This directory contains the gen_ipext shell script, which can help with the process of generating certification paths suitable for use with SEND router discovery. This directory also contains the output of the example shown below. Note: the password for the CA's keying material in the example is “send”.

The script uses the term “id” as a handle to the certificate, keying material, and configuration files associated with a single entity. You set up ids by editing the script file itself; all material for each id is placed in a directory of the same name.

gen_ipext operates as follows:

```
  # gen_ipext chain
```

Generate a new certificate chain according to the configuration set in the script.

```
  # gen_ipext chain <new id> <signer id>
```

Generate a single new certificate in the chain.

```
  # gen_ipext ipext
```

Add IP extensions to a preexisting chain.

```
  # gen_ipext ipext <new id> <signer id>
```

Add an IP extension to a single certificate.

The 'chain' commands do generate new certificates, and then call the ipext commands. So if you already have a certificate chain, you can skip right to the ipext command (as long as your certificates and keying material is in the order the script needs).

You set the configuration for these operations by editing the script. At the top, there is a list 'ids' that contains the ids to be created or processed. This list is in the order of the certificate path. The CA is first, but you do not need to add it explicitly.

For each id in the list, you also need to provide the prefixes to be added to that id's certificate as IP extensions. Create a parameter of the form pfxs_$id that contains one or more prefixes of the form “prefix XXX::/64; YYY::/64; ...”. For example, to set the CA's authorized prefixes:

```
  pfxs_ca="prefix 2003::/64;
     prefix 2004::/64;
     prefix 2005::/64;"
```

Next set CA to the location of CA.pl(1) on your system, and rsa_bits to the desired RSA key size.

The following example will create these certification paths:

```
                              ar1
                             /
                            /
          ca------------lvl1---ar2
                            \
                             \
                              ar3
```

First we create a single certificate chain with the path ca -> lvl1 -> ar1, and later we will add ar2 and ar3. Set ids to

```
  ids=”lvl1 ar1”
```

Assign authorized prefixes to each id:

```
  pfxs_ca="prefix 2003::/64;
     prefix 2004::/64;
     prefix 2005::/64;"
  pfxs_lvl1="prefix 2003::/64;
     prefix 2004::/64;"
  pfxs_ar1="prefix 2003::/64;"
```

Now cd to wherever you want everything to be stored

```
  # cd /etc/sendd
```

Run gen_ipext:

```
  # /usr/src/send/examples/ipext/gen_ipext chain
  ****************************************************
  Making new top level CA
  ****************************************************
  CA certificate filename (or enter to create)
  <return>

  <... follow instructions to create the CA certificate>

  ****************************************************
  Creating certificate for lvl1
  ****************************************************

  <... follow instructions to create the certificate>
```

```
****************************************************
  Creating certificate for ar1
  ****************************************************

  <... follow instructions to create the certificate>
  <don't enter pass phrases for leaf nodes>

  Enter PEM pass phrase:
  <enter CA's pass phrase>
```

You should end up with the following files and directories:

```
ar1  demoCA      demoCA.ca    ipext_verify.conf  newreq.pem
ca   demoCA.ar1  demoCA.lvl1  lvl1
```

Each id subdirectory (ar1 and lvl1) contains a certificate with IP extensions (i.e. ar1/cert_ipext.pem), an RSA key (i.e. ar1/key.pem), and a ipext configuration file suitable for use with send (i.e. ar1/ipext.conf). Other files are not interesting – cert.pem is the certificate without IP extensions, and ipext_add.conf is the configuration file used to create cert_ipext.pem.

Now we will add ar2 and ar3. Edit ids to contain just the id we are adding:

```
  ids=”ar2”
```

Add a prefix definition for ar2:

```
  pfxs_ar2="prefix 2004::/64;"
```

Run gen_ipext with lvl1 as ar2's signer:

```
  # /usr/src/send/examples/ipext/gen_ipext chain ar2 lvl1

  <... follow instructions to create the certificate>
  <don't enter pass phrases for leaf nodes>
```

Repeat for ar3.

Now we can generate CGAs from the keys generated. The following generates a CGA for ar1:

```
  # cgatool --gen -p 2003:: -k ar1/key.pem -s 1 -o ar1/cga.params
  2003::3cb2:38a0:589c:4100
```

Finally, here is the sendd.conf for ar1:

```
snd_cga_params=/etc/sendd/ar1/params.conf
snd_pkixip_conf=/etc/sendd/ar1/ipext.conf
```

and /etc/sendd/ar1/params.conf:

```
named default {
     snd_cga_priv   /etc/sendd/ar1/key.pem;
     snd_cga_params /etc/sendd/ar1/cga.params;
     snd_cga_sec    1;
}
```