Sixth Key Generation

Created by Dane Foster , last modified by Josh Simpson just a moment ago

Version: 12.00

Last modification: March 10th, 2017 15:31

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 / TV	1
Long HDMI Cable	1
Printer	1
Photocopier	1
Flash Drives properly labelled and formatted	6
USB hub	1
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
Keystore backups from previous ceremony, provided by each representative	4

Participants

Role	Organization	Printed Name	Signature	Date	Time
DSO1	NZRS	Dane Foster		13/3/2017	10:06
SA/DSO2	NZRS	Jean-Marc Messina		13/3/17	10:08
KSO1	NZRS	Dave Baker		13/3/17	10.06
KSO2	NZRS	Jay Daley			
KSO3	NZRS	Brenda Wallace			
KGA/DSO3	NZRS	Josh Simpson		13/3/17	10.09
DSO4	OSS	Declan Brady		13/3/17	10:07
DSO5	NZRS	Daniel Griggs		13/3/2017	10:06
KSO5	NZRS	Sebastian Castro		13/3/2017	11:11

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. All participants 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. USB hub
    10. Tamper-Evident Bags
    11. Cello tape

Laptop setup

Estimated time: 15 min

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

19. SA verifies the integrity of the Live CD by comparing the digest

```
openssl dgst -c -sha256 /dev/sr0
SHA256(/dev/sr0)= f0:c1:51:a8:3d:4c:b3:ac:3d:26:16:57:54:76:06:78:
b4:47:56:56:12:4d:67:43:45:c5:75:66:26:10:3c:d8
```

TIME 10:33

Matches Record: (YES) / NO

20.
SA verifies time and date on the laptop

```
root@laptop# date
```

TIME 10:33

21.
KGA records date and time on their script copy

Date	Mon March 13 10:33:20 NZDT 2017
Time	10:33:20

22.
KGA selects USB hub and plugs into laptop and records the printed serial number on their script copy.

USB Hub Serial #	1402000723

23.
KGA selects Flash Drive labeled Utils, records the serial number on their script copy and hands it out to SA

UTILS FD Serial #	AAJJD746YYND2VC8

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

25.
SA elevates privileges to access the Flash Drive

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

TIME 10:36

26.
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 AAJJD746YYND2VC8
```

05DC:4B1D

TIME 10:40

27.
SA copies SSH key and config for access to signers to the laptop

```
mkdir /root/.ssh

chmod 0600 /root/.ssh

cp /media/UTILS/SA_KEY/id_rsa /root/.ssh/id_rsa

cp /media/UTILS/SA_KEY/config /root/.ssh/config

chmod 0600 /root/.ssh/id_rsa
```

TIME 10:42

28.
SA unmount and ejects UTILS FD

```
eject /media/UTILS
```

TIME 10:42

Access to the signing box

Estimated time: 5 min

29.
KGA selects Flash Drive labeled **KEYGEN LOG**, records the serial number on their script copy and hands it out to SA

KEYGEN LOG FD Serial #	AAWBIUCDHP5xZKQF

30. SA plugs in the Flash Drive. By default the Flash Drive will be auto-mounted and its contents available at **/media/KEYGEN_LOG**

31.
SA verifies the FD serial number matches the serial number recorded in the script

```
lsusb -v -d 05dc:a81d | grep -C 1 iProduct
iManufacturer 1
iProduct 2 USB Flash Drive
iSerial 3 AAWBIUCDHP5XZKQF
```

TIME 10:44

32.
SA starts logging via **script**

```
root@laptop# cd /media/KEYGEN_LOG
root@laptop# script script-$(date +%Y%m%d)-1.log
Script started, file is script-20131206.log
```

TIME 10:45

33.
SA accesses the standby signing box via SSH using their own account, providing their own SSH identity in the first terminal tab

```
ssh sysadmin@sign2.internal.srs.net.nz
```

TIME 10:46

34.
KGA checks the fingerprint for the server matches the records

sign1 fingerprint	b2:29:9f:b3:b9:b9:88:5b:4e:80:d6:c3:64:ff:ff:9b
sign2 fingerprint	ed:73:ee:03:6c:4c:c0:26:3a:e8:f4:cc:60:26:a1:81
srsplog1 fingerprint	ae:b0:a4:17:0c:8b:82:30:1c:bb:73:11:4a:4f:1e:84
srsslog1 fingerprint	a9:4c:d8:20:a9:66:ef:7c:0a:9d:60:f3:77:16:4c:b9

```
The authenticity of host 'sign2.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 10:47

Matches Record? (YES) / NO

35.
SA enters the directory /var/lib/dnssec/keygen. Files generated during the key generation procedure will be stored here for later retrieval.

```
admin@sign2: sudo -s
[sudo] password for admin:
[/home/admin]
root@sign2: cd /var/lib/dnssec/keygen
[/var/lib/dnssec/keygen]
root@sign2:
```

TIME 10:51

36.
In the second terminal tab, sudo to root and start the logging:

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

root@laptop# cd /media/KEYGEN_LOG

root@laptop# script script-$(date +%Y%m%d)-2.log
Script started, file is script-20131206-2.log
```

TIME 10:53

37.

And still in the second tab, login to the same signer and enter the same directory

```
ssh admin@sign2.internal.srs.net.nz
admin@sign2: sudo -s
[sudo] password for admin:
[/home/admin]
root@sign2: cd /var/lib/dnssec/keygen
[/var/lib/dnssec/keygen]
root@sign2:
```

TIME 10:55

38. Switch back to the first tab before proceeding.

HSM Verification

Estimated time: 5 min

39.
SA retrieves the HSM public key fingerprint

```
root@sign2: scadiag -f mca0
4fbd-91b8-f9e8-56a2-bc42-ad7d-321c-9846-f47f-2936
```

TIME 10:55

40.
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 in NZRS Staff, one of the existing DSO 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

41.
DSO5 access the board and authenticates themselves.

```
root@sign2: scamgr -D

Security Officer Login: nz-dso5

Security Officer Password:

scamgr{mca0@localhost, nz-dso5}>
```

TIME 10:58

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: 36:30:35:35:33:38
Key Fingerprint: 4fbd-91b8-f9e8-56a2-bc42-ad7d-321c-9846-f47f-2936
-------------------------------------------------------------
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 10:58

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

42.
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 10:58

43.
DSO2 creates its own account

```
scamgr{mca0@localhost, nz-dso5}> create so nz-dso2

Enter new security officer password:

Confirm password:

Security Officer nz-dso2 created successfully.
```

TIME 10:59

44.
DSO5 checks all expected DSOs accounts 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 10:59

45.
DSO5 logs out from the session

```
scamgr{mca0@localhost, nz-dso5}> quit
```

TIME 10:59

Key Purging

Estimated time: 5 min

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

46.
SA verifies the signer is the standby signer, output must indicate the **standby_signer** is **LOCAL**

```
root@sign2: get_active_signer
active_signer: 192.168.62.14|FULLY_AGREE|REMOTE
standby_signer: 192.168.58.14|FULLY_AGREE|LOCAL
```

TIME 11:00

47.
SA lists the contents of the HSM. It must contain the same number of keys as seen after the previous Key Generation Procedure

```
ods-hsmutil list sca6000 |head -5

Listing keys in repository: sca6000
230 keys found.
Repository ID Type
---------- -- ----
sca6000 5e8a6f9da462b82298088b833807fe37 RSA/2048
sca6000 a5ff380ed3a1da14e01446c12f36d6a7 RSA/2048
sca6000 7cd5390b10997cedc096deb6f09c60ec RSA/2048
sca6000 99bb4e22f67db09bd4d4023f23f89a3b RSA/2048
sca6000 c9aa6e3b333e773edc06ac09054e6f86 RSA/2048
```

TIME 11:01

48.
Proceed to delete all expired keys in active policies

```
sudo -u opendnssec ods-purge-keys.sh
```

TIME ~~10~~ 11:06

49.
SA lists the contents of the HSM, to show a reduced number of keys. **NOTE:** the actual value listed may vary.

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

132 keys

TIME ~~10~~ 11:06

Key generation

Estimated time: 15 min

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

50.
SA executes the script to generate the keys for all active policies

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

TIME 11:09

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

51.
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
200 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
```

235 Keys

TIME 11:09

Backup generation

Estimated time: 10 min

52.
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

```
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 11:12

53.
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 11:13

54.
SA executes the HSM interface in the second window

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

TIME 11:13

55.
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 11:14

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

56.
KSO closes the second HSM interface and window

```
scamgr> quit
```

TIME 11:15

57.
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-ksol                    KSo 5   KS01
Update: Authenticated security officers: nz-ksol 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
4a:8d:31:ef:ac:7f:e8:bf:b9:6d:bd:11:dc:aa:35:09:f8:79:99:15:45:b4:d6:a6:7b:40:3f:d9:df:07:c9:db

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 11:15

58.
DSO2 authorizes the device backup with their password

```
Security Officer Login: nz-dso2
Security Officer Password:
```

TIME 11:16

59. SA retrieves device password from KGA

60.
DSO2 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
29:ed:62:3a:d2:84:b6:7d:dd:20:a3:4f:82:e6:a5:86:44:ef:4c:bd:61:03:d8:9d:9b:c7:7e:38:0e:72:f6:02

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:
2c:2e:12:e2:3e:13:38:58:1f:68:59:77:83:19:f3:11
43:cb:10:50:cd:83:89:5d:2f:a4:29:1a:a5:18:85:2c )
```

TIME 11:18

61.
SA reads the digest from the screen, KGA records on its script copy

Keystore backup file digest	CE:D1:26:FF:A5:D8:9B:38 4D:39:43:1E:4F:93:F9:66 F5:D7:39:84:5E:83:58:C3 E2:C1:3A:46:D2:3B:D3:AC

62.
SA closes the root session

```
root@sign2: exit
```

TIME 11:19

63.
SA logs outs from the signing box

```
sysadmin@sign2: exit
Connection to sign2.internal.srs.net.nz closed.
```

TIME 11:19

Creating Master Backup Copy

Estimated time: 5 min

64.
KGA takes the Flash Drive labeled as **MASTER COPY** to serve as Master Copy Container. KGA records the serial number on its script copy.

MASTER COPY FD Serial #	AAWCATEOGCW9VXPR

65. KGA passes the Flash Drive to SA

66. SA plugs Flash Drive into the laptop

67.
SA verifies the FD serial number matches the serial number recorded on the script.

```
lsusb -v -d 13fe:4200 | grep -C 1 iProduct

iManufacturer 1
iProduct 2 USB DISK 2.0
iSerial 3 AAWCATEOGCW8VXPR
```

TIME 11:23

68.
SA copies the backup files from the signer to the Flash Drive

```
scp sysadmin@sign2:/var/lib/dnssec/keygen/key-backup-* /media/MASTER_COPY/
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
```

TIME 11:25

69.
SA checks the backup file integrity

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

TIME 11:27

Creating Backup Operative Copies

Wellington Operative Backup Copy

Estimated time: 5 min

70.
KGA picks Flash Drive labeled **WELLINGTON**, and records the serial number in its script copy.

Wellington FD Serial #	AAJ1RVH6CP2MSWAP

71. KGA hands over the Flash Drive to SA

72. SA plugs the FD into the laptop

73.
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 Copy and one for the Wellington Flash Drive.

```
lsusb -v -d 13fe:4200 | grep -C 1 iProduct
iManufacturer 1
iProduct 2 USB DISK 2.0
iSerial 3 AAJ1RVH6CP2MSWAP
--
iManufacturer 1
iProduct 2 USB DISK 2.0
iSerial 3 AAWCATEOGCW8VXPR

--
iManufacturer 1
iProduct 2 USB DISK 2.0
iSerial 3 AAWBIUCDHP5XZKQF
```

TIME 11:29

74.
SA copies the Master Backup Copy FD contents into the Wellington Operational Backup FD

```
rsync -avW /media/MASTER_COPY/ /media/WELLINGTON/
```

TIME 11:29

75.
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 11:30

76.
SA unmounts and unplugs the OBC FD

```
cd /
eject /media/WELLINGTON
```

TIME 11:30

77. SA hands over the FD to the KGA

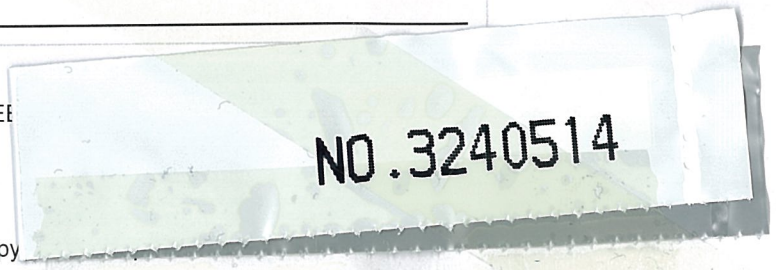
78. KGA labels a TEB as **WELLINGTON, <DATE>, NZRS DNSSEC Key Backup**

79.
KGA records the TEB serial number in its script copy

TEB Serial #	3240514

80. KGA places the WELLINGTON OBC FD in the TEB

81. KGA places copy of the Device Backup Password in the TEB



82. KGA seals the TEB

83.
KGA tears off the TEB pre-perforated tab, and tapes it to its copy
84. KGA hands over the TEB to KSO1

85.
KSO1 confirms the TEB serial matches the script log and signs in acknowledgement

KSO1 Signature	

Auckland Operative Backup Copy

Estimated time: 5 min

86.
KGA picks Flash Drive labeled **AUCKLAND**, and records the serial number in its script copy

AUCKLAND FD Serial #	AAKH3QAWOARAS3RP

87. KGA hands over the FD to the SA

88. SA plugs the FD into the laptop

89.
SA verifies the FD serial number matches the serial number recorded on the script

```
lsusb -v -d 05dc:a20b | grep -C 1 iProduct
iManufacturer 1
iProduct 2 USB Flash Drive
iSerial 3 AAKH3QAWOARAS3RP

--
iManufacturer 1
iProduct 2 USB DISK 2.0
iSerial 3 AAWCATEOGCW8VXPR

--
iManufacturer 1
iProduct 2 USB DISK 2.0
iSerial 3 AAWBIUCDHP5XZKQF
```

TIME 11:36

90.
SA copies the MCB FD contents into the AUCKLAND OBC FD

```
rsync -avW /media/MASTER_COPY/ /media/AUCKLAND
```

TIME 11:37

91.
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 11:37

92.
SA unmounts and unplugs the OBC FD

```
cd /
eject /media/AUCKLAND
```

TIME 11:37

93. SA hands over the FD to the KGA

94. KGA labels a TEB as **AUCKLAND, <DATE>, NZRS DNSSEC Key Backup**

95.
KGA records the TEB serial number in its script copy

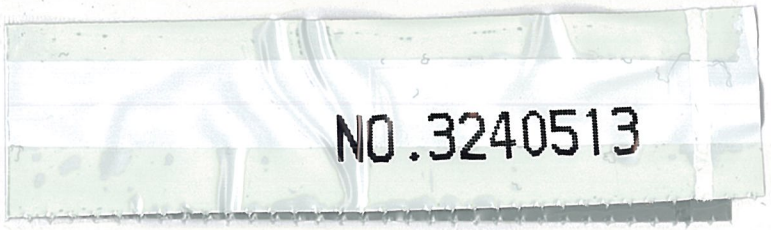
TEB SERIAL	3240513

96. KGA places the AUCKLAND OBC FD in the TEB

97. KGA places copy of the Device Backup Password in the TEB

98. KGA seals the TEB

99.
KGA tears off the TEB pre-perforated tab, and tapes it to its copy of the script



100. KGA hands over TEB to OSS Representative

101.
OSS Representative confirms the TEB serial matches the script log and signs in acknowledgement

OSS Rep Signature	

102. OSS Representative hands over the TEB with serial number **3240504,** containing the Key Backup generated during the previous Key Generation Ceremony.

103.
KGA confirms the TEB serial matches the previous script log and signs in acknowledgement

KGA Signature	

Finishing steps

Estimated time: 3 min

104.
SA unmounts and unplugs the MBC FD

```
cd /
eject /media/MASTER_COPY
```

TIME 11:40

105. SA hands over the MBC FD to the KGA

106. KGA labels a TEB as **Master Copy, <DATE>, NZRS DNSSEC Key Backup**

107.
KGA records the TEB serial number in its script copy

TEB SERIAL	

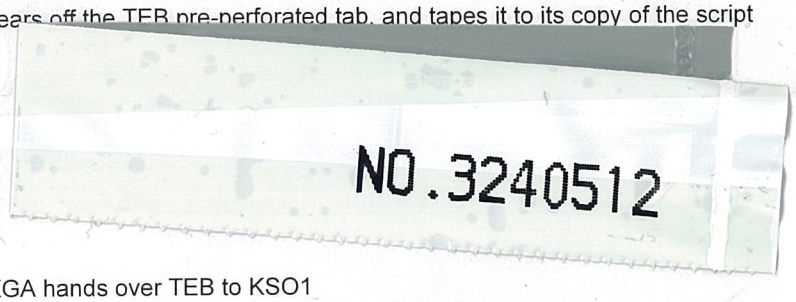
3240512

108. KGA places the MBC FD in the TEB

109. KGA places copy of the Device Backup Password in the TEB

110. KGA seals the TEB

111.
KGA tears off the TEB pre-perforated tab, and tapes it to its copy of the script



112. KGA hands over TEB to KSO1

113.
KSO1 confirms the TEB serial matches the script log and signs in acknowledgement

KSO1 Signature	

Closing steps

Estimated time: 12 min

114.
SA finishes script logging

```
root@laptop> exit
```

TIME 11:45

115. KGA selects Flash Drive labeled **Key Gen Copy** and hands it out to SA

116. SA plugs in the Flash Drive

117.
SA verifies the FD serial number matches the serial number recorded on the script

```
lsusb -v -d 05dc:a81d | grep -C 1 iProduct
iManufacturer 1
iProduct 2 USB DISK 2.0
iSerial 3 AAROGYC0CNSPKAG5
--
iManufacturer 1
iProduct 2 USB DISK 2.0
iSerial 3 AAWBIUCDHP5XZKQF
```

TIME 11:46

118.
SA copies **Key Gen Log** Flash Drive contents into **Key Gen Copy** Flash Drive

```
rsync -avW /media/KEYGEN_LOG/ /media/KEYGEN_COPY
```

TIME 11:48

119.
SA generates a printable copy of the script

```
cd /media/KEYGEN_COPY

enscript -G -U 2 -o script-$(date +%Y%m%d)-1.ps script-$(date +%Y%m%d)-1.log

enscript -G -U 2 -o script-$(date +%Y%m%d)-2.ps script-$(date +%Y%m%d)-2.log
```

TIME 11:49

120.
SA generates sha256 digest for the printable copy of the script from each terminal window. Output should look like this:

```
openssl dgst -c -sha256 script-$(date +%Y%m%d)-1.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

openssl dgst -c -sha256 script-$(date +%Y%m%d)-2.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 11:53

121.
KGA records the sha256 digest into the script copy

Script 1 Keystore backup file digest	5B:73:13:19:49:85:DA:21 BF:A3:2B:80:73:2F:0C:6A BD:E0:42:A7:C3:2E:76:65 82:F1:4C:C2:3D:1C:57:3F
Script 2 Keystore backup file digest	5C:7C:25:8C:36:EF:AC:52 C9:17:B0:59:E6:7C:2A:0E B9:29:66:DC:F3:47:88:25 D8:4D:AC:76:72:E0:E9:5D

122.
SA prints the script

```
lpr script-$(date +%Y%m%d)-1.ps
lpr script-$(date +%Y%m%d)-2.ps
```

TIME 11:53

123.
SA copies the printable copy to the **KEYGEN LOG** Flash Drive

```
cp script-$(date +%Y%m%d)-1.ps /media/KEYGEN_LOG
cp script-$(date +%Y%m%d)-2.ps /media/KEYGEN_LOG
```

TIME 11:55

124.
SA unmounts KEY_GEN_LOG FD

```
cd /
eject /media/KEYGEN_LOG
```

TIME 11:56

125. SA unplugs Flash Drive and hands it out to KGA

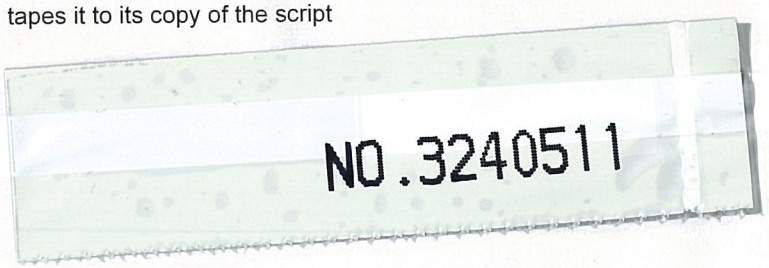
126.
KGA takes a TEB and records the serial number in its script copy

TEB Serial #	3240511

127. KGA places KEYGEN_LOG FD in the TEB and seals it

128. KGA labels the TEB as KEYGEN_LOG and seals it

129.
KGA tears off the TEB pre-perforated tab, and tapes it to its copy of the script



130.
SA unmounts KEYGEN_COPY FD and hands it out to KGA

```
cd /
eject /media/KEYGEN_COPY
```

TIME 11:59

131.
SA shuts down laptop

```
shutdown -h now
```

TIME 11:59

132. SA disconnects cables from laptop

133. Unplug laptop cables

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

135.
KGA signs off the key generation procedure

KGA Signature	
Date / Time	12:00 13-03-2017

136. 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.