Sixth Key Generation

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

<table>
<thead>
<tr>
<th>Description</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laptop</td>
<td>1</td>
</tr>
<tr>
<td>CD with Live Linux Distribution</td>
<td>3</td>
</tr>
<tr>
<td>Projector / TV</td>
<td>1</td>
</tr>
<tr>
<td>Long HDMI Cable</td>
<td>1</td>
</tr>
<tr>
<td>Printer</td>
<td>1</td>
</tr>
<tr>
<td>Photocopier</td>
<td>1</td>
</tr>
<tr>
<td>Flash Drives properly labelled and formatted</td>
<td>6</td>
</tr>
<tr>
<td>USB hub</td>
<td>1</td>
</tr>
<tr>
<td>Spare formatted Flash Drives</td>
<td>2</td>
</tr>
<tr>
<td>Tamper-Evident Bags</td>
<td>6</td>
</tr>
<tr>
<td>Pre-generated secure password set for device backup</td>
<td>2</td>
</tr>
<tr>
<td>Sysadmin brings ssh key to access the signer</td>
<td>1</td>
</tr>
<tr>
<td>Hard copies of this script</td>
<td>8</td>
</tr>
<tr>
<td>Copy of previous Key Generation Procedure script</td>
<td>1</td>
</tr>
<tr>
<td>Copy of previous HSM restoration from Backup script</td>
<td>1</td>
</tr>
<tr>
<td>Participant sign-in sheet</td>
<td>1</td>
</tr>
<tr>
<td>Keystore backups from previous ceremony, provided by each representative</td>
<td>4</td>
</tr>
</tbody>
</table>

Participants
<table>
<thead>
<tr>
<th>Role</th>
<th>Organization</th>
<th>Printed Name</th>
<th>Signature</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSO1</td>
<td>NZRS</td>
<td>Dane Foster</td>
<td>[Signature]</td>
<td>13/3/17</td>
<td>10:06</td>
</tr>
<tr>
<td>SA/DSO2</td>
<td>NZRS</td>
<td>Jean-Marc Messina</td>
<td></td>
<td>13/3/17</td>
<td>10:08</td>
</tr>
<tr>
<td>KSO1</td>
<td>NZRS</td>
<td>Dave Baker</td>
<td></td>
<td>13/3/17</td>
<td>10:06</td>
</tr>
<tr>
<td>KSO2</td>
<td>NZRS</td>
<td>Jay Daley</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KSO3</td>
<td>NZRS</td>
<td>Brenda Wallace</td>
<td></td>
<td>13/3/17</td>
<td>10:08</td>
</tr>
<tr>
<td>KGA/DSO3</td>
<td>NZRS</td>
<td>Josh Simpson</td>
<td></td>
<td>13/3/17</td>
<td>10:09</td>
</tr>
<tr>
<td>DSO4</td>
<td>OSS</td>
<td>Declan Brady</td>
<td></td>
<td>13/3/17</td>
<td>10:07</td>
</tr>
<tr>
<td>DSO5</td>
<td>NZRS</td>
<td>Daniel Griggs</td>
<td></td>
<td>13/3/17</td>
<td>10:06</td>
</tr>
<tr>
<td>KSO5</td>
<td>NZRS</td>
<td>Sebastian Castro</td>
<td></td>
<td>13/3/17</td>
<td></td>
</tr>
</tbody>
</table>

### 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
Matches Record:  
**YES**  NO

20. SA verifies time and date on the laptop

```
root@laptop# date
Mon Mar 13 10:33:20 NZDT 2017
```

21. KGA records date and time on their script copy

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mon March 13 10:33:20</strong></td>
<td><strong>10:33</strong></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>USB Hub</th>
<th>Serial #</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1462666723</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>UTILS FD</th>
<th>Serial #</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AAJJD746YYNDL2VC8</td>
</tr>
</tbody>
</table>

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#
```

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

```
lsusb -v -d 0x0a:0x0 | grep -C 1 iProduct
iProduct 1
iProduct 2 USB DISK 2.0
iSerial 3 AAJJD746YYNDL2VC8
```

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
```

28. SA unmounts and ejects UTILS FD

```
eject /media/UTILS
```
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 #](image)

AAWBIUCPHP5XZKQF

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 65dc:a81d | grep -C 1 iProduct
  iManufacturer: iProduct
  iProduct: 2 USB Flash Drive
  iSerial: 3 AAWBIUCPHP5XZKQF
```

**TIME:** 10:44

32. SA starts logging via script

```
root@laptop# cd /media/KEYGEN_LOG
root@laptop# script script-9(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 eyeadmin@sign2.internal.srs.net.nz
```

**TIME:** 10:46

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

- **SRSlog1 fingerprint**: ae:b0:a4:17:0c:8b:82:30:1c:bb:73:11:4a:4f:1e:84

The authenticity of host 'sign2.internal.srs.net.nz (192.168.58.14)' can't be established.
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-9(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

```bash
ssh admin@sign2.internal.srs.net.nz
admin@sign2: sudo -x
/home/admin
root@sign2: cd /var/lib/dnsssec/keygen
/var/lib/dnsssec/keygen
root@sign2:
```

38. Switch back to the first tab before proceeding.

**HSM Verification**  
*Estimated time: 5 min*

39. SA retrieves the HSM public key fingerprint

```bash
root@sign2: scadigy -f mca0
4fb4-91b8-f9e8-56a7-bc62-ad7d-321c-9846-f47f-2976
```

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.

```bash
root@sign2: scadigy -D
Security Officer Login: ns-dso5
Security Officer Password:  
scadigy(mca0@localhost, ns-dso5)>
```

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: 3613035353131313
Key Fingerprint: 4fb4-91b8-f9e8-56a7-bc62-ad7d-321c-9846-f47f-2976
```

Please select an action:
1. Abort this connection
2. Trust the board for this session only.
3. Trust the board for all future sessions.

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

42. DSO5 deletes existing account DSO2

```bash
scadigy(mca0@localhost, ns-dso5)>
delete so ns-dso2
Delete security officer ns-dso2? (Y/Yes/N/No) [No]: y
Security officer ns-dso2 deleted.
```

43. DSO2 creates its own account
44. DSO5 checks all expected DSOs accounts are created (order may vary)

    scsmgr(mcas01@localhost, ns-dso0) > show so
    Security Officer Multi-Admin Role
    -------------------------------------------------------------
    ns-dso0 Disabled
    ns-dso1 Disabled
    ns-dso2 Disabled
    ns-dso3 Disabled
    ns-dso4 Disabled
    ns-dso5 Disabled
    -------------------------------------------------------------

45. DSO5 logs out from the session

    scsmgr(mcas01@localhost, ns-dso5) > quit

---

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: 192.168.62.16[FULLY_AGREE|REMOTE
    standby_signer: 192.168.58.14[FULLY_AGREE|LOCAL

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 scea000 | head -5
    Listing keys in repository: scea000
    230 keys found.
    Repository ID Type
    5 - scea000 5e2a6f2d9e4e462b8223808db33807fa37 RSA/2048
    scea000 a3ff3806ca3a16e0146ec2f3666a7 RSA/2048
    scea000 4c053b0b1099c4dd096de0b0965638c RSA/2048
    scea000 59bb8222679db9964403232f3b3f89c3b RSA/2048
    scea000 c8a6e3b333c775ed06060346d259 RSA/2048

48. Proceed to delete all expired keys in active policies

    sudo -u opensmsec ods-purge-keys.sh

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 scea000 | head -5
    Listing keys in repository: scea000
    132 keys found.

---

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 opensmsec ods-keygen.sh 24M
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.

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

```
oda-hsmutil list scea6000 | head -5
Listing keys in repository: scea6000
200 keys found.
Repository ID: Type
------------- -----
scea6000 140d239b6e518c11c2b3e3c6a12c3560e155c83e5a5d545e7a5f653f545e1a5edd RSA/2048
scea6000 13b6a7e12219b9a7e6935d05e5e2bcbf RSA/2048
scea6000 9e89394e23b9a588bdc013f0245a0451a5 RSA/2048
scea6000 5ac5e4ed62654329f5e37bc85op024r8 RSA/2048
scea6000 76394a27f61e324ed4964e4b4b59dd53 RSA/2048
```

Backup generation

Estimated time: 10 min

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 RASP database...
SQLite database set to: /var/opendnssec/keyp.db
Back up keystore nz-dnssec-keystore...
You will be prompted for Keystone Security Officer(KSO) credentials. After entering them, the backup will pause while other Keystone Security Officers authorize the backup operation.
Press enter to continue.
```

KSO1 authorizes the backup using their password

```
Keystore = nz-dnssec-keystore.600121.(b12f5f5fa) (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.
```

SA executes the HSM interface in the second window

```
scampr -k nz-dnssec-keystore
Keystore = nz-dnssec-keystore.600121.(b12f5f5fa) (local)
```

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
Authorization successful
```

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

KSO closes the second HSM interface and window

```
scampr> quit
```

The first terminal will show the backup command was authorized and will proceed. Output will look like the following example:
Update: Authenticated security officers: ns-kso1
Update: Authenticated security officers: ns-kso1 ns-kso2
Backup to /tmp/tmp.cghKV5v1862/ns-dsssec-keystore-full-keystore-backup-YYYY-MM-DD successful.

Done backing up keystore ns-dsssec-keystore. The sha256sum of this full keystore backup is

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

58. DSO2 authorizes the device backup with their password

Security Officer Login: ns-dso2
Security Officer Password:

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.cghKV5v1862/device-backup-YYYY-MM-DD successful.

Done backing up HMS device. The sha256sum of this device backup is

Exported keystore Info:
Keystore : ns-dsssec-keystore
Serial # : 605403
Keystore ID : 519920a1
All backups have been exported to /var/lib/dnsee/keystore/key-export-YYYY-MM-DD.tar.gz

Backup of key-backup-YYYY-MM-DD.tar.gz has been created to key-backup-YYYY-MM-DD.tar.gz.sha256sum (sha256sum:

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

Keystore backup file digest

<table>
<thead>
<tr>
<th>CE</th>
<th>D1</th>
<th>26</th>
<th>FF</th>
<th>A5</th>
<th>08</th>
<th>9B</th>
<th>37</th>
</tr>
</thead>
<tbody>
<tr>
<td>4D</td>
<td>39</td>
<td>43</td>
<td>1E</td>
<td>4F</td>
<td>93</td>
<td>F9</td>
<td>C6</td>
</tr>
<tr>
<td>F5</td>
<td>07</td>
<td>39</td>
<td>84</td>
<td>5E</td>
<td>C2</td>
<td>58</td>
<td>C3</td>
</tr>
<tr>
<td>EL</td>
<td>C1</td>
<td>3A</td>
<td>4C</td>
<td>02</td>
<td>3B</td>
<td>03</td>
<td>AC</td>
</tr>
</tbody>
</table>

62. SA closes the root session

root# sign2: exit

63. SA logs outs from the signing box

sysadmin#sign2: exit
Connection to sign2.internal.ara.net.nz closed.

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 #

AAWCATE0GCW9UXPR

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
Manufacturer 1
iProduct 2 USB DISK 2.0
iSerial 3 AAMCATEGCHNVXFR
```  

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

```
scp synadmin@signer:var/lib/dnsec/keygen/key-backup+/ /media/MASTER_COPY/
Enter passphrase for key 'synadmin-sah-key':
key-backup-YYYY-MM-DD.tar.gz: 100% 453KB
key-backup-YYYY-MM-DD.tar.gz.sha256sum: 100% 95
```  

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
```  

Creating Backup Operative Copies

Wellington Operative Backup Copy

*Estimated time: 5 min*

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

```
Wellington FD Serial #
AAJ1RVH6CP2M5WAP
```  

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
Manufacturer 1
iProduct 2 USB DISK 2.0
iSerial 3 AAMCATEGCHNVXFR
```  

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

```
raroc -xW /media/MASTER_COPY/ /media/WELLINGTON/
```

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
```  

76. SA unmounts and unplugs the OBC FD

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

[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

```bash
lsusb -v | grep -c Product | grep -c Manufacturer
Product 2 USB Flash Drive
serial 3 AAKH3QAWOARAS3RP
```

TIME: 11:36

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

```bash
rsync -avw /media/MASTER_COPY/ /media/AUCKLAND
```

TIME: 11:37

91. SA checks the integrity of the backup

```bash
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

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

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
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

Closing steps
Estimated time: 12 min

114. SA finishes script logging

```
root@laptop> exit
```

115. KGA selects Flash Drive labeled **Key Gen Copy** and hands it 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 056c:e01d | grep -c 1
```

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

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

119. SA generates a printable copy of the script

```
exit
cd /media/KEYGEN_COPY
encript -G -U 2 -o script-0[1].ps script-0[1].log
encript -G -U 2 -o script-2[1].ps script-2[1].log
```

120. SA generates sha256 digest for the printable copy of the script from each terminal window. Output should look like this:
121.
KGA records the sha256 digest into the script copy

<table>
<thead>
<tr>
<th>Script 1 Keystore backup file digest</th>
</tr>
</thead>
<tbody>
<tr>
<td>92:F1:4C:C2:3B:1C:57:3F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Script 2 Keystore backup file digest</th>
</tr>
</thead>
<tbody>
<tr>
<td>5C:7C:25:8C:3C:EF:AC:52</td>
</tr>
</tbody>
</table>

122.
SA prints the script

```bash
lpr script-$[date +Ymjd]-1.ps
lpr script-$[date +Ymjd]-2.ps
```

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

```bash
cp script-$[date +Ymjd]-1.ps /media/KEYGEN_LOG
cp script-$[date +Ymjd]-2.ps /media/KEYGEN_LOG
```

124.
SA unmounts KEY_GEN_LOG FD

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

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

```
<table>
<thead>
<tr>
<th>TEB Serial #</th>
</tr>
</thead>
<tbody>
<tr>
<td>3240511</td>
</tr>
</tbody>
</table>
```

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

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.