KGA Copy

Third Key Generation

Version:	26
Last modification:	Dec 05, 2013 16:07

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

Keystore backups from previous ceremony, provided by each representative	4

Participants

Role	Organization	Printed Name	Signature	Date	Time
KGA / DSO5	NZRS	Sebastian Castro		9:09	6-Dec-2013
SA	Catalyst IT	James Dempsey		9:09	6-Dec-2013
DSO1/ KSO1	NZRS	Dave Baker		9:10	6/12/13
DSO2	Knossos	John Rumsey	John Rumsey	6:12:13	09:10
DSO3	Catalyst IT	Andrew Ruthven		9:15	6/12/13
DSO4	OSS	Vince Hagon		6-12-13	09:10
KSO2	NZRS	Jay Daley		6/12/13	09:56

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: 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. 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 9:13
    10. Connects power cable, network cable, and projector
    11. Powers up laptop, hit ENTER to access boot menu
    12. Boot-up laptop using a bootable CD
    13. Enables display
    14. Configures printer and print test page 9:23
    15. Open terminal, and maximize for visibility

16.
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 9:24

Matches record? [✓ YES] / NO 9:33

17.
SA verifies time and date on the laptop

```
root@laptop# date
```

TIME 9:36

18.
KGA records date and time on their script copy

Date: 9:36 NZDT

Time: Fri Dec 6 2013

Access to the signing box

Estimated time: 5 min

19.
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 # 4C53200001 0910123021

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

21.
SA elevate privileges to access the Flash Drive

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

TIME 9:37

22.
SA verifies the FD serial number matches the serial number recorded on the script

```
lsusb -v -d 0x0781:0x5572 | grep -C 1 iProduct
iManufacturer 1 SanDisk
iProduct 2 Cruzer Switch
iSerial 3 4C532000010910123021
```

TIME 9:39

23.
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-20131206.log
```

TIME 9:39

24.
SA accesses the standby 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 9:43

25.
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**

```
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 9:44

Matches record? [✓ YES] / NO

26.
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 9:44

HSM Verification

Estimated time: 5 min

27.
SA retrieves the HSM public key fingerprint

```
sysadmin@sign1: scadiag -f mca0
4fbd-91b8-f9e8-56a2-bc42-ad7d-321c-9846-f47f-2936
```

TIME 9:45

28.

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

Matches record? [✓ YES] / NO

Key Purging

Estimated time: 5 min

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

29.
SA verifies the signer is the standby signer, output must indicate the **standby_signer** is **LOCAL**

```
sysadmin@sign1: get_active_signer
active_signer: 192.168.62.14|FULLY_AGREE|REMOTE
standby_signer: 192.168.58.14|FULLY_AGREE|LOCAL
```

TIME 9:46

30.
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
240 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 9:46

31.
Proceed to delete all unused keys in active policies

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

TIME 9:47

32.
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.
```

121 keys

TIME 9:47

Key generation

Estimated time: 15 min

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

33.
SA executes the script to generate the keys for all active policies

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

TIME 9:49

> 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

34.
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
```

229 keys

TIME
9:49

Backup generation

Estimated time: 10 min

35.
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
9:51

36.
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

```
sudo -s
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

37.
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

38.

SA executes the HSM interface in the second window

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

TIME

39.

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

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

40.

SA closes the second HSM interface and window

```
scamgr> quit
```

TIME 10:03

41.

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-backu
p-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:f8:79:99:1
5:45: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

42.
DSO1 authorizes the device backup with their password

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

TIME
10:05

43.
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
29:ed:62:3a:d2:84:b6:7d:dd:20:a3:4f:82:e6:a5:86:44:ef:4c:b
d: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

44.
SA reads the digest from the screen, KGA records on its script copy

Keystore backup file digest

e6 : 43 : 8e : 5b : 41 : 62 : da : fc :
1b : ce : 92 : b9 : 1e : 88 : c0 : 59 :
77 : 29 : 8b : a2 : 07 : 3c : bc : c4 :

9f : 21 : 9c : c2 : 95 : f2 : d0 : 34

45.
SA closes the root session

```
root@sign1: exit
```

TIME 10:09

46.
SA logs outs from the signing box

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

TIME 10:09

Creating Master Backup Copy

Estimated time: 5 min

47.
KGA takes the Flash Drive labeled as **Master Copy** to serve as Master Copy Container. KGA records the serial number on its script copy.

Flash Drive Serial # 0019e06b5884fb61874a20ab

48. KGA passes the Flash Drive to SA
49. SA plugs Flash Drive into the laptop
50.
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
```

TIME 10:12

51.
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 10:13

52.
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 10:14

Creating Backup Operative Copies

Wellington Operative Backup Copy

Estimated time: 5 min

53.
KGA picks Flash Drive labeled **WELLINGTON**, and records the serial number in its script copy.

Flash Drive Serial # 00147854 4884 fb61874220 4a

54. KGA hands over the Flash Drive to SA
55. SA plugs the FD into the laptop
56.

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.

```
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
```

TIME 10:15

57.

SA copies the MBC FD contents into the Wellington OBC FD

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

TIME 10:15

58.

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 10:15

59.

SA unmounts and unplugs the OBC FD

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

TIME

60. SA hands over the FD to the KGA
61. KGA labels a TEB as **WELLINGTON, <DATE>, NZRS DNSSEC Key Backup**
62.

KGA records the TEB serial number in its script copy

TEB Serial # 3187090

63. KGA places the WELLINGTON OBC FD in the TEB
64. KGA places copy of the Device Backup Password in the TEB
65. KGA seals the TEB
66.

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

NO.3187090

67. KGA hands over the TEB to Catalyst Representative

68.

Catalyst Representative confirms the TEB serial matches the script log and signs in acknowledgement

Catalyst Representative signature	

69. Catalyst Representative hands over the TEB with serial number **3234864,** containing the Key Backup generated during the previous Key Generation Ceremony.

70.

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

KGA signature	

Albany Operative Backup Copy

Estimated time: 5 min

71.

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

Flash Drive Serial # 0019e06b587b fb61 87 432154

72. KGA hands over the FD to the SA

73. SA plugs the FD into the laptop

74.

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
```

TIME 10:20

75.

SA copies the MCB FD contents into the Albany OBC FD

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

TIME 10:21

76.

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 10:21

77.

SA unmounts and unplugs the OBC FD

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

TIME 10:21

78. SA hands over the FD to the KGA

79. KGA labels a TEB as **ALBANY, <DATE>, NZRS DNSSEC Key Backup**

80.

KGA records the TEB serial number in its script copy

TEB Serial # 3234 861

81. KGA places the ALBANY OBC FD in the TEB

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

83. KGA seals the TEB

84.

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

NO.3234861

85. KGA hands over the TEB to Knossos Representative

86.

Knossos Representative confirms the TEB serial matches the script log and signs in acknowledgement

Knossos Representative signature	John R Rumsey

87. Knossos Representative hands over the TEB with serial number **3234868,** containing the Key Backup generated during the previous Key Generation Ceremony.

88.

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

KGA signature	AG

Auckland Operative Backup Copy

Estimated time: 5 min

89.

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

Flash Drive Serial # 0019e06b0842fb6187ae20fc

90. KGA hands over the FD to the SA

91. SA plugs the FD into the laptop

92.

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
```

TIME 10:25

93.

SA copies the MCB FD contents into the AUCKLAND OBC FD

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

TIME 10:25

94.

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 10:25

95.
SA unmounts and unplugs the OBC FD

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

TIME

96. SA hands over the FD to the KGA
97. KGA labels a TEB as **AUCKLAND, <DATE>, NZRS DNSSEC Key Backup**
98.
KGA records the TEB serial number in its script copy

TEB Serial # 3234860

99. KGA places the AUCKLAND OBC FD in the TEB
100. KGA places copy of the Device Backup Password in the TEB
101. KGA seals the TEB
102.
KGA tears off the TEB pre-perforated tab, and tapes it to its copy of the script

NO. 3234860

103. KGA hands over TEB to OSS Representative
104.
OSS Representative confirms the TEB serial matches the script log and signs in acknowledgement

OSS Representative signature	

105. OSS Representative hands over the TEB with serial number **3234867,** containing the Key Backup generated during the previous Key Generation Ceremony.
106.
KGA confirms the TEB serial matches the previous script log and signs in acknowledgement

KGA signature	

Finishing steps

Estimated time: 3 min

107.
SA unmounts and unplugs the MBC FD

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

TIME 10:28

108. SA hands over the MBC FD to the KGA
109. KGA labels a TEB as **Master Copy, <DATE>, NZRS DNSSEC Key Backup**
110.
KGA records the TEB serial number in its script copy

TEB Serial # 3234859

111. KGA places the MBC FD in the TEB
112. KGA places copy of the Device Backup Password in the TEB
113. KGA seals the TEB
114.

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

NO.3234859

115. KGA hands over TEB to KSO1

116.

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

KSO1 signature	

117. KSO1 hands over the TEB with serial number **3187084,** containing the Key Backup generated during the previous Key Generation Ceremony.

118.

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

KGA signature	

Closing steps

Estimated time: 12 min

119.

SA finishes script logging

```
root@laptop> exit
```

TIME 10:31

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

121. SA plugs in the Flash Drive

122.

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

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

TIME

123.

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

124.

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

125.

KGA records the sha256 digest into the script copy

sha256 digest b6 : 9e : b6 : ee : a2 : 96 : 44 : d9 :
7b : 38 : 18 : 1a : 32 : fa : 17 : 9e :

c5 : 00 : 98 : ee : c8 : 3d : a0 : 03 :
93 : 3e : 8c : 7b : 90 : fb : 54 : a7

126.
SA prints the script

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

TIME 10:36

127.
SA copies the printable copy to the **Key Gen Log**Flash Drive

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

TIME 10:39

128.
SA unmounts KEY_GEN_LOG FD

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

TIME 10:39

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

130.
KGA takes a TEB and records the serial number in its script copy

TEB Serial # 3234858

131. KGA places KeyGen_Log FD in the TEB and seals it

132.
KGA tears off the TEB pre-perforated tab, and tapes it to its copy of the script

NO.3234858

133.
SA unmounts KEYGEN_COPY FD and hands it out to KGA

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

TIME 10:43

134. SA unmounts and unplugs the Flash Drive carrying his key

135.
SA shuts down laptop

```
shutdown -h now
```

TIME 10:43

136. SA disconnects cables from laptop

137. Unplug laptop cables

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

139.
KGA signs off the key generation procedure

Signature	
Date/Time	10:43 6-12-2013

140. 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.