Fifth Key Generation

Created by Dane Foster , last modified 2 minutes ago

Version:	29.00
Last modification:	Feb 3, 2016 09:14

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
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

KGA/DSO1	NZRS	Dane Foster	3/2/16 14:01
SA/DSO2	NZRS	Mike Forbes	3/2/16 14:00
KSO1	NZRS	Dave Baker	3/12/16 14:02
KSO2	NZRS	Jay Daley	
KSO3	NZRS	Brenda Wallace	
DSO3	NZRS	Josh Simpson	3/2/16 14:01
DSO4	OSS	Tom Weber	3/2/16 13:59
DSO5	NZRS	Daniel Griggs	3/2/16 14:02
KSO5	NZRS	Sebastian Castro	3/2/16 14:49

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:a8:3a: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 14:25

Matches record? YES / NO

20.
SA verifies time and date on the laptop

```
root@laptop# date
```

TIME: 14:25

21.
KGA records date and time on their script copy

Date: Wed Feb 3 2016

Time: 14:25:36

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

USB hub serial # 1402000723

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

Flash Drive Serial # 070B5160B828837

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: 14:27

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 070B516D1B828837
```

TIME: 14:28

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: 14:29

28.
SA unmount and ejects UTILS FD

```
eject /media/UTILS
```

TIME: 14:29

Access to the signing box

Estimated time: 5 min

29.
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 # AAY5CQES5GT0BLY

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 AAY5CQES5GT0YBLY
```

TIME: 14:31

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 14:32

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 14:32

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 14:33

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.

```
sysadmin@sign2: sudo -s
[sudo] password for sysadmin:
[/home/sysadmin]
root@sign2: cd /var/lib/dnssec/keygen
[/var/lib/dnssec/keygen]
root@sign2:
```
TIME 14:35

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 14:36

37.
And still in the second tab, login to the same signer and enter the same directory

```
ssh sysadmin@sign2.internal.srs.net.nz
sysadmin@sign2: sudo -s
[sudo] password for sysadmin:
[/home/sysadmin]
root@sign2: cd /var/lib/dnssec/keygen
[/var/lib/dnssec/keygen]
root@sign2:
```
TIME 14:38

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 14:39

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 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

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 14:39

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 14:39

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

42.
DSO4 creates its own account

```
scamgr{mca0@localhost, nz-dso5}> create so nz-dso4

Enter new security officer password:

Confirm password:

Security Officer nz-dso4 created successfully.
```

TIME 14:42

43.
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 14:42

44.
DSO1 logs out from the session

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

TIME 14:42

Key Purging

Estimated time: 5 min

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

45.
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 14:43

46.
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 14:44

47.
Proceed to delete all expired keys in active policies

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

TIME 14:45

48.
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.
```

TIME 14:45

Key generation

Estimated time: 15 min

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

49.
SA executes the script to generate the keys for all active policies

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

TIME 14:47

> 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

50.
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
```

TIME 14:47

Backup generation

Estimated time: 10 min

51.
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 14:49 14:54

52.
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 14:49/14:54

53.
SA executes the HSM interface in the second window

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

TIME 14:49/14:55

54.
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 14:50/14:56

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

55.
KSO closes the second HSM interface and window

```
scamgr> quit
```

TIME 14:50/14:56

56.
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
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 14:51

57.
DSO2 authorizes the device backup with their password

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

TIME 14:56

58. SA retrieves device password from KGA
59.
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 14:57

60.
SA reads the digest from the screen, KGA records on its script copy

Keystore backup file digest

~~B9:FD:71:10:F8:87:9C:15~~ DA:33:58:B2:C5:6A:43:9D
~~11:39:51:34:5E:C0:B6:CA~~ 5B:C2:BF:05:97:DA:D3:48
~~FD:9E:DD:66:4F:7F:AA:03~~ CE:18:F6:2F:46:AE:E3:66
~~2F:11:97:9F:36:C2:06:96~~ 99:CB:9C:FD:CC:62:C8:19

61.
SA closes the root session

```
root@sign2: exit
```
TIME 15:02

62.
SA logs outs from the signing box

```
sysadmin@sign2: exit
Connection to sign2.internal.srs.net.nz closed.
```
TIME 15:02

Creating Master Backup Copy

Estimated time: 5 min

63.
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 # 070B47248E6E9214

64. KGA passes the Flash Drive to SA
65. SA plugs Flash Drive into the laptop
66.
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 070B47248E6E9214
```
TIME 15:03

67.
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 15:04

68.
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 15:04

Creating Backup Operative Copies

Wellington Operative Backup Copy

Estimated time: 5 min

69.
KGA picks Flash Drive labeled **WELLINGTON**, and records the serial number in its script copy.

Flash Drive Serial # 070B4B28B5B2DC80

70. KGA hands over the Flash Drive to SA
71. SA plugs the FD into the laptop
72.
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 070B4B28B5B2DC80
--
iManufacturer 1
iProduct 2 USB DISK 2.0
iSerial 3 070B516E2B29CC98

--

iManufacturer 1
iProduct 2 USB DISK 2.0

iSerial 3 070B47248E6E9214
```
TIME 15:06

73.
SA copies the Master Backup Copy FD contents into the Wellington Operational Backup FD

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

TIME 15:07

74.
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 1507

75.
SA unmounts and unplugs the OBC FD

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

TIME

76. SA hands over the FD to the KGA
77. KGA labels a TEB as **WELLINGTON, <DATE>, NZRS DNSSEC Key Backup**
78.
KGA records the TEB serial number in its script copy

TEB Serial # 32400505

79. KGA places the WELLINGTON OBC FD in the TEB
80. KGA places copy of the Device Backup Password in the TEB
81. KGA seals the TEB
82.
KGA tears off the TEB pre-perforated tab, and tapes it to its copy of the script

NO.3240505

83. KGA hands over the TEB to KSO1

84.
KSO1 confirms the TEB serial matches the script log and signs in acknowledgement

KSO1 signature	

Auckland Operative Backup Copy

Estimated time: 5 min

85.
KGA picks Flash Drive labeled **AUCKLAND**, and records the serial number in its script copy

Flash Drive Serial # AAGPYJNPZNYEELGQ

86. KGA hands over the FD to the SA
87. SA plugs the FD into the laptop
88.
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 AAGPYJNPZNYEELGQ
--
iManufacturer 1
iProduct 2 USB DISK 2.0
iSerial 3 070B516E2B29CC98

--

iManufacturer 1
iProduct 2 USB DISK 2.0

iSerial 3 070B516E3BB4CE31
```

TIME 15:13

89.
SA copies the MCB FD contents into the AUCKLAND OBC FD

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

TIME 15:14

90.
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 15:14

91.
SA unmounts and unplugs the OBC FD

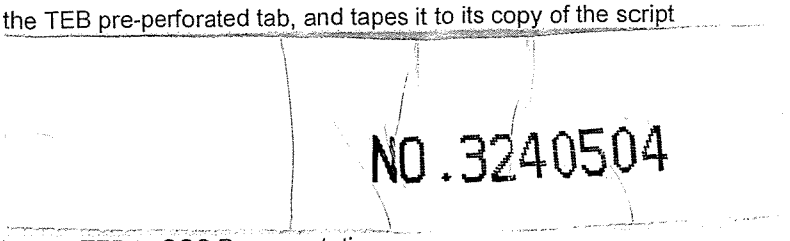
```
cd /
eject /media/AUCKLAND
```

TIME 15:14

92. SA hands over the FD to the KGA
93. KGA labels a TEB as **AUCKLAND, <DATE>, NZRS DNSSEC Key Backup**
94.
KGA records the TEB serial number in its script copy

TEB Serial # 3240504

95. KGA places the AUCKLAND OBC FD in the TEB
96. KGA places copy of the Device Backup Password in the TEB
97. KGA seals the TEB
98.
KGA tears off the TEB pre-perforated tab, and tapes it to its copy of the script



99. KGA hands over TEB to OSS Representative
100.
OSS Representative confirms the TEB serial matches the script log and signs in acknowledgement

OSS Representative signature	Thomas P. Wilson

101. OSS Representative hands over the TEB with serial number **3234935**, containing the Key Backup generated during the previous Key Generation Ceremony.
102.
KGA confirms the TEB serial matches the previous script log and signs in acknowledgement

KGA signature	

Finishing steps

Estimated time: 3 min

103.
SA unmounts and unplugs the MBC FD

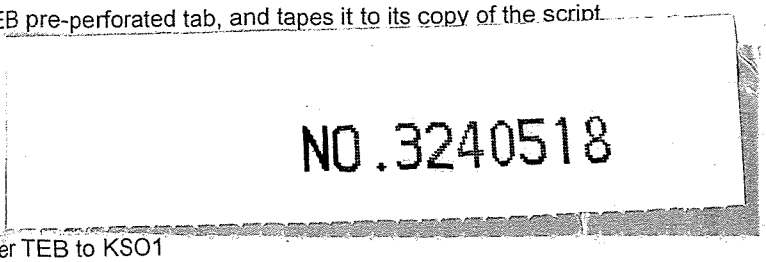
```
cd /
eject /media/MASTER_~~BACK~~
                    COPY
```

TIME 15:21

104. SA hands over the MBC FD to the KGA
105. KGA labels a TEB as **Master Copy, <DATE>, NZRS DNSSEC Key Backup**
106.
KGA records the TEB serial number in its script copy

TEB Serial # 3240518

107. KGA places the MBC FD in the TEB
108. KGA places copy of the Device Backup Password in the TEB
109. KGA seals the TEB
110.
KGA tears off the TEB pre-perforated tab, and tapes it to its copy of the script



111. KGA hands over TEB to KSO1
112.
KSO1 confirms the TEB serial matches the script log and signs in acknowledgement

KSO1 signature

Closing steps

Estimated time: 12 min

113.
SA finishes script logging

```
root@laptop> exit
```

TIME 15:23

114. KGA selects Flash Drive labeled **Key Gen Copy** and hands it out to SA
115. SA plugs in the Flash Drive
116.
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 AAAEP38OH4N1LFY4
--
iManufacturer 1
iProduct 2 USB DISK 2.0
iSerial 3 070B516F148A2877
```

SRARE AAT7VAJ5M3EHA[illegible]OT

TIME ~~15:24~~ 15:30

117.
SA copies **Key Gen Log** Flash Drive contents into **Key Gen Copy** Flash Drive

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

TIME 15:31

118.
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 15:31

119.
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 15:31

120.
KGA records the sha256 digest into the script copy

	script1							script2							
sha256 digest	7D	A8	8B	D3	53	38	C8	3F	B0	A1	80	9C	DE	N4	B5	70
	13	6E	74	1F	0B	29	93	45	4F	91	E6	18	6B	B3	F2	BF
	56	A4	44	15	AF	55	33	92	C7	4D	15	14	1D	D4	C3	9D
	CE	03	DE	90	C4	9C	55	05	77	CE	AE	CA	1F	67	15	8C

121.
SA prints the script

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

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

TIME 15:36

122.
SA copies the printable copy to the **Key Gen 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 15:37

123.
SA unmounts KEY_GEN_LOG FD

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

TIME 15:37

124. SA unplugs Flash Drive and hands it out to KGA
125.
KGA takes a TEB and records the serial number in its script copy

TEB Serial # 3240517

126. KGA places KeyGen_Log FD in the TEB and seals it
127. KGA labels the TEB as KEYGEN_LOG and seals it
128.
KGA tears off the TEB pre-perforated tab, and tapes it to its copy of the script

NO.3240517

129.
SA unmounts KEYGEN_COPY FD and hands it out to KGA

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

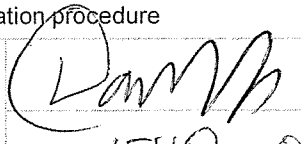
TIME 15:41

130. SA shuts down laptop
131.

```
shutdown -h now
```

TIME 15:41

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

Signature	
Date/Time	15:48 03/02/2014

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.