*** ON YOUR AUTHORITATIVE SERVER ***

1. Change to the directory where the zone resides, and make a backup
   of the zone (assuming it's called "mytld"), just in case

```
$ cd /etc/namedb/master
$ sudo cp mytld mytld.backup
```

  Also create a directory for the keys to live in, and let's create them

```
$ sudo mkdir /etc/namedb/keys
$ sudo chown bind /etc/namedb/keys

$ cd /etc/namedb/keys
```

2. Generate first key pair (Zone Signing Key)

```
$ sudo dnssec-keygen -a RSASHA1 -b 1024 -n ZONE mytld
```

    The output will be something like:

```
Generating key pair.....................+++++
+ ..................................................................................
.........................++++++
Kmytld.+005+51333
```

4. Generate second key pair (Key Signing Key)

```
$ sudo dnssec-keygen -f KSK -a RSASHA1 -b 2048 -n ZONE mytld
```

    Again you will see output similar to this:

```
Generating key pair.............................................++
+ ..........................................................................+++
Kmytld.+005+52159
```

4. Let's look at the keys:

```
# ls -l Kmytld.+005+5*
-rw-r--r--  1 root  wheel   203 Nov 29 00:07 Kmytld.+005+51333.key
-rw-------  1 root  wheel   937 Nov 29 00:07 Kmytld.+005+51333.private
-rw-r--r--  1 root  wheel   247 Nov 29 00:07 Kmytld.+005+52159.key
-rw-------  1 root  wheel  1125 Nov 29 00:07 Kmytld.+005+52159.private
```

4. Add the public keys to the end of the zone file:

    Edit the zone file for "mytld" and add the keys at the end:

```
$ cd /etc/namedb/master
```

    (edit the file "mytld" or whatever name you picked, and add the
    lines corresponding to your keys). To know which files to include:

```
$ ls -lC1 /etc/namedb/keys/K*key
```

    (copy the file names so you don't have to type then in by hand)

```
$ sudo ee mytld
```

```
    ; Keys to be published in DNSKEY RRset

    $include "/etc/namedb/keys/Kmytld.+005+51333.key"     ; ZSK
    $include "/etc/namedb/keys/Kmytld.+005+52159.key"     ; KSK

    Increment the serial number.
    Save and exit.

5. Sign the zone with the keys

    $ cd /etc/namedb/keys
    $ sudo dnssec-signzone -o mytld ../master/mytld

    The output will look something like:

Verifying the zone using the following algorithms: RSASHA1.
Zone signing complete:
Algorithm: RSASHA1: KSKs: 1 active, 0 stand-by, 0 revoked
                    ZSKs: 1 active, 0 stand-by, 0 revoked
../master/mytld.signed

    NOTE: that we don't need to specifiy which keys we'll be using - by
    default dnssec-signzone will sign using the keys it finds listed in the
    zone (those we added via the $include statement in step 4).

    If you wanted to explicitly specify which keys to use, you'd write
    something like (don't run this):

    $ sudo dnssec-signzone -o mytld -k Kmytld.+005+52159 ../master/mytld Kmytld.
+005+51333

    The signed zone has been written out in the master/ directory, so let's
    check it out:

    $ cd /etc/namedb/master/
    $ ls -l mytld*

    -rw-r--r--  1 root  wheel   292 Nov 29 00:08 mytld
    -rw-r--r--  1 root  wheel   292 Nov 29 00:10 mytld.backup
    -rw-r--r--  1 root  wheel  4294 Nov 29 00:20 mytld.signed

    Take a look at the zone contents, and observe the new records and
    signatures.

6. Notice that a set of DS records has been generated, and is ready to
   be communicated to your parent zone:

    $ cd /etc/namedb/keys/
    $ ls -l dsset-*

    -rw-r--r--  1 root  wheel  155 Nov 29 00:22 dsset-mytld.

   Look at the contents of the dsset:

    $ cat dsset-mytld.

    You should see two lines, one for each hashing algorithm used on
    the KSK.
```

7. Change the /etc/namedb/named.conf definition that loads the zone, to point to the signed zone:

    $ sudo ee /etc/namedb/named.conf

    ```
    zone "mytld" {
            type master;
            file "/etc/namedb/master/mytld.signed"; // load the signed zone
    };
    ```

8. Also in the named.conf, enable dnssec (for the authoritative part):

    ... in the options { .. }; section, add the following

    ```
    dnssec-enable yes;
    ```

9. Reconfigure/restart your nameserver

    $ sudo rndc reconfig

    You may also want to, but it is probably not necessary, do:

    $ sudo rndc reload mytld

    ... to "force" a reload of the zone. reconfig should normally do this, but it doesn't hurt :)

10. Test that the nameserver is answering with DNSSEC records:

    $ dig @127.0.0.1 mytld SOA +dnssec

11. Now you need to make sure that your slave has ALSO configured their nameserver to enable dnssec in their configuration (step 8). They should have done it since they are working on the same lab, but check anyway!

    To test:

    $ dig @10.20.Y.1 mytld SOA +dnssec

    ... where Y is the IP of the partner you picked to be slave for your domain - this could be the instructor, check with them.

12. You now need to communicate the DS to your parent

    Talk to your root manager on the way to communicate the DS.
    It could be using SCP or using a web interface.

    a) if using the RZM:

    Go to https://rzm.dnssek.org/

    Login (you should have signed up earlier)

    Check to see under Trust Anchor Details that your DS has automatically appeared AND matches. It is NOT automatically activated - the only thing the the RZM has done is "grab" the key from you and is waiting for your confirmation to enable th DS in the parent zone.

b) if not using the RZM:

If your root manager says to use scp, do as follows:

```
$ cd /etc/namedb/keys
$ scp dsset-mytld. sysadm@a.root-servers.net:
```

... this will copy the file "dsset-mytld." into the "sysadm" user directory on the a.root-server, where the root manager will include it into the root zone for signing.

Notify them when you have uploaded the file if using the "scp" method.

13. Wait a few minutes, until you are certain that the DS is included in the parent (root) zone.

    Then, using dig:

    dig @a.root-servers.net DS mytld.

    ... then you can begin to test validation!

14. Test that the AD bit is set:

    # dig @10.20.0.230 +dnssec www.MYTLD.

    Is it ?

    If not, note that the root manager may not have necessarily signed the root zone with your DS included yet, OR due to the *negative TTL*, the DS record may not be in the cache of the resolver. You may have to wait, but check with your root manager, and you can always check at the root:

    # dig @a.root-servers.net DS mytld.

    ... to verify that the DS is published. Then it's a matter of waiting for the cache to expire on the resolver, before you can verify your signatures.

    Alternatively, don't wait and proceed to enable validation in your own resolver (resolv.grpX.dns.nsrc.org) - see the relevant lab!