie

AUTHORS

clkeys is written by Ralf Senderek <innovation@senderek.ie>.
Cryptlib is written and maintained by Peter Gutmann <pgut001@cs.auckland.ac.nz>

COPYRIGHT

SEE ALSO

cryptlib, clsmime, claes