

NZ SRS Registrar Kit Installation guide

April 2009

Table of Contents

1. Prerequisites.....	3
1.1 Operating System.....	3
1.2 Perl.....	3
1.3 Perl Modules.....	3
1.4 PGP Key.....	5
2. Untarring.....	6
3. Adding modules to include path.....	6
4. Viewing Documentation.....	7
5. Running the client.....	7

1. Prerequisites

1.1 Operating System

The Client and Modules have been tested on Debian GNU/Linux 4.0 (Etch). For compatibility reasons, we recommend the use of this OS. However, other recent distributions of GNU/Linux should also be acceptable. No testing has been done in Windows environments.

1.2 Perl

The Registrar Kit requires Perl version 5.6.1. Earlier 5.x versions could possibly work, but they are untested with the Registrar Kit.

1.3 Perl Modules

Below is a list of the modules required. Some of these can be installed with Debian packages, while others must be installed off CPAN. In all cases, the latest version should be used (note that some debian package versions are older than the CPAN version. In these cases, either version can be used). On non-Debian system, use CPAN modules, not RPMs.

To install a Debian package type (as root):

```
apt-get install <package-name>
```

To install a CPAN module type (as root):

```
perl -MCPAN -eshell
```

And then at the cpan command prompt:

```
install <module-name>
```

(Note, if this is the first time you have used the CPAN module on this system, you will be prompted for configuration information. Follow the instructions to configure the CPAN Module).

Alternatively, CPAN modules can be installed by manually downloading them from <http://search.cpan.org> or another CPAN member site, untarring, and typing:

NZ SRS registrar kit installation guide

```
perl Makefile.PL
make
make test
make install
```

Modules should be installed in the order they are listed in.

- MIME::Base64 (debian pkg: perl)
- Compress::Zlib (debian pkg: libcompress-zlib-perl)
- Data::Buffer (debian pkg: libdata-buffer-perl)
- Math::Pari (see note below)
- Crypt::Blowfish (debian pkg: libcrypt-blowfish-perl)
- Crypt::DES (debian pkg: libcrypt-des-perl)
- Crypt::DES_EDE3 (debian pkg: libcrypt-des-ede3-perl)
- Crypt::RIPEMD160
- Digest::MD2 (debian pkg: libdigest-md2-perl)
- Digest::MD5 (debian pkg: libdigest-md5-perl)
- Digest::SHA1 (debian pkg: libdigest-sha1-perl)
- Crypt::DSA
- Crypt::RSA
- Class::Loader
- Crypt::Random
- Convert::ASN1 (debian pkg: libconvert-asn1-perl)
- Convert::PEM
- Convert::ASCII::Armour
- Crypt::CBC
- Sort::Versions
- Tie::EncryptedHash
- Crypt::Primes
- Crypt::SSLeay (debian pkg: libcrypt-ssleay-perl)
- Date::Calc (debian pkg: libdate-calc-perl)
- Math::BigInt (debian pkg: perl-modules)
- Math::BigInt::GMP (debian pkg: libmath-bigint-gmp-perl)
- Crypt::OpenPGP
- HTML::Tagset (debian pkg: libhtml-tagset-perl)
- HTML::HeadParser (debian pkg: libhtml-parser-perl)
- LWP (debian pkg: libwww-perl)
- Devel::StackTrace (debian pkg: libdevel-stacktrace-perl)
- Log::Agent (debian pkg: liblog-agent-perl)
- Date::Parse
- Time::Timezone (debian pkg: libdatetime-timezone-perl)

The Crypt::DSA module has a couple of important bugs filed against it - the most relevant being <http://rt.cpan.org/Public/Bug/Display.html?id=21968>. The fact that this bug is outstanding means that on many systems the OS is not able to produce enough entropy to sign a requests immediatly. This has been noted as a significant performance problem for some registrars - and in fact the SRS servers run a modified version of this package to avoid the problem

NZ SRS registrar kit installation guide

(we use '/dev/urandom' in place of '/dev/random')

The Math::Pari module must link to the Pari C module. If installing Math::Pari through the CPAN module, Pari C should be downloaded automatically. If installing manually, download the Math::Pari module, untar it, then download the latest stable release of Pari from <http://pari.math.u-bordeaux.fr/download.html> (or do a web search on 'Pari' if this link no longer works), and untar into a directory at the same level as Math::Pari. The Makefile.PL script should find the Pari sources when run.

LibXML and related perl modules also need installing, all of which are in Debian Etch repositories:

- libxml2
- libxml-libxml-perl
- libxml-namespacesupport-perl
- libxml-sax-perl

On non-Debian systems, the GNOME libxml libraries must be downloaded and installed from <http://www.xmlsoft.org/>. The following packages can then be installed through CPAN:

- XML::LibXML
- XML::SAX
- XML::NamespaceSupport

(Note: previous versions used the XML::Xerces module. This has been replaced with XML::LibXML).

1.4 PGP Key

In order to connect to the server, you must be set up as an authorised registrar. This requires a PGP key to be generated. The public key needs to be placed in the registrar account in the server.

It is recommended that the GnuPG tool be used for this (debian package: gnupg, <http://www.gnupg.org/>).

Make sure all the following commands are executed as the user that will be running the command line client, or any of the SRS::Client modules.

To generate a key, type:

```
gpg --gen-key
```

Follow the instructions the the gpg application gives you. Choose a 'DSA and ElGamal' type key, with keysize '2048', and '0' expiry (unless you have reason

NZ SRS registrar kit installation guide

to choose non-default settings). Leave the passphrase blank.

Once the key is generated, you can export it by typing:

```
gpg --export --armour <username>
```

Username is either the 'Real Name', 'Email Address' or both, that you entered for the key (type: 'gpg --list-keys' to view usernames for your keys). This is also the name you need to pass to the command line client, or the SRS::Client modules. (However, the most recently added secret key is your default secret key, and will be used if you don't specify a username).

The export command will print the armoured key to STOUT. If it's more convenient, you can redirect this to a file:

```
gpg --export --armour <username> > pub.key
```

You will also have to import the registry's public key to your keyring, so you can verify the signatures sent with responses by the registry. To do this, type:

```
gpg --import reg.key
```

The registry's public key should be included in this release. You will obviously have to specify the path to the key file if you're executing 'gpg' in a directory other than the one containing the key file.

Please note that the minimum PGP Key size we allow is '1024' bytes and NZRS recommend that a key size of '2048' bytes is used.

NZRS's Security Policy recommends that PGP keys used by Registrars to authenticate their SRS transactions have a maximum lifespan of two years. The key should have no expiry. Registrar's should use the RegistrarUpdate transaction to add a new key, check that using this new key works, and then send another RegistrarUpdate transaction to delete the old key.

If you have more than one key in your GPG keyring it may be necessary to specify which GPG identity should be used. Depending on how you are using the RIK there are a number of different ways this can be done:

- * For the sendXML program you can specify using the GNUPGID environment variable

- * For the SRSCient program you can specify a '-u' parameter

- * For the webserver you can specify an 'Id' value within the 'Crypto' block.

In all cases you should specify the real-name of the GPG id, not the fingerprint

2. Untarring

Move the registrar kit tar file to the desired installation directory and type

NZ SRS registrar kit installation guide

```
tar xvzf srs-rik-x.x.x.tar.gz
```

(Replace x.x.x with the version number)

3. Adding modules to include path

The directory structure under lib/perl5/ in the installation directory must be in the Perl include path. The easiest way to do this is to add the path to the PERL5LIB environment variable:

```
export PERL5LIB=/path/to/install/dir/lib/perl5
```

4. Viewing Documentation

If you have the perldoc tool installed (recommended) you can view the documentation on using the SRSCClient, by typing:

```
cd /path/to/install/dir  
perldoc SRSCClient
```

(The perldoc tool is in the debian 'perl-doc' package).

Alternatively, view the *.html documentation in the doc/ directory.

5. Running the client

The client can be run by typing

```
cd /path/to/install/dir  
./SRSCClient <arguments>
```