

Fourth Key Generation

Version:	30
Last modification:	Feb 16, 2015 11:28

Estimated time: 1 hour and 45 minutes

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.






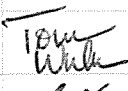


Abbreviations

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 labelled and formatted	6
Spare formatted Flash Drives	2
Tamper-Evident Bags	6
Pre-generated secure password set for device backup	2
Sysadmin brings ssh key to access the signer	1
Hard copies of this script	8
Copy of previous Key Generation Procedure script	1
Copy of previous HSM restoration from Backup script	1
Participant sign-in sheet	1

Participants

Role	Organization	Printed Name	Signature	Date	Time
KGA/DSO1	NZRS	Dane Foster		16/2/15	1:06
SA/DSO2	NZRS	Josh Simpson		Feb 14/2015	1:09pm
KSO1	NZRS	Dave Baker		Feb 16 2015	1:04pm
KSO2	NZRS	Jay Daley		FEB 16 2015	2:09 pm.
KSO3	NZRS	Brenda Wallace		16/2/15	1331 - left 14:10
DSO3	NZRS	Mike Forbes			
DSO4	OSS	Tom Weber		Feb 16, 15	1:03pm
DSO5	NZRS	Daniel Griggs		16/2/2015	1:04pm
KSO5	NZRS	Sebastian Castro		16/2/2015	1:04pm

Safety Instructions

Estimated time: 5 min

KGA 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: 5 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. KGA retrieves:
 5. Laptop (includes power cable, video cable, power extension)
 6. Printer (includes power + usb cable, and paper)
 7. CD,
 8. Flash Drives
 9. Tamper-Evident Bags
 10. Cello tape

Laptop setup

Estimated time: 15 min

- 11. SA sets up the laptop for the key generation procedure
 - 12. Connects power cable, network cable, and projector
 - 13. Powers up laptop, hit ENTER to access boot menu
 - 14. Boot-up laptop using a bootable CD
 - 15. Enables display
 - 16. Configures printer and print test page
 - 17. Open terminal, and maximize for visibility

- 18. SA verifies the integrity of the Live CD by comparing the digest

```
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
13:15

Matches record? YES / NO

- 19. SA verifies time and date on the laptop

```
root@laptop# date
```

TIME
13:25

- 20. KGA records date and time on their script copy

Date: Mon Feb 16 2015
Time: 13:25:35

- 21. KGA select Flash Drive labeled Utils, records the serial number on their script copy and hands it out to SA
Flash Drive Serial #

- 22. SA plugs in the Flash Drive, By default the Flash Drive will be auto-mounted and its contents available at /media/UTIL

- 23. 070B516D1B828837
SA elevates privileges to access the Flash Drive

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

TIME
13:28

- 24. SA verifies the FD serial number matches the serial number recorded in the script

```
lsusb -v -d 13fe:4200 | grep -C 1 iProduct
iManufacturer 1
iProduct 2 USB DISK 2.0
iSerial 3 070B516D1B828837
```

TIME
13:28

- 25. SA copies SSH key and config for access to signers to the laptop

```
cp /media/UTIL/SA_KEY/id_rsa /root/.ssh/id_rsa
cp /media/UTIL/SA_KEY/config /root/.ssh/config
chmod 0600 /root/.ssh/sa_key
```

TIME
13:32

- 26. SA unmount and ejects Util FD

```
eject /media/UTIL
```

TIME
13:32

Access to the signing box

Estimated time: 5 min

- 27. 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 # 070B516E3BB4CE31

28. SA plugs in the Flash Drive. By default the Flash Drive will be auto-mounted and its contents available at `/media/KEY_GEN_LOG`.

29.

SA verifies the FD serial number matches the serial number recorded in the script

<pre>lsusb -v -d 13fe:4200 grep -C 1 iProduct iManufacturer 1 iProduct 2 USB DISK 2.0 iSerial 3 070B516E3BB4CE31</pre>	TIME 13:34
--	---------------

30.

SA starts logging via script

<pre>root@laptop# cd /media/KEY_GEN_LOG root@laptop# script script-\$(date +%Y%m%d)-1.log Script started, file is script-20131206.log</pre>	TIME 13:34
---	---------------

31.

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

<pre>ssh sysadmin@sign1.internal.srs.net.nz</pre>	TIME 13:34
---	---------------

32.

KGA checks the fingerprint for the server matches the records

sign1 fingerprint	<u>b2:29:9f:b3:b9:b9:88:5b:4e:80:d6:c3:64:ff:ff:9b</u> ✓
sign2 fingerprint	<u>ed:73:ee:03:6c:4c:c0:26:3a:e8:f4:cc:60:26:a1:81</u>
srsplug1 fingerprint	<u>ae:b0:a4:17:0c:8b:82:30:1c:bb:73:11:4a:4f:1e:84</u> ✓
srslog1 fingerprint	<u>a9:4c:d8:20:a9:66:ef:7c:0a:9d:60:f3:77:16:4c:b9</u>

<pre>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</pre>	TIME 13:40
--	---------------

Matches record?

YES / NO

33.

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

<pre>sysadmin@sign1: sudo -s [sudo] password for sysadmin: [/home/sysadmin] root@sign1: cd /var/lib/dnssec/keygen [/var/lib/dnssec/keygen] root@sign1:</pre>	TIME 13:41
--	---------------

HSM Verification

Estimated time: 5 min

34.

SA retrieves the HSM public key fingerprint

<pre>root@sign1: scadiag -f mca0 4fbd-91b8-f9e8-56a2-bc42-ad7d-321c-9846-f47f-2936</pre>	TIME 13:41
--	---------------

35.

KGA verifies the HSM Fingerprint matches what's recorded in the previous script (step 28)

Matches record?

YES / NO

Roles clean-up and additions

Due to changes related to insourcing, some of the existing DSO and KSO roles need to be reassigned. An acceptable password requires eight characters minimum, three characters must be alphabetic, and one character must be non-alphabetic.

Replacing DSO roles

Estimated time: 5 min

36.
DSO5 access the board and authenticates themselves.

root@sign1: scamgr -D Security Officer Login: nz-dso5 Security Officer Password: scamgr{mca0@localhost, nz-dso5}>	TIME 13:49 13:50
--	----------------------------

You may see the following output:

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: 36:30:30:34:34:39 Key Fingerprint: d34d-ba64-ac50-eb28-b785-5c09-ebee-201f-db7c-13ef ----- Please select an action: 1. Abort this connection 2. Trust the board for this session only. 3. Trust the board for all future sessions.	TIME 13:49
--	-------------------

If this is the case, verify the serial number once again and enter 3.

37.
DSO5 deletes existing account DSO1

scamgr{mca0@localhost, nz-dso5}> delete so nz-dso1 Delete security officer nz-dso1? (Y/Yes/N/No) [No]: y Security Officer nz-dso1 deleted.	TIME 13:50
--	-------------------

38.
DSO5 deletes existing account DSO2

scamgr{mca0@localhost, nz-dso5}> delete so nz-dso2 Delete security officer nz-dso2? (Y/Yes/N/No) [No]: y Security Officer nz-dso2 deleted.	TIME 13:50
--	-------------------

39.
DSO5 deletes existing account DSO3

scamgr{mca0@localhost, nz-dso5}> delete so nz-dso3 Delete security officer nz-dso3? (Y/Yes/N/No) [No]: y Security Officer nz-dso3 deleted.	TIME 13:51
--	-------------------

40.
DSO1 creates its own account (**nz-dso1**)

scamgr{mca0@localhost, nz-dso5}> create so nz-dso1 Enter new security officer password: Confirm password: Security Officer nz-dso1 created successfully.	TIME 13:52
--	-------------------

41.
DSO2 creates its own account (**nz-dso2**)

<pre>scamgr{mca0@localhost, nz-dso5}> create so nz-dso2 Enter new security officer password: Confirm password: Security Officer nz-dso2 created successfully.</pre>	TIME 13:53
--	---------------

42.
DSO3 creates its own account (nz-dso3)

<pre>scamgr{mca0@localhost, nz-dso5}> create so nz-dso3 Enter new security officer password: Confirm password: Security Officer nz-dso3 created successfully.</pre>	TIME
--	------

43.
DSO4 creates its own account (nz-dso4)

<pre>scamgr{mca0@localhost, nz-dso5}> create so nz-dso4 Enter new security officer password: Confirm password: Security Officer nz-dso4 created successfully.</pre>	TIME 13:56
--	---------------

44.
DSO3 logs out current session and logs in back

<pre>scamgr{mca0@localhost, nz-dso5}> quit root@sign1: scamgr -D Security Officer Login: nz-dso3 Security Officer Password:</pre>	TIME 13:56
--	---------------

45.
DSO3 deletes existing DSO5 account

<pre>scamgr{mca0@localhost, nz-dso3}> delete so nz-dso5 Delete security officer nz-dso5? (Y/Yes/N/No) [No]: y Security Officer nz-dso5 deleted.</pre>	TIME 13:57
--	---------------

46.
DSO5 creates its own account

<pre>scamgr{mca0@localhost, nz-dso3}> create so nz-dso5 Enter new security officer password: Confirm password: Security Officer nz-dso5 created successfully.</pre>	TIME 13:58
--	---------------

47.
DSO5 checks all expected DSOs accounts are created (order may vary)

<pre>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 -----</pre>	TIME 13:58
---	---------------

48.
DSO5 logs out from the session

<pre>scamgr{mca0@localhost, nz-dso3}> quit</pre>	TIME 13:58
---	---------------

Replace KSO roles

49.
KSO1 logs in as **nz-kso1**

<pre>root@sign1: scamgr -k nz-dnssec-keystore Keystore = nz-dnssec-keystore.DDDDDD.{xxxxxxxx} (local) Security Officer Login: nz-kso1 Security Officer Password: scamgr{mca0@localhost, nz-kso1}></pre>	<p>TIME</p> <p>14:02</p>
--	--------------------------

50.
KSO1 disables multiadmin mode

<pre>scamgr{mca0@localhost, nz-kso1}> disable multiadmin WARNING: Issuing this command will take the board out of multi-admin mode and return it to the single-administrator mode of authentication. Proceed with change? (Y/Yes/N/No) [No]: y NOTICE: Please wait while the other required 1 security officer authenticates this command. This command will time out in 5 minutes. Update: Authenticated security officers: nz-kso1</pre>	<p>TIME</p> <p>14:03</p>
---	--------------------------

51.
SA opens a second terminal and logs into the signer

<pre>root@laptop# ssh sysadmin@sign1.internal.srs.net.nz root@sign1: sudo bash root@sign1: cd /media/KEY_GEN_LOG root@sign1: script script-\$(date +%Y%m%d)-2.log Script started, file is script-20131206.log</pre>	<p>TIME</p> <p>14:05</p>
---	--------------------------

52.
On a second terminal connected to the signer, KSO2 authenticates and authorizes the command

<pre>root@sign1: scamgr -k nz-dnssec-keystore Keystore = nz-dnssec-keystore.DDDDDD.{xxxxxxxx} (local) 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: disable multiadmin Initiating SO: nz-kso1 Authorize this command? (Y/Yes/N/No) [No]: y Authorization successful scamgr> quit</pre>	<p>TIME</p> <p>14:07</p>
--	--------------------------

53.
First terminal will show progress and the multimode will be disabled

<pre>Update: Authenticated security officers: nz-kso1 nz-kso2 Multi-admin mode disabled.</pre>	<p>TIME</p> <p>14:08</p>
--	--------------------------

54.
KSO1 proceeds to delete existing KSO3 role

<pre>scamgr{mca0@localhost, nz-kso1}> delete so nz-kso3 Delete security officer nz-kso3? (Y/Yes/N/No) [No]: y Security Officer nz-kso3 deleted.</pre>	<p>TIME</p> <p>14:08</p>
--	--------------------------

55.
KSO1 proceeds to delete existing KSO5 role

<pre>scamgr{mca0@localhost, nz-kso1}> delete so nz-kso5 Delete security officer nz-kso5? (Y/Yes/N/No) [No]: y Security Officer nz-kso5 deleted.</pre>	<p>TIME</p> <p>14:09</p>
--	--------------------------

56.
KSO3 creates its own account

<pre>scamgr{mca0@localhost, nz-ksol}> create so nz-kso3 Enter new security officer password: Confirm password: Security Officer nz-kso3 created successfully.</pre>	TIME 14:10
--	---------------

57.
KSO5 creates its own account

<pre>scamgr{mca0@localhost, nz-ksol}> create so nz-kso5 Enter new security officer password: Confirm password: Security Officer nz-kso5 created successfully.</pre>	TIME 14:11
--	---------------

58.
KSO1 verifies the list of Security Officers is complete

<pre>scamgr{mca0@localhost, nz-dsol}> 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 -----</pre>	TIME 14:12
---	---------------

59.
KSO1 enables newly created KSO3 and KSO5 accounts as authorized members of Multi-Admin mode

<pre>scamgr{mca0@localhost, nz-ksol}> enable authmember nz-kso3 Added multi-admin role to Security Officer nz-kso3. scamgr{mca0@localhost, nz-ksol}> enable authmember nz-kso5 Added multi-admin role to Security Officer nz-kso5.</pre>	TIME 14:13
---	---------------

60.
KSO1 confirms the list of authorized Multi-Admin Security Officers is complete

<pre>scamgr{mca0@localhost, nz-ksol}> 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 -----</pre>	TIME 14:13
--	---------------

61.
KSO1 activates the Multi-Admin mode for the keystore

<pre>scamgr{mca0@localhost, nz-ksol}> 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.</pre>	TIME 14:14
---	---------------

62.
KSO1 logs out from the board

<pre>scamgr{mca0@localhost, nz-ksol}> exit</pre>	TIME 14:14
---	---------------

Key Purging

Estimated time: 5 min

Delete all the keys stored in the HSM that are no longer needed.

63.

SA verifies the signer is the standby signer, output must indicate the **standby_signer** is **LOCAL**

<pre>root@sign1: get_active_signer active_signer: 192.168.62.14 FULLY_AGREE REMOTE standby_signer: 192.168.58.14 FULLY_AGREE LOCAL</pre>	TIME 14:15
--	---------------

64.

SA lists the contents of the HSM. It must contain the same number of keys as seen after the previous Key Generation Procedure

<pre>ods-hsmutil list sca6000 head -5 Listing keys in repository: sca6000 240 keys found. Repository ID Type ----- sca6000 160d29b6d32b301356a22f545e1a5ddd RSA/2048 sca6000 33b6e77e122419a7e6893d2c5e2bcffb RSA/2048 sca6000 9d893962239be58bfcd3fd45a6454a5 RSA/2048 sca6000 5ac0c4de0626543295d37bc850200f86 RSA/2048 sca6000 76394a2af741e324ad49646b4b59dd53 RSA/2048</pre>	TIME 14:17
--	---------------

65.

Proceed to delete all unused keys in active policies

<pre>sudo -u opensnssec ods-purge-keys.sh</pre>	TIME 14:18
---	---------------

66.

SA lists the contents of the HSM, to show a reduced number of keys. **NOTE:** the actual value listed may vary.

<pre>ods-hsmutil list sca6000 head -5 Listing keys in repository: sca6000 115 keys found.</pre>	TIME 14:19
---	---------------

Key generation

Estimated time: 15 min

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

67.

SA executes the script to generate the keys for all active policies

<pre>sudo -u opensnssec ods-keygen.sh P14M</pre>	TIME 14:22
--	---------------



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

68.

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

<pre>ods-hsmutil list sca6000 head -5 Listing keys in repository: sca6000 200 keys found. Repository ID Type ----- sca6000 160d29b6d32b301356a22f545e1a5ddd RSA/2048 sca6000 33b6e77e122419a7e6893d2c5e2bcffb RSA/2048 sca6000 9d893962239be58bfcd3fd45a6454a5 RSA/2048 sca6000 5ac0c4de0626543295d37bc850200f86 RSA/2048 sca6000 76394a2af741e324ad49646b4b59dd53 RSA/2048</pre>	TIME 14:22
--	---------------

Backup generation

Estimated time: 10 min

69.
SA switches to the second terminal

70.
SA executes backup script in the first terminal. The backup files will be written to /var/lib/dnssec/keygen/key-backup-YYYY-MM-DD.tar.gz

<pre>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.</pre>	TIME 14:23
---	-------------------

71.
KSO1 authorizes the backup using their password

<pre>Keystore = nz-dnssec-keystore.600121.{b129f5fa} (local) Security Officer Login: nz-ksol Security Officer Password: NOTICE: Please wait while the other required 1 security officers authenticate this command. This command will time out in 5 minutes.</pre>	TIME 14:25
--	-------------------

72.
SA executes the HSM interface in the second window

<pre>scamgr -k nz-dnssec-keystore Keystore = nz-dnssec-keystore.600121.{b129f5fa} (local)</pre>	TIME 14:25
---	---------------

73.
A second KSO logs into the HSM using the second terminal to authorize the backup.

<pre>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-ksol Authorize this command? (Y/Yes/N/No) [No]: Y Authorization successful</pre>	TIME 14:26
--	-------------------

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

74.
KSO closes the second HSM interface and window

<pre>scamgr> quit</pre>	TIME 14:26
----------------------------	---------------

75.
The first terminal will show the backup command was authorized and will proceed. Output will look like the following example:

84. SA plugs Flash Drive into the laptop

85.

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

<pre>lsusb -v -d 13fe:4200 grep -C 1 iProduct iManufacturer 1 iProduct 2 USB DISK 2.0 iSerial 3 070B516E2B29CC98</pre>	TIME 14:32
--	---------------

86.

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

<pre>scp sysadmin@sign1:/var/lib/dnssec/keygen/key-backup-*/ /media/MASTER_BACK/ Enter passphrase for key 'sysadmin-ssh-key': key-backup-YYYY-MM-DD.tar.gz 100% 453KB key-backup-YYYY-MM-DD.tar.gz.sha256sum 100% 95</pre>	TIME 14:33
--	---------------

87.

SA checks the backup file integrity

<pre>cd /media/MASTER_BACK sha256sum -c key-backup-YYYY-MM-DD.tar.gz.sha256sum key-backup-YYYY-MM-DD.tar.gz: OK</pre>	TIME 14:34
---	---------------

Creating Backup Operative Copies

Wellington Operative Backup Copy

Estimated time: 5 min

88.

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

Flash Drive Serial #

070B516D21A9B261

89. KGA hands over the Flash Drive to SA

90. SA plugs the FD into the laptop

91.

SA verifies the FD serial number matches the serial number recorded on the script. This command will show two serial numbers, one for the Master Backup and one for the Wellington Flash Drive.

<pre>lsusb -v -d 13fe:4200 grep -C 1 iProduct iManufacturer 1 iProduct 2 USB DISK 2.0 iSerial 3 070B516D21A9B261 -- iManufacturer 1 iProduct 2 USB DISK 2.0 iSerial 3 070B516E2B29CC98 -- iManufacturer 1 iProduct 2 USB DISK 2.0 iSerial 3 070B516E3BB4CE31</pre>	TIME 14:30
--	---------------

92.

SA copies the Master Backup Copy FD contents into the Wellington Operational Backup FD

<pre>rsync -avW /media/MASTER_BACK/ /media/WELLINGTON/</pre>	TIME 14:30
--	---------------

93.

SA checks the integrity of the backup

<pre>cd /media/WELLINGTON sha256sum -c key-backup-YYYY-MM-DD.tar.gz.sha256sum key-backup-YYYY-MM-DD.tar.gz: OK</pre>	TIME 14:36
--	---------------

94.

SA unmounts and unplugs the OBC FD

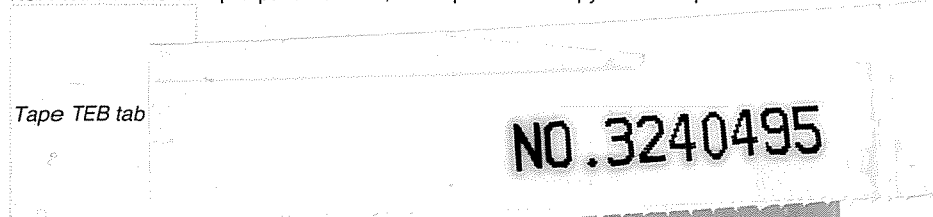
<pre>cd / eject /media/WELLINGTON</pre>	TIME 14:37
---	---------------

- 95. SA hands over the FD to the KGA
- 96. KGA labels a TEB as **WELLINGTON, <DATE>, NZRS DNSSEC Key Backup**
- 97. KGA records the TEB serial number in its script copy

TEB Serial #

3240495

- 98. KGA places the WELLINGTON OBC FD in the TEB
- 99. KGA places copy of the Device Backup Password in the TEB
- 100. KGA seals the TEB
- 101. KGA tears off the TEB pre-perforated tab, and tapes it to its copy of the script



- 102. KGA hands over the TEB to the SA
- 103. KSO1 confirms the TEB serial matches the script log and signs in acknowledgement

KSO1 signature

Auckland Operative Backup Copy

Estimated time: 5 min

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

Flash Drive Serial #

070B516044828874

- 105. KGA hands over the FD to the SA
- 106. SA plugs the FD into the laptop
- 107. SA verifies the FD serial number matches the serial number recorded on the script

<pre>lsusb -v -d 13fe:4200 grep -C 1 iProduct iManufacturer 1 iProduct 2 USB DISK 2.0 iSerial 3 070B516044828874 -- iManufacturer 1 iProduct 2 USB DISK 2.0 iSerial 3 070B516E2B29CC98 -- iManufacturer 1 iProduct 2 USB DISK 2.0 iSerial 3 070B516E3BB4CE31</pre>	<p>TIME</p> <p>14:41</p>
--	--------------------------

- 108. SA copies the MCB FD contents into the AUCKLAND OBC FD

<pre>rsync -avW /media/MASTER_BACK/ /media/AUCKLAND</pre>	<p>TIME</p> <p>14:42</p>
---	--------------------------

- 109. SA checks the integrity of the backup

<pre>cd /media/AUCKLAND sha256sum -c key-backup-YYYY-MM-DD.tar.gz.sha256sum key-backup-YYYY-MM-DD.tar.gz: OK</pre>	<p>TIME</p> <p>14:42</p>
--	--------------------------

- 110. SA unmounts and unplugs the OBC FD

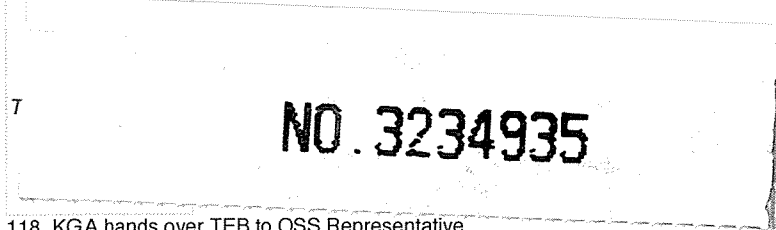
<pre>cd / eject /media/AUCKLAND</pre>	<p>TIME</p> <p>14:43</p>
---------------------------------------	--------------------------

- 111. SA hands over the FD to the KGA
- 112. KGA labels a TEB as **AUCKLAND, <DATE>, NZRS DNSSEC Key Backup**
- 113. KGA records the TEB serial number in its script copy

TEB Serial #

000 3234935

- 114. KGA places the AUCKLAND OBC FD in the TEB
- 115. KGA places copy of the Device Backup Password in the TEB
- 116. KGA seals the TEB
- 117. KGA tears off the TEB pre-perforated tab, and tapes it to its copy of the script



- 118. KGA hands over TEB to OSS Representative
- 119. OSS Representative confirms the TEB serial matches the script log and signs in acknowledgement

OSS Representative signature

Thomas Wule

- 120. OSS Representative hands over the TEB with serial number **3234860**, containing the Key Backup generated during the previous Key Generation Ceremony.
- 121. KGA confirms the TEB serial matches the previous script log and signs in acknowledgement

KGA signature

DM

Finishing steps

Estimated time: 3 min

- 122. SA unmounts and unplugs the MBC FD

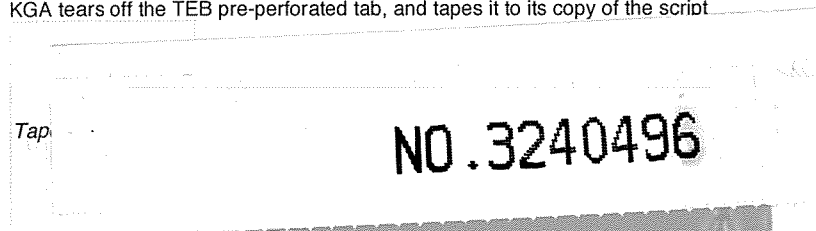
cd / eject /media/MASTER_BACK	TIME
----------------------------------	------

- 123. SA hands over the MBC FD to the KGA
- 124. KGA labels a TEB as **Master Copy, <DATE>, NZRS DNSSEC Key Backup**
- 125. KGA records the TEB serial number in its script copy

TEB Serial #

3240496

- 126. KGA places the MBC FD in the TEB
- 127. KGA places copy of the Device Backup Password in the TEB
- 128. KGA seals the TEB
- 129. KGA tears off the TEB pre-perforated tab, and tapes it to its copy of the script



- 130. KGA hands over TEB to KSO1
- 131. KSO1 confirms the TEB serial matches the script log and signs in acknowledgement

KSO1 signature

[Signature]

Closing steps

Estimated time: 12 min

- 132. SA finishes script logging

root@laptop> exit	TIME
-------------------	------

- 133. KGA selects Flash Drive labeled **Key Gen Copy** and hands it out to SA
- 134. SA plugs in the Flash Drive
- 135. SA verifies the FD serial number matches the serial number recorded on the script

070B516F148A2877

lsusb -v -d 13fe:4200 grep -C 1 iProduct iManufacturer 1 iProduct 2 USB DISK 2.0 iSerial 3 070B516E3BB4CE31 -- iManufacturer 1 iProduct 2 USB DISK 2.0 iSerial 3 070B516F148A2877	TIME 14:52
--	-------------------

- 136. SA copies **Key Gen Log** Flash Drive contents into **Key Gen Copy** Flash Drive

rsync -avW /media/KEY_GEN_LOG/ /media/KEYGEN_COPY	TIME 14:52
---	---------------

- 137. 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 14:54
---	---------------

- 138. 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 14:56
---	---------------

- 139. KGA records the sha256 digest into the script copy

sha256 digest

23:36:32:C1:D3:BA:75:43:	91:46:5F:B9:AE:55:6E 9B:BE:D1:06:19:B6:4B 8A:56:DF:D0:63:DE:AB F9:BE:6A:8B:E6:0A:FD A2:4E:C0:6E
05:AF:2D:11:75:0B:41:38:	
93:0B:74:9E:7E:6D:88:97:	
97:4A:B3:40:9F:EE:3D:F8:	

- 140. SA prints the script

lpr script-\$(date +%Y%m%d).ps	TIME 14:59
--------------------------------	---------------

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

cp script-\$(date +%Y%m%d).ps /media/KEY_GEN_LOG	TIME 15:00
--	---------------

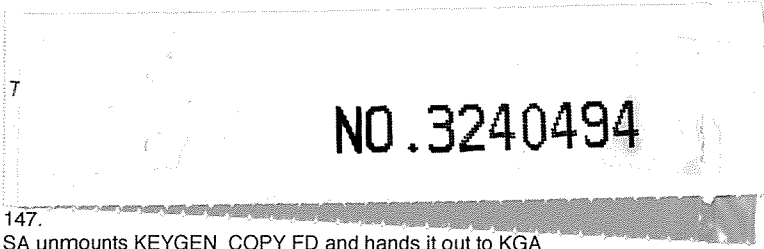
- 142. SA unmounts KEY_GEN_LOG FD

cd / eject /media/KEY_GEN_LOG	TIME 15:00
----------------------------------	---------------

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

TEB Serial # 3240494

- 145. KGA places KeyGen_Log FD in the TEB and seals it
- 146. KGA tears off the TEB pre-perforated tab, and tapes it to its copy of the script



147.
SA unmounts KEYGEN_COPY FD and hands it out to KGA

<code>cd / eject /media/KEYGEN_COPY</code>	TIME 15:02
--	---------------

148. SA shuts down laptop

149.

<code>shutdown -h now</code>	TIME 15:03
------------------------------	---------------

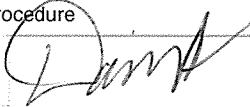
150. SA disconnects cables from laptop

151. Unplug laptop cables

152. KSO1 takes TEB containing Key Generation Log FD, TEB containing Master Backup Copy and copies of the script log for secure storage

153.

KGA signs off the key generation procedure

Signature	
Date/Time	16/02/2015 15:05

154. 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.