First Key Generation

Version:	165
Last modification:	Nov 17, 2011 16:25

Estimated time: 2 hours and 20 minutes (full procedure)

Roles

- KGA (Key Generation Administrator) facilitates key generation procedure and records data on their script copy
- SA (System Administrator) provides access to the signing box
- KSO (Keystore Security Officer) authorize keystore related operations, including backup and restoration
- DSO (Device Security Officer) authorize device related operations, including backup and restoration
- WI (Witness) attends the event as an observer.
- SAU (Security Auditor) reviews and audits the key generation procedure.

Abbreviations

TEB: Tamper-Evident Bag
MBC: Master Backup Copy
OBC: Operative Backup Copy
FD : Flash Drive

Materials

Description	**Quantity**
Laptop	1
CD with Live Linux Distribution	3
Projector	1
Printer	1
Photocopier	1
Flash Drives properly labeled and formatted	5
Spare formatted Flash Drives	2
Tamper-Evident bags	6
Pre-generated secure password for keystore user, device backup, and operations KSO	3
Sysadmin brings ssh key to access the signer	1
Hard copies of this script	12
Participant sign-in sheet	1

Participants

Title	**Org**	**Printed Name**	**Signature**	**Date**	**Time**
KGA	NZRS	Sebastian Castro		08/11/11	8:40
SA	Catalyst	James Dempsey		18/11/11	8:38
DSO1	NZRS	Dave Baker		18/11/11	08:35.
DSO2	Knossos	John Rumsey	John Rumsey	18/11/11	08:35
DSO3	Catalyst	Andrew Ruthven		18/11/11	08:40
DSO4	OSS	Vince Hag~~a~~n o		18/11	0835

DSO5	NZRS	Sebastian Castro		18/11/11	08:39
KSO1	NZRS	Dave Baker		18/11/11	08:35
KSO2	NZRS	Jay Daley		18/11/11	08.36
KSO3	NZRS	Doug Mercer		18/11/11	08.37
KSO4	NZRS	Richard Currey		18/11/11	08:35
KSO5	NZRS	Michael Wallmannsberger		18/11/11	08:34
WI1					
WI2					
SAU	Lateral Security	Israel Reyes		18/11/11	08:39

Safety Instructions

Estimated time: 5 min

Catalyst representative explains the safety procedures to follow in case of fire or earthquake, including Emergency Exits, Fire-fighting equipment and Assembly Point.

Internal Security Policy

Estimated time: 3 min

During the execution of this procedure, personal electronic devices may be used, as long as usage doesn't interfere with the normal course of the procedure. This includes mobile phones, laptops, etc. Mobile phones could be used to make phone calls in case of an emergency. One still camera may be present to take single images for archiving purposes. Video cameras and recording devices are not permitted.

Procedure

Initial preparation

Estimated time: 10 min

1. All the participants enter the room
2. KGA proceeds to validate the presence of all required participants
    3. Each participant will sign the KGA script copy. If the participant is not fulfilling a trusted role, it must provide a government-issued identification.
4. SA retrieves:
    5. Laptop (includes power cable, video cable, power extension)
    6. CD,
    7. Flash Drives
    8. Tamper-Evident Bags

Laptop setup

Estimated time: 15 min

9. SA sets up the laptop for the key generation procedure
    10. Connects power cable, network cable, and projector
    11. Boot-up laptop using a bootable CD
    12. Enables display
    13. Configures printer and print test page
    14. Open terminal, and maximize for visibility
15. SA verifies the integrity of the Live CD by comparing the digest 8:46

```
openssl dgst -c -sha256 /dev/sr0
SHA256(/dev/sr0)= f0:c1:51:a8:3a:4c:b3:ac:3d:26:16:f7:54:76:0e:78:
ba:47:5e:5a:12:4d:67:43:4b:c5:75:6e:26:19:3c:d3
```

TIME 8:53

16. SA verifies time and date on the laptop
17. KGA records date and time on their script copy

Date: 18 - Nov - 2011

Time: 8:53

Access to the signing box

Estimated time: 5 min

18. KGA selects Flash Drive labeled **Key Gen Log**, records the serial number on their script copy and hands it out to SA

Flash Drive Serial #	0019 E06B588B-FB6187B322BB

19. SA plugs in the Flash Drive. By default the Flash Drive will be auto-mounted and its contents available at **/MEDIA/KEY_GEN_LOG**.
20. SA elevate privileges to access the Flash Drive

```
user@laptop$ sudo bash
root@laptop#
```

TIME 8:55

21. SA verifies the FD serial number matches the serial number recorded on the script

```
lsusb -v -d 0x0951:0x1653 | grep -C 1 iProduct
iManufacturer 1 Kingston
iProduct 2 DT 100 G2
iSerial 3 0019E06B588BFB6187B322BB
```

TIME 8:56

22. SA starts logging via **script**

```
root@laptop# cd /media/KEY_GEN_LOG
root@laptop# script script-`date +"%Y%m%d"`.log
Script started, file is script-20100120.log
```

TIME 8:56

23. SA accesses the signing box via SSH using their own account, providing their own SSH identity

```
ssh -i catalyst-sysadmin-ssh-key sysadmin@sign1.internal.srs.net.nz
```

TIME 8:57

24. KGA checks the fingerprint for the server matches **b2:29:9f:b3:b9:b9:88:5b:4e:80:d6:c3:64:ff:ff:9b**

```
The authenticity of host 'sign1.internal.srs.net.nz (192.168.58.14)'
can't be established.
RSA key fingerprint is b2:29:9f:b3:b9:b9:88:5b:4e:80:d6:c3:64:ff:ff:9b
.
Are you sure you want to continue connecting (yes/no)? yes
```

TIME 8:58

25. SA enters the directory /var/lib/dnssec/keygen. Files generated during the key generation procedure will be stored here for later retrieval.

```
sysadmin@sign1: sudo -s
[sudo] password for sysadmin:
[/home/sysadmin]
root@sign1: cd /var/lib/dnssec/keygen
[/var/lib/dnssec/keygen]
root@sign1:
```

TIME 8:58

HSM Acceptance Test

Before putting an HSM into production, it should be tested and reset to factory default (zeroization):

HSM Diagnostics

Estimated time: < 8 min

For this procedure, interact with the HSM via the host command-line.

26. SA shows the installed devices

```
scadiag -l
mca/0
```

TIME 8:59

27. SA forces device into offline mode

```
scadiag -m offline mca0
Device mca0 is now offline
```

TIME 8:59

28. SA displays the device version numbers. Output will look like the example below.

```
scadiag -v mca0
Device mca0 version numbers:
Hardware : 1.5.50
Bootrom  : 1.0.10
Firmware : 1.1.2
```

TIME 9:00

29. KGA notes the version numbers

Hardware version #	1.4.50
Bootrom version #	1.0.10
Firmware version #	1.1.7

30. SA starts diagnostics

```
scadiag -d mca0
```

TIME 9:00

Diagnostics output should look like this:

```
Running mca0 on-board diagnostics.
Diagnostics on mca0 PASSED.
```

TIME 9:00

31. SA resets device

```
scadiag -r mca0
```

TIME 9:01

Reset output should look like this:

```
Resetting device mca0, this may take a minute.
Please be patient.
Device mca0 reset ok.
```

TIME 9:01

HSM Zeroize

Estimated time: < 5 min

32. SA zeroizes device

```
scadiag -z mca0
```

TIME 9:01

Output should like something like this (on console):

```
Zeroizing device mca0, this may take a few minutes.
Please be patient.
Device mca0 zeroized.
```

TIME 9:02

Once this is completed, the HSM is ready to be used in production

HSM Initialization

Connecting for the first time

Estimated time: 8 min

During this process the HSM will create a new public key used to connect securely to the device, in addition to an initial Device Security Officer. For this procedure, the NZRS DSO1 will be the initial DSO and they will be named **nz-dso1**.

⚠ **REMEMBER:** It's not possible to retrieve a forgotten password
Password must comply with the following:

- Minimum 8 characters
- At least three characters must be alphabetic
- At least one must be nonalphabetic.
- At least one Uppercase and one lowercase character

33. SA initializes the board. Output will look the following example, Serial ID and Key Fingerprint will differ.

```
root@sign1: scamgr -D
Warning: Serial ID and Public Key Not Found
-------------------------------------------------------------
The Serial ID and public key presented by this board were
not found in your trust database.

Serial ID: 36:30:35:34:30:33
Key Fingerprint: 630b-ec3b-450f-78bc-57db-9a92-3ba8-520c-5c12-6f84
-------------------------------------------------------------
Please select an action:

1. Abort this connection
2. Trust the board for this session only.
3. Replace the trusted key with the new key.

Your Choice --> 2

This board is uninitialized.
You will now initialize the board. You may either
initialize the board with a new configuration or
restore the configuration from a device backup file.

1. Initialize board with new configuration
2. Initialize board from device backup file
Your Choice (0 to exit) --> 1
Run in FIPS 140-2 mode? (Y/Yes/N/No) [No]: Y
```

TIME

9:05

34. DSO1 inputs their credentials

```
Initial Security Officer Name: nz-dso1
Initial Security Officer Password:
Confirm password:
```

TIME

9:06

35. SA confirms initialization

```
Board initialization parameters:
-----------------------------------------------------------------
Initial Security Officer Name: nz-dso1
Run in FIPS 140-2 Mode: Yes
-----------------------------------------------------------------

Is this correct? (Y/Yes/N/No) [No]: Y
Initializing crypto accelerator board. This may take a few minutes...
The board is ready to be administered.
As part of the initialization process, a new remote access key has
been
generated. The key fingerprint is listed below. This should be the
fingerprint presented by the board the next time you connect to it.
Key Fingerprint: 7b48-0854-dce0-253a-a3a1-9a2d-7070-f7fe-787e-14f8
```

TIME 9:07

36. KGA records the fingerprint provided by the HSM to be verified during the next key generation procedure

Serial ID 36 : 30 : 30 : 31 : 32 : 31

Key Fingerprint 4fbd - 91b6 - f9e8 - 56a2 - bc42 - ad7d - 321c - 9846 - f47f - 2936

Disconnect, Reconnect and set trusted key fingerprint

Estimated time: 3 min

37. SA disconnects from the HSM, cancelling the current connection

```
Security Officer Login: Control-C
```

TIME 9:07

38. SA reconnects to the board.
39. KGA validates fingerprint and serial number.
40. SA sets to trust the fingerprint if fingerprint and serial number match (option 3)

```
root@sign1: scamgr -D
Warning: Serial ID and Public Key Not Found
-----------------------------------------------------------------
The Serial ID and public key presented by this board were
not found in your trust database.

Serial ID: 36:30:35:34:30:33
Key Fingerprint: c478-bd1b-2b18-30ae-2946-607d-eaff-5bc4-ba2f-9aa3
-----------------------------------------------------------------
Please select an action:

1. Abort this connection
2. Trust the board for this session only.
3. Trust the board for all future sessions.

Your Choice --> 3
```

TIME 9:09

41. **DSO1** authenticates.

```
Security Officer Login: nz-dso1
Security Officer Password:
```

TIME 9:10

Set the password requirements

Estimated time: 1 min

42. SA sets the password requirements for the device

```
scamgr{mca0@localhost, nz-dso1}> set passreq high
New password security level: HIGH
```

TIME 9:10

Create the remaining DSO roles

Estimated time: 3 min

43. SA creates DSO2 (**nz-dso2)**,
44. DSO2 inputs their credential

```
scamgr{mca0@localhost, nz-dso1}> create so nz-dso2
Enter new security officer password:
Confirm password:
Security Officer nz-dso2 created successfully.
```

TIME
9:11

45. SA creates DSO3 (**nz-dso3)**,
46. DSO3 inputs their credential

```
scamgr{mca0@localhost, nz-dso1}> create so nz-dso3
Enter new security officer password:
Confirm password:
Security Officer nz-dso3 created successfully.
```

TIME
9:12

47. SA creates DSO4 (**nz-dso4)**,
48. DSO4 inputs their credential

```
scamgr{mca0@localhost, nz-dso1}> create so nz-dso4
Enter new security officer password:
Confirm password:
Security Officer nz-dso4 created successfully.
```

TIME
9:13

49. SA creates DSO5 (**nz-dso5)**,
50. DSO5 inputs their credential

```
scamgr{mca0@localhost, nz-dso1}> create so nz-dso5
Enter new security officer password:
Confirm password:
Security Officer nz-dso5 created successfully.
```

TIME
9:13

51. SA checks the DSOs are created (order may vary)

```
scamgr{mca0@localhost, nz-dso1}> show so
Security Officer Multi-Admin Role
-----------------------------------------------------------------
nz-dso2 Disabled
nz-dso3 Disabled
nz-dso1 Disabled
nz-dso4 Disabled
nz-dso5 Disabled
-----------------------------------------------------------------
```

TIME
9:13

52. SA logs out as DSO1

```
scamgr{mca0@localhost, nz-dso1}> quit
```

TIME
9:14

Keystore creation and initialization

Keystore creation

Estimated time: 5 min

During the creation of the keystore, the first KSO has to be created as well.
The keystore will be named **nz-dnssec-keystore**, created as a **Local Keystore**, running in "FIPS 140-2 mode" and the Keystore Security Officers named **nz-kso<N>** where *<N>* is a digit between 1 and 5.

53. SA executes HSM interface and sets the keystore parameters

```
root@sign1: scamgr
No keystore data returned by card

Select Keystore:
1. Create new keystore
2. Load keystore from backup

Selection (0 to exit)-> 1
FIPS Keystore Name: nz-dnssec-keystore
Keystore type ([L]ocal/[C]entralized) [Local]: L
```

TIME

9:15

54. KSO1 inputs their password.

```
Initial Security Officer Name: nz-kso1
Initial Security Officer Password:
Confirm password:
```

TIME

9:16

55. SA confirms the creation of the keystore

```
Keystore creation parameters:
----------------------------------------------------------------
Keystore Name: nz-dnssec-keystore
Keystore Type: Local
Initial Security Officer Name: nz-kso1
Run in FIPS 140-2 Mode: Yes
----------------------------------------------------------------

Is this correct? (Y/Yes/N/No) [No]: Y
Creating keystore...
<This step takes some time>
nz-dnssec-keystore.600121.{b129f5fa} successfully created.
```

TIME

9:17

Keystore initialization

Estimated time: 15 min

56. KSO1 logs in as the **nz-kso1** created in the previous step

```
Security Officer Login: nz-kso1
Security Officer Password:
scamgr{mca0@localhost, nz-kso1}>
```

TIME

9:17

57. SA changes the password setting to high

```
scamgr{mca0@localhost, nz-kso1}> set passreq high
New password security level: HIGH
```

TIME

9:18

58. SA sets the auditing level to 6, in order to record any access to the keystore objects.

```
scamgr{mca0@localhost, nz-kso1}> set audit-level 6
Audit level = 6 (Token)
```

TIME

9:18

59. SA creates the remaining Security Officers. This step requires each KSO to enter their credentials.
    60. SA creates Keystore Security Officer 2. KSO2 types their own password.

```
scamgr{mca0@localhost, nz-kso1}> create so nz-kso2
Enter new security officer password:
Confirm password:
Security Officer nz-kso2 created successfully.
```

TIME

9:18

61. SA creates Keystore Security Officer 3. KSO3 types their own password.

```
scamgr{mca0@localhost, nz-kso1}> create so nz-kso3
Enter new security officer password:
Confirm password:
Security Officer nz-kso3 created successfully.
```

TIME

9:19

62. SA creates Keystore Security Officer 4. KSO4 types their own password.

```
scamgr{mca0@localhost, nz-kso1}> create so nz-kso4
Enter new security officer password:
Confirm password:
Security Officer nz-kso4 created successfully.
```

TIME

9:20

63. SA creates Keystore Security Officer 5. KSO5 types their own password.

```
scamgr{mca0@localhost, nz-kso1}> create so nz-kso5
Enter new security officer password:
Confirm password:
Security Officer nz-kso5 created successfully.
```

TIME

9:21

64. SA creates Keystore Security Officer **nz-kso-ops** for maintenance tasks. Use a pre-generated password for this account.

```
scamgr{mca0@localhost, nz-kso1}> create so nz-kso-ops
Enter new security officer password:
Confirm password:
Security Officer nz-kso-ops created successfully.
```

TIME

9:22

65. SA checks the list of Security Officers is complete

```
scamgr{mca0@localhost, nz-dso1}> show so
Security Officer Multi-Admin Role
-----------------------------------------------------------------
nz-kso1 Disabled
nz-kso2 Disabled
nz-kso3 Disabled
nz-kso4 Disabled
nz-kso5 Disabled
nz-kso-ops Disabled
-----------------------------------------------------------------
```

TIME

9:22

66. SA enables all the Keystore Security Officers but **nz-kso-ops** as authorized members of Multi-Admin mode

```
scamgr{mca0@localhost, nz-kso1}> enable authmember nz-kso1
Added multi-admin role to Security Officer nz-kso1.

scamgr{mca0@localhost, nz-kso1}> enable authmember nz-kso2
Added multi-admin role to Security Officer nz-kso2.

scamgr{mca0@localhost, nz-kso1}> enable authmember nz-kso3
Added multi-admin role to Security Officer nz-kso3.

scamgr{mca0@localhost, nz-kso1}> enable authmember nz-kso4
Added multi-admin role to Security Officer nz-kso4.

scamgr{mca0@localhost, nz-kso1}> enable authmember nz-kso5
Added multi-admin role to Security Officer nz-kso5.
```

TIME

9:23

67. SA checks the list of authorized Multi-Admin Security Officers is complete

```
scamgr{mca0@localhost, nz-kso1}> show so
Security Officer Multi-Admin Role
----------------------------------------------------------------
nz-kso5 Enabled
nz-kso3 Enabled
nz-kso-ops Disabled
nz-kso1 Enabled
nz-kso2 Enabled
nz-kso4 Enabled
----------------------------------------------------------------
```

TIME 9:23

68. SA creates a user for the keystore. This credential will be used by the signing engine to interact with the HSM, Use a pre-generated password for this account.

```
scamgr{mca0@localhost, nz-kso1}> create user nz-dnssec-user
Enter new user password:
Confirm password:
User nz-dnssec-user created successfully.
```

TIME 9:24

69. SA sets the minimum number of KSO needed to authorize a command

```
scamgr{mca0@localhost, nz-kso1}> set multiadmin minauth 2
Multi-admin mode now requires 2 security officers to authenticate.
```

TIME 9:24

70. SA sets the maximum time to wait for the KSO credentials

```
scamgr{mca0@localhost, nz-kso1}> set multiadmin timeout 5
New multi-admin timeout value is 5 minutes.
```

TIME 9:25

71. SA activates the Multi-Admin mode for the keystore

```
scamgr{mca0@localhost, nz-kso1}> enable multiadmin
WARNING: This command will place the device in multi-
admin mode. This mode will require multiple
security officers to authenticate for certain
commands to be executed.

Enable Multi-Admin Mode? (Y/Yes/N/No) [No]: Y

Multi-Admin mode parameters:
----------------------------------------------------------------
Minimum number of security officers: 2
Multi-Admin command timeout: 5 minutes
----------------------------------------------------------------

Is this correct? (Y/Yes/N/No) [No]: Y
The board is now in multi-admin mode.
```

TIME 9:25

72. SA disconnects from the board

```
scamgr{mca0@localhost, nz-kso1}> exit
```

TIME 9:25

Key generation

Estimated time: 15 min

Create all the necessary keys for fourteen months of operation (one year plus two months extra for overlap).

73. SA starts the pkcsslotd daemon

```
/etc/init.d/pkcsslotd start
Starting pkcsslotd: [ OK ]
```

TIME 9:26

74. SA set the TokenLabel and PIN for the HSM in OpenDNSSEC configuration (using the opendnssec user)

```
sudo -u opendnssec update-config-password.pl sca6000
This program will take a username and password from the user and
update the OpenDNSSEC config such that the HSM can be accessed.
The password must:
- be at least 12 characters long
- contain at least three letters
- at least one letter must be capital
- at least one letter must be lower-case
- contain at least one digit
- contain at least one non-alphanumeric character
Username: nz-dnssec-user
Password: ************
Password (again): ************
New configuration file passes OpenDNSSEC validation checks.
Verified access to HSM
```

TIME 9:27

75. SA lists the contents of the HSM. It must contain no keys.

```
ods-hsmutil list sca6000
```

TIME 9:27

76. SA execute the script to generate the keys for all active policies

```
sudo -u opendnssec ods-keygen.sh P14M
```

TIME 9:28

> The key generation script will run a sanity check on the list of keys previous and after the generation step, to make sure only new keys are added and no existing keys are deleted

77. SA prints the number of keys present in the HSM. Output would look as below:

```
ods-hsmutil list sca6000 | head -5
Listing keys in repository: sca6000
140 keys found.

Repository ID Type
---------- -- ----
sca6000 160d29b6d32b301356a22f545e1a5ddd RSA/2048
sca6000 33b6e77e122419a7e6893d2c5e2bcffb RSA/2048
sca6000 9d893962239be58bfcdb3fd45a6454a5 RSA/2048
sca6000 5ac0c4de0626543295d37bc850200f86 RSA/2048
sca6000 76394a2af741e324ad49646b4b59dd53 RSA/2048
```

TIME 9:30

Backup generation

Estimated time: 10 min

78. SA opens a second terminal and logs into the signing box using their own account.

```
ssh -i catalyst-sysadmin-ssh-key sysadmin@sign1.internal.srs.net.nz
```

TIME 9:31

79. SA executes backup script. The backup files will be written to /var/lib/dnssec/keygen/key-backup-YYYY-MM-DD.tar.gz

```
export-keydata nz-dnssec-keystore
Backups will be written to
/var/lib/dnssec/keygen/key-backup-YYYY-MM-DD.tar.gz
Exporting KASP database...
SQLite database set to: /var/opendnssec/kasp.db

Backing up keystore nz-dnssec-keystore...

You will be prompted for Keystore Security Officer(KSO) credentials.
After entering them, the backup will pause while other Keystore
Security Officers authorize the backup operation.

Press enter to continue.
```

TIME 9:32

80. KSO1 authorizes the backup using their password

```
Keystore = nz-dnssec-keystore.600121.{b129f5fa} (local)
Security Officer Login: nz-kso1
Security Officer Password:
NOTICE: Please wait while the other required 1 security officers
authenticate this command. This command will time out
in 5 minutes.
```

TIME 9:33

81. SA executes the HSM interface in the second window

```
scamgr -k nz-dnssec-keystore
Keystore = nz-dnssec-keystore.600121.{b129f5fa} (local)
```

TIME 9:34

82. A second KSO logs into the HSM using the second terminal to authorize the backup.

```
Security Officer Login: nz-kso2
Security Officer Password:
NOTICE: A Multi-Admin command is currently in progress.
You are a member of the Multi-Admin role and
may approve this command.
Command: backup
Initiating SO: nz-kso1

Authorize this command? (Y/Yes/N/No) [No]: Y
Authorization successful
```

TIME 9:34

> Any KSO pair combination can carry out this operation, using nz-kso1, and nz-kso2 is only relevant for the example

83. SA closes the second HSM interface and window

```
scamgr> quit
```

TIME 9:35

84. The first terminal will show the backup command was authorized and will proceed. Output will look like the following example:

```
Update: Authenticated security officers: nz-kso1
Update: Authenticated security officers: nz-kso1 nz-kso2
Backup to
/tmp/tmp.cgHkVs1862/nz-dnssec-keystore-full-keystore-backup-YYYY-MM-DD
successful.

Done backing up keystore nz-dnssec-keystore. The sha256sum of this
full keystore backup is 8b:42:9f:fb:d6:40:7b:52:90:b4:94:18:49:48:
4b:a6:55:11:42:70:b8:0f:51:8b:62:50:37:e8:14:1e:71:b9

Backing up HSM Device Configuration...
You will be prompted for Device Security Officer(DSO) credentials and
a Password to encrypt to the device backup.

Press enter to continue.
```

TIME 9:35

85. DSO1 authorizes the device backup with their password

```
Security Officer Login: nz-dso1
Security Officer Password:
```

TIME 9:36

86. SA enters the password to protect the backup, using a pre-generated password. Output should look as below:

```
Enter a password to protect the data:
Confirm password:
Backup to /tmp/tmp.cgHkVs1862/device-backup-YYYY-MM-DD successful.

Done backing up HSM device. The sha256sum of this device backup is
a4:cd:83:45:02:51:7c:3b:38:5d:88:8d:22:2a:47:8f:67:7c:60:47:2d:ea:
56:17:1b:b8:6c:95:e0:bc:d0:32

Exported keystore Info:
Keystore : nz-dnssec-keystore
Serial # : 605403
Keystore ID : 519920a1
All backups have been exported to
/var/lib/dnssec/keygen/key-backup-YYYY-MM-DD.tar.gz
Hash of key-backup-YYYY-MM-DD.tar.gz has been written to
key-backup-YYYY-MM-DD.tar.gz.sha256sum (sha256sum:
66:2c:1d:ad:32:7c:00:e4:25:96:cb:fb:c4:6e:9d:b6
:e9:be:1d:fb:ad:46:d1:e7:85:eb:eb:23:2c:48:78:eb )
```

TIME
9:38

87. SA reads the digest from the screen, KGA records on its script copy

Keystore backup file digest	48:cd:79:72:5b:99:92:e1:62:8b:b2:c1:2a:ed:a5:a4: 84:d3:0b:df:2e:56:39:41:7e:85:4f:4d:80:cd:8f:57:

88. SA closes the root session

```
root@sign1: exit
```

TIME
9:40

89. SA logs outs from the signing box

```
sysadmin@sign1: exit
Connection to sign1.internal.srs.net.nz closed.
```

TIME
9:40

Creating Master Backup Copy

Estimated time: 5 min

90. KGA takes the Flash Drive labeled as **Master Copy** to serve as Master Copy Container. KGA will record the serial number on its script copy.

Flash Drive Serial #	0019e06b5884 - fb61874a20ab

91. KGA passes the Flash Drive to SA
92. SA plugs Flash Drive into the laptop
93. SA verifies the FD serial number matches the serial number recorded on the script

```
lsusb -v -d 0x0951:0x1653 | grep -C 1 iProduct
iManufacturer 1 Kingston
iProduct 2 DT 100 G2
iSerial 3 0019E06B5884FB61874A20AB *
--
iManufacturer 1 Kingston
iProduct 2 DT 100 G2
iSerial 3 0019E06B588BFB6187B322BB
```

TIME
9:43

94. SA copies the backup files from the signer to the Flash Drive

```
scp -i catalyst-sysadmin-ssh-key
admin@sign1:/var/lib/dnssec/keygen/key-backup-* /media/MASTER_BACKUP/
Enter passphrase for key 'catalyst-sysadmin-ssh-key':
key-backup-YYYY-MM-DD.tar.gz 100% 453KB
key-backup-YYYY-MM-DD.tar.gz.sha256sum 100% 95
```

TIME
9:44

95. SA checks the backup file integrity

```
cd /media/MASTER_BACKUP
sha256sum -c key-backup-YYYY-MM-DD.tar.gz.sha256sum
key-backup-YYYY-MM-DD.tar.gz: OK
```

TIME 9:46

Creating Backup Operative Copies

Wellington Operative Backup Copy

Estimated time: 5 min

96. KGA picks Flash Drive labeled **WELLINGTON**, and records the serial number in its script copy.

Flash Drive Serial # 001478544884 - fb618742204A

97. KGA hands out the FD to the SA
98. SA plugs the FD into the laptop
99. SA verifies the FD serial number matches the serial number recorded on the script. This command will show three serial numbers, one for the KeyGen-Log Flash Drive, one for the Master Backup and one for the Wellington Flash Drive.

```
lsusb -v -d 0x0951:0x1653 | grep -C 1 iProduct
iManufacturer 1 Kingston
iProduct  2 DT 100 G2
iSerial  3 0019E06B5884FB61874A20AB
-
iManufacturer 1 Kingston
iProduct  2 DT 100 G2
iSerial  3 001478544884FB618742204A
-
iManufacturer 1 Kingston
iProduct  2 DT 100 G2
iSerial  3 0019E06B588BFB6187B322BB
```

TIME 9:47

100. SA copies the MBC FD contents into the Wellington OBC FD

```
rsync -avW /media/MASTER_BACKUP/ /media/WELLINGTON/
```

TIME 9:48

101. SA checks the integrity of the backup

```
cd /media/WELLINGTON
sha256sum -c key-backup-YYYY-MM-DD.tar.gz.sha256sum
key-backup-YYYY-MM-DD.tar.gz: OK
```

TIME 9:48

102. SA unmounts and unplugs the OBC FD

```
cd /
umount /media/WELLINGTON
```

TIME

103. SA hands out the FD to the KGA
104. KGA labels a TEB as **WELLINGTON, <DATE>, NZRS DNSSEC Key Backup**
105. KGA records the TEB serial number in its script copy

TEB Serial # 3187081

106. KGA places the WELLINGTON OBC FD in the TEB
107. KGA places copy of the Device Backup Password, KSO Ops Password and nz-dnssec-user Password in the TEB
108. KGA seals the TEB
109. KGA hands out the TEB to Catalyst Representative
110. Catalyst Representative confirms the TEB serial matches the script log and signs in acknowledgement

Catalyst Representative signature

Albany Operative Backup Copy

Estimated time: 5 min

111. KGA picks the Flash Drive labeled **ALBANY**, and records the serial number in its script copy.

Flash Driver Serial #	0019e06b587b-fb6187432154

112. KGA hands out the FD to the SA
113. SA plugs the FD into the laptop
114. SA verifies the FD serial number matches the serial number recorded on the script

```
lsusb -v -d 0x0951:0x1653 | grep -C 1 iProduct
iManufacturer 1 Kingston
iProduct 2 DT 100 G2
iSerial 3 0019E06B5884FB61874A20AB
-
iManufacturer 1 Kingston
iProduct 2 DT 100 G2
iSerial 3 0019E06B587BFB6187432154
-
iManufacturer 1 Kingston
iProduct 2 DT 100 G2
iSerial 3 0019E06B588BFB6187B322BB
```

TIME 9:52

115. SA copies the MCB FD contents into the Albany OBC FD

```
rsync -avW /media/MASTER_BACKUP/ /media/ALBANY/
```

TIME 9:53

116. SA checks the integrity of the backup

```
cd /media/ALBANY
sha256sum -c key-backup-YYYY-MM-DD.tar.gz.sha256sum
key-backup-YYYY-MM-DD.tar.gz: OK
```

TIME 9:53

117. SA unmounts and unplugs the OBC FD

```
cd /
umount /media/ALBANY
```

TIME 9:54

118. SA hands out the FD to the KGA
119. KGA labels a TEB as **ALBANY, <DATE>, NZRS DNSSEC Key Backup**
120. KGA records the TEB serial number in its script copy

TEB Serial #	3187083

121. KGA places the ALBANY OBC FD in the TEB
122. KGA places copy of the Device Backup Password, KSO Ops Password and nz-dnssec-user Password in the TEB
123. KGA seals the TEB
124. KGA hands out the TEB to Knossos Representative
125. Knossos Representative confirms the TEB serial matches the script log and signs in acknowledgement

Knossos Representative signature	John R Rumsey

Auckland Operative Backup Copy

Estimated time: 5 min

126. KGA picks Flash Drive labeled **AUCKLAND**, and records the serial number in its script copy

Flash Drive Serial #	0019e06b0842-fb6187ae20fc

127. KGA hands out the FD to the SA
128. SA plugs the FD into the laptop
129. SA verifies the FD serial number matches the serial number recorded on the script

```
lsusb -v -d 0x0951:0x1653 | grep -C 1 iProduct
iManufacturer 1 Kingston
iProduct 2 DT 100 G2
iSerial 3 0019E06B5884FB61874A20AB
-
iManufacturer 1 Kingston
iProduct 2 DT 100 G2
iSerial 3 0019E06B0842FB6187AE20FC
-
iManufacturer 1 Kingston
iProduct 2 DT 100 G2
iSerial 3 0019E06B588BFB6187B322BB
```

TIME 9:57

130. SA copies the MCB FD contents into the AUCKLAND OBC FD

```
rsync -avW /media/MASTER_BACKUP/ /media/AUCKLAND
```

TIME 9:58

131. SA checks the integrity of the backup

```
cd /media/AUCKLAND
sha256sum -c key-backup-YYYY-MM-DD.tar.gz.sha256sum
key-backup-YYYY-MM-DD.tar.gz: OK
```

TIME 9:58

132. SA unmounts and unplugs the OBC FD

```
cd /
umount /media/AUCKLAND
```

TIME 10:00

133. SA hands out the FD to the KGA
134. KGA labels a TEB as **AUCKLAND, <DATE>, NZRS DNSSEC Key Backup**
135. KGA records the TEB serial number in its script copy

TEB Serial # 3187085

136. KGA places the AUCKLAND OBC FD in the TEB
137. KGA places copy of the Device Backup Password, KSO Ops Password and nz-dnssec-user Password in the TEB
138. KGA seals the TEB
139. KGA hands out TEB to Richard Currey
140. Richard Currey confirms the TEB serial matches the script log and signs in acknowledgement

Richard Currey signature

Finishing steps

Estimated time: 3 min

141. SA unmounts and unplugs the MBC FD

```
cd /
umount /media/MASTER_BACKUP
```

TIME 10:00

142. SA hands out the MBC FD to the KGA
143. KGA labels a TEB as **Master Copy, <DATE>, NZRS DNSSEC Key Backup**
144. KGA records the TEB serial number in its script copy

TEB Serial # 3187087

145. KGA places the MBC FD in the TEB
146. KGA places copy of the Device Backup Password, KSO Ops Password and nz-dnssec-user Password in the TEB
147. KGA seals the TEB
148. KGA hands out TEB to KSO1
149. KSO1 confirms the TEB serial matches the script log and signs in acknowledgement

KSO1 signature

DAVE BAKER.

Closing steps

Estimated time: 12 min

150. SA finishes script logging

```
root@laptop> exit
```

TIME 10:02

151. KGA selects Flash Drive labeled **Key Gen Copy** and hands it out to SA
152. SA plugs in the Flash Drive
153. SA copies **Key Gen Log** Flash Drive contents into **Key Gen Copy** Flash Drive

```
rsync -avW /media/KEY_GEN_LOG/ /media/KEYGEN_COPY
```

TIME

154. SA generates a printable copy of the script

```
cd /media/KEYGEN_COPY
enscript -G -U 2 -o script-`date +"%Y%m%d"`.ps script-`date
+"%Y%m%d"`.log
```

TIME

155. SA generates sha256 digest for the printable copy of the script. Output should look like this:

```
openssl dgst -c -sha256 script-`date +"%Y%m%d"`.ps
SHA256(script-YYYYMMDD.ps)=
a6:83:6e:17:cb:37:ed:f2:06:41:b0:47:25:d3:1b:e4
:8f:11:a5:56:38:bd:b2:a5:ec:dc:17:45:fb:9a:6d:94
```

TIME

156. KGA records the sha256 digest into the script copy

sha256 digest	cc:fe:0c:f9:c6:e3:b9:78:70:94:07:42:0d:61:5d:4d: 91:5f:0e:08:ea:ea:3f:ab:92:3a:9d:28:f3:d8:30:d7:

157. SA prints the script

```
lpr script-`date +"%Y%m%d"`.ps
```

TIME 10:11

158. SA copies the printable copy to the **Key Gen Log** Flash Drive

```
cp /media/KEYGEN_COPY/script-`date +"%Y%m%d"`.log.ps
/media/KEY_GEN_LOG
```

TIME 10:13

159. SA unmounts KEY_GEN_LOG FD

```
cd /
umount /media/KEY_GEN_LOG
```

TIME 10:13

160. SA unplugs Flash Drive and hands it out to KGA
161. KGA takes a TEB and records the serial number in its script copy

TEB Serial #	3187089

162. KGA places KeyGen_Log FD in the TEB and seals it
163. SA unmounts KEYGEN_COPY FD and hands it out to KGA

```
cd /
umount /media/KEYGEN_COPY
```

TIME 10:16

164. SA unmounts and unplugs the Flash Drive carrying his key
165. SA shuts down laptop

NO.3187089

`shutdown -h now`	TIME

166. SA disconnects cables from laptop
167. Unplug laptop cables
168. KSO1 takes TEB containing Key Generation Log FD, TEB containing Master Backup Copy and copies of the script log for secure storage
169. KGA signs off the key generation procedure

Signature	
Date/Time	18-Nov-2011, 10:16

170. KGA makes at least 3 photocopies of its copy of the script: one for onsite storage, offsite storage, one for KGA. Additional copies can be made by participants request.

Key Generation Event Record

Event #	1
Date/Time	18-Nov-2011

Description

Signer runs on UTC, time/date for key backup file may be wrong.

KGA signature	

Key Generation Event Record

Event #	2
Date/Time	18-Nov-2011, 10:12

Description

Step 158, file is called log.ps but output file was named .ps

KGA signature	

Key Generation Event Record

Event #	3
Date/Time	18-Nov-2011, 10:15

Description

No labelling step for the Key Gen Log bag.

KGA signature	