The Corda Network
Root Key Generation Ceremony

14th September 2018

Table of Contents

Document Information

Version	Date	Author	details
1.0	13th September 2018	Jonathan Sartin	Incorporated final dress rehearsal updates. First release

Distribution List

Name	Organisation	Email	Telephone
Matt Wilgus	Schellman	matt.wilgus@schellman.com	
Scott Zelko	Schellman	scott.zelko@schellman.com	
Dan Paulson	Schellman	Dan.paulson@schellman.com	

Confidentiality

This document remains the property of R3 LLC. Distribution outside the company is limited to those individuals named above. Further distribution or copying by those individuals should only take place with the approval of the Author.

About the ceremony

Introduction

This document is a draft of the Corda Network root key generation script. Its intended audience is R3 internal staff and external service auditors only.

Foundational to the Corda identity model, are the private and public keys and certificates that provide the anchor (or root), from which trust in the identity of entities in a network is derived. In the case of the Corda Network, its root key and certificate are also expected to have a long operational life and will only be rotated in extremis. This longevity, combined with the implicit trust that the network will assume for the root, requires the support of an appropriately rigorous and transparent security control environment.

Part of this control environment is the root key generation ceremony, a detailed, scripted procedure for the generation, signing and activation of the cryptographic materials described above. This procedure will provide reasonable assurance to network participants that the ceremony was carried out according to the Corda Network's detailed key generation procedures and declared business practices. To achieve this, the key generation ceremony must be independently witnessed by internal staff, external service auditors and may also be witnessed by members of the wider community.

The root key generation ceremony will be broken down into a number of sequential phases, or "acts". These acts are as follows:

Act 1 - Initiate Ceremony and retrieve equipment – Briefing the participants, setting up the room and identifying the resources required to complete the ceremony.

Act 2 - Setup equipment – Setting up the equipment in a known secure state so that it may be activated, and cryptographic material generated

Act 3 - Activate HSM and generate Keys and Certificates – generating the root key material in a secure manner.

Act 4 - Secure Hardware and key material – Ensuring that the resulting key material is prepared for storage and the equipment may be put away in a secure state.

Act 5 - Close the Key signing Ceremony – Ceremony closedown and collection of audit material.

Location and timing

The Corda Network root key generation ceremony is intended to take place in R3's London office on 14th September 2018 between 08:00 and 18:00 BST (GMT +1). The ceremony will take place in the Merkle and da Vinci meeting in R3's office at 2 London Wall Place, London, EC2Y 5AU.

Participants

Instructions: At the end of the ceremony, participants sign on IW's copy. IW records time upon completion.

The following is a list of key participants in the ceremony:

Role	Name	Signature	Time arrived	Time departed
CD	Nigel King			
CA	Wawrzek Niewodniczański			
CA2	Thomas Nicholson			
IW	Jonathan Sartin			
IW2	James Carlyle			
AVO	Josie Tenga			
SO	Mike Cave			
SO2	James Brown			
MC1	Przemyslaw Bak			
MC2	Tudor Malene			
SA	Dan Paulson (Schellman)			
EW	Andy Challis (Barclays)			
EW	Christoper Murphy (Ediphy Markets)			
EW	Rob Oddy (BAML)			
EW	Frederic Dalibard (Natixis			
EW	Greg Muecke (Tradewind)			
EW	Jamie Steiner (Guard Time)			

Roles

Senior trusted persons

Senior trusted persons have additional responsibilities with regards ensuring dual occupancy of the room.

The **Ceremony Director (CD)** ensures that participants understand their roles, that the security of the ceremony venue is preserved, and that the ceremony proceeds according to this script

The **Ceremony Administrator** (**CA)** leads the technical aspects of ceremony and is responsible for executing the technical aspects of the script

The **Assistant Ceremony Administrator** (CA2) leads the technical aspects of ceremony and is responsible for executing the technical aspects of the script

The **Internal Witness** (IW) is responsible for signing off each step in the ceremony and will attest to the fact that the key signing ceremony was carried out in accordance with the script and that any deviations from the script were accurately documented.

The **Security Officer** (SO) ensures that physical media is appropriately secured and received and delivered from the correct locations. The SO is also responsible for the integrity of the premises and equipment before, during and after the ceremony.

Other Participants

The **Audio-Visual Operator (AVO)** ensures that AV equipment is functioning throughout the ceremony and that any recording or live streaming of the ceremony is carried out.

The **Media Couriers (MC)** (also known as Secure Storage Controllers) ensure that backup and Master Backup Key material for the subordinate CA and HSM is delivered to the remote storage sites.

The **Service Auditor (SA)** is present to provide independent attestation that the ceremony was conducted in accordance with the script.

External Witnesses (EW) from interested parties may be present to ensure the transparency of the ceremony.

Room Occupancy Rules

- There must be TWO senior trusted persons present at all times.
- Only senior trusted people can enter the room un-escorted.
- An SO is required to escort other participants into or out of the room.
- If the IW needs to temporarily leave the ceremony, proceedings must pause until their return.
- If anybody other than the senior trusted persons are present, the IW must also be present in the room.
- The CA is responsible for enforcing all ceremony practices.

Dual Occupancy of the ceremony room must be enforced. IW with CA or SO must remain inside the Ceremony room if participants are present in that room. CAs, All participants are required to sign in and out of the Ceremony room using the visitor log. The AVO starts filming before the participants enter the Ceremony room.

Emergency evacuation

In the event that the building must be evacuated, the ceremony must be re-started.

Offsite storage locations

Location Name	Abbreviation	Description
Secure Data offsite location 1	SL01	Root cryptographic materials secure location 1
Secure Data offsite location 2	SL02	Root cryptographic materials secure location 2
Secure Data offsite location 3	SL03	Root cryptographic materials secure location 3
R3 safety deposit box location 1	SL04	Subordinate cryptographic materials secure location 1
R3 safety deposit box location 2	SL05	Subordinate cryptographic materials secure location 1
R3 Computer room safe	SL06	Subordinate cryptographic materials secure location 1

Cloud storage

Throughout the script, reference is made to Egnyte, R3's cloud storage solution. Egnyte is used for artefacts of the ceremony, such as logs or certificate material for which availability and integrity requirements are high, but confidentiality low. All items will be stored in R3's Egnyte service at

```
/Egnyte/Shared/Tech & Devs/security/security projects/SEC006 Corda Network
RKG/ceremony-20180914
```

Phonetic Alphabet

Some steps during the ceremony require the participants to tell and/or confirm identifiers composed of numbers and letters. When spelling identifiers, the phonetic alphabet shown below will be used:

A	Alpha	AL-FAH
B	Bravo	BRAH-VOH
C	Charlie	CHAR-LEE
D	Delta	DELL-TAH
E	Echo	ECK-OH
F	Foxtrot	FOKS-TROT
G	Golf	GOLF
H	Hotel	HOH-TEL
I	India	IN-DEE-AH
J	Juliet	JEW-LEE-ETT
K	Kilo	KEY-LOH
L	Lima	LEE-MAH
M	Mike	MIKE
N	November	NO-VEM-ER
O	Oscar	OSS-CAH
P	Papa	PAH-PAH
Q	Quebec	KEH-BECK
R	Romeo	ROW-ME-OH
S	Sierra	SEE-AIR-RAH
T	Tango	TANG-GO
U	Uniform	YOU-NEE-FORM
V	Victor	VIK-TAH
W	Whiskey	WISS-KEY
X	X-ray	ECKS-RAY
Y	Yankee	YANG-KEY
Z	Zulu	ZOO-LOO

Act 1 - Initiate Ceremony and retrieve equipment

Briefing the participants, setting up the room and identifying the resources required to complete the ceremony.

Start cameras

Step	Activity	Initials	Time
1.1	**The AVO starts the cameras recording**		

Briefing for participants

Step	Activity	Initials	Time
1.2	**CD provides briefing for participants** including: • Emergency evacuation • Personal devices / cameras etc • Purposes of Ceremony • Room Occupancy Rules		

Date and Time

Step	Activity	Initials	Time
1.3	**The IW records the Start date and Time in his copy of the script**		

Record of Participants

Step	Activity	Initials	Time
1.4	**IW ensures that record of participants is complete**		

Resources to be used in the ceremony

Step	Activity	Initials	Time
1.5	**The CD may indicate the ceremony resources to the attendees**		

ID	Device	Description
LON18-HSM01	Utimaco Security Server SE Hardware Security Module (HSM)	HSM that will be used to generate the root key for the Corda Network Governing Body (CNGB)
LON18-HSM02	Utimaco Security Server SE Hardware Security Module (HSM)	HSM that will be used to generate the subordinate keys for the Corda Network Operator (CNO)
2 x HSM management laptop (primary and backup)	Lenovo ThinkPad (boxed, sealed)	Laptop to be used to interact with HSM and generate keys and certificates for Ceremony with spare
2 x External DVD drives	Samsung or similar external USB DVD drives (boxed, sealed)	DVD drives to check Operating System DVD hash and to boot HSM management laptop
2 x Operating System DVDs	DVDs of operating system release (nnn) stored in TEB number nnn	OS for HSM management laptop
Secondary Laptop	Apple MacBook pro, Lenovo ThinkPad or similar	Office laptop to validate the Operating System DVD hash, maintain the KeePass password database and to show script progress and certificate details where necessary. This laptop has no specific security provisions.
5 pelicases, labelled for transport	Cases for secure transport and storage of HSM MBKs and cryptographic material.	3 boxes for root keys offsite 2 boxes for intermediate keys offsite

ID	Device	Description
6 x root offsite storage padlocks	padlocks	6 master keyed padlocks for root offsite storage. Green sticker top left.
4 x subordinate offsite storage padlocks	padlocks	4 master keyed padlocks for subordinate offsite storage. Red sticker top left.
ZIP ties	Zip ties with serial numbers	used to ensure peli cases are tamper evident
TEBs	Supply of tamper evident bags for the ceremony	
Card Cases	supply of card cases for the ceremony	Maximum 20 cards needed.
Smartcards x 20	Utimaco Smartcards	6 cards for root HSM MBK, 6 cards for subordinate HSM MBK, 4 cards for admin and oversight users, 4 cards for subordinate signing users. Total 20 cards needed.
USB flash storage - manufacturer 1	USB flash storage for encrypted key backup and certificate export	3 per HSM, 2 for certificate export, 2 spares
USB flash storage - manufacturer 2	USB flash storage for encrypted key backup and certificate export	3 per HSM, 2 for certificate export, 2 spares

Act 2 - Setup Equipment

Operating System DVD Acceptance test

Step	Activity	Initials	Time

2.1	**The Operating System DVD 1 is checked against its published hash value and if the value matches, is accepted for the ceremony.**
	
	 CA2 Commentary: "I will now compute the SHA256 hash for the Operating System DVD and compare it to that which R3 is publishing for the DVD"
	
	 The CA2 removes the two Operating System DVDs from the TEB #nnnnnnn (OS version tcn-rkg-2018-09-13-08.iso) and places them on the table in front of him.
	
	 The CA2 unboxes an external DVD drive (if necessary), connects it to the secondary laptop (windows or mac) and inserts the first of two O/S DVDs into the DVD.
	
	 The CA2 computes the SHA256 hash of the O/S DVD using the following command (or similar):
	
	 `sudo umount /dev/disk2`
	
	 `openssl dgst –sha256 /dev/disk2`
	
	 where /dev/disk2 refers to the raw DVD drive. It the hash does not match, then terminate the ceremony. Otherwise remove the DVD and place it on the table where it may be visible to the camera and participants
	
	 SHA S256 hash for tcn-rkg-2018-09-13-09.iso:
	
	 `22474de672afdf323bf45a681440e7a75b9318e17792c81a31d8d4f9b5bb4181`		

2.2	The Operating System DVD 2 is checked against its published hash value and if the value matches, is accepted for the ceremony. CA2 Commentary: "I will now repeat the process for the second copy of the DVD" The CA2 repeats the preceding step for the second copy of the O/S DVD If neither copy can be accepted, the ceremony terminates		
2.3	The IW records acceptance of the Operating System DVD. The IW records the date, time and signature here upon successful completion: Date: Time: Signature:		

Setup HSM management machine

Step	Activity	Initials	Time
2.4	**The CA prepares the management laptop** CA commentary "We will now unbox the HSM management machine and its backup demonstrate that the box seals are intact" The following steps are carried out in parallel by the CA and CA2 on two separate laptops. 1. The CA demonstrates that the laptop box is sealed 2. The CA un-boxes laptop and places on desk 3. The CA unboxes the external DVD drive for the management laptop and connects it to the management laptop.		
2.5	**The CA boots the management laptop ensuring that unnecessary components are disabled** CA Commentary: "We will now boot and configure the HSM management laptop and it's backup. We will be configuring both together at the same time. I will configure the primary and my colleague will configure the backup. Only the primary will be shown on the screen, We invite the IW and SA to come forward and watch the process" The following steps are carried out in parallel by the CA and CA2 on two separate laptops The CA starts the laptop into BIOS set up Hold F2 at start-up to enter start-up Menu (note, check whether this is F2 or "enter") for the device in use. Press F1 at menu to select BIOS Security > I/O Port Access > <Enter> Wireless LAN > Disable Wireless WAN> Disable Bluetooth > Disable <ESC> > Startup > Boot > <Enter> Set 1 to USB CD HL-DT-ST DVDRAM GP57EW40 Exit > Do you want to save changes and exit now? > Yes The CA starts the laptop into OS and select the only available option from boot menu The following three steps are carried out on the primary HSM management laptop only. After the bash prompt appears, the CA plugs in the HDMI console to start console recording and displaying to the participants. The CA requests that the AVO starts the external laptop recording device. The CA enables logging from the management laptop console using script:		

Step	Activity	Initials	Time

	`mkdir /tmp/audit`
	`script /tmp/audit/tcn-rkg-2018-08-14.log`
	
	The CA unmounts and unplugs the external DVD drive.
	
	    `umount /media`
	
	The CA sets the UK keyboard map
	
	    `loadkeys uk`
	
	The CA compares current time on management laptop with wall clock in the room. The CA displays laptop time 2 command with:
	
	  `hwclock --show`
	
	  `date`
	
	In necessary the CA sets current time on management laptop with time provided by IW:
	
	  `hwclock --set --date "<time provided by IW>" --localtme`
	
	  `(e.g. hwlock --set --date "2018-09-14 08:34:00" --localime)`
	
	  `hwclock --hctosys`
	
	The CA confirms that time has been updated
	
	  `hwclock --show`
	
	  `date`
	
	The CA demonstrates that:
	
	    no persistent storage is configured on the laptop
	
	  `mount`
	
	    expect to show no mounted block device (/dev/sd*)
	
	    that laptop is not networked in any way and to show no wireless is available
	
	  `ip link`
	
	    expect to show no available wireless devices are available
	
	  `ip address`
	
	    expect to show ethernet has no preconfigured networking
	
	    Laptop screen output and logging must be enabled and confirmed.
	
	The CA configures the management laptop network interface
	
	    Set ip address (used device name obtained from ip link above)		

Step	Activity	Initials	Time
	The primary HSM management laptop will be configured as:		
	`ip address add 192.168.4.201/24 dev enp0s31f6 broadcast +`		
	The backup HSM management laptop will be configured as:		
	`ip address add 192.168.4.203/24 dev enp0s31f6 broadcast +`		
	Succeeds silently		
	`ip link set enp0s31f6 up`		
	demonstrate ip address is successfully set		
	`ip address`		
	demonstrate routing table		
	`ip route`		
	The CA runs a script to simplify HSM tools		
	The CA presents a content of the script		
	`cat /tcn/prepare-tools.sh`		
	The CA loads script into current shell session (has to be repated after starting each shell session)		
	`source /tcn/prepare-tools.sh`		
	The CA shows the hosts file to the participants to indicate that "subs" and "root" refer to HSM IP addresses.		

Configure secondary laptop for KeePass database

Step	Activity	Initials	Time
2.6	**CA2 creates the KeePass database on secondary laptop**		
	CA2 commentary: "I am creating a password database for passwords and PINs needed during the ceremony".		
	CA2 opens the KeePassXC program		
	`Database → New database`		
	Save the database as: rkg-passwords.kdbx		
	Change the master key with a Password and Key File		
	SO to provide a memorable password		
	SO writes down the password and stores it in a labelled envelope and keeps it safe		
	CA2 enters the password		
	CA2 sets the key file name to: rkg-passwords.key		

Step	Activity	Initials	Time
	CA2 create two new groups in the KeePass database: pins passwords Save the database		

Create passphrases for backup

Step	Activity	Initials	Time

2.7	**The CA2 creates two complex passphrases for HSM key and user database backup on the secondary laptop using KeePass. They are imported on to the HSM Management laptop ready for later use. The USB device used to transfer them is marked for future destruction.**
	
	The CA requests a USB flash device from the SO
	
	The CA inserts the USB flash drive in the management laptop, formats it.
	
	``lsblk
	mkfs.vfat /dev/sdX```
	
	Where X matches selected device
	
	The CA hands the USB flash drive to the CA 2, who creates and exports the HSM backup passphrases:
	
	    The CA2 creates and records a long passphrase with sufficient entropy for the root key backup using KeePass.
	
	    The CA2 copies this passphrase as **root_backup_pass.txt** on the USB flash drive
	
	    The CA2 creates and records a long passphrase with sufficient entropy for the subordinate key backup using KeePass.
	
	    The CA2 copies this passphrase as **subs_backup_pass.txt** on the USB flash drive.
	
	CA 2 hands the USB devices to CA, who copies the files to to /tmp/gpg/ on the HSM management laptop and formats the USB device when he's done
	
	``mkdir /tmp/gpg
	lsblk
	mount /dev/sdX```
	
	Where X matches selected device
	
	The CA copies the files to HSM management laptop, ensuring that they have arrived in place.
	
	``cp /mnt/*.txt /tmp/gpg/
	ls –al /tmp/gpg```
	
	to confirm the key files are present on the HSM management laptop		

Step	Activity	Initials	Time

	`umount /mnt`
	`lsblk`
	`mkfs.vfat /dev/sdX`
	
	Where X matches selected device
	
	CA provides USB device to SO who sets it aside for destruction at the end of the ceremony.		

Present root HSM and check network settings

Step	Activity	Initials	Time
2.8	**The CA starts the root HSM** CA Commentary "I will now power up the root HSM and check it's IP address" 1. The CA presents the back of root HSM LON18-HSM01 and shows only one network cable 2. The CA turns the root HSM on 3. If needed The CA opens the front hatch to power on 4. The CA confirms HSM is in operational state 5. The CA moves to front of HSM and shows network ethernet 0 configuration for HSM. CSLAN Administration -> Configuration -> Network -> IP Address -> IPv4 -> eth0 Expected output: 192.168.4.203/24		

Present subordinate HSM and check network settings

Step	Activity	Initials	Time

2.9	**The CA starts the subordinate HSM**
	
	CA Commentary "I will now power up the subordinate HSM and check it's IP address"
	
	1. The CA presents the back of subordinate HSM LON18-HSM01 and shows only one network cable
	2. The CA turns root HSM on
	3. if needed The CA opens the front hatch to power on
	4. The CA confirms HSM is in operational state
	5. The CA moves to front of HSM and shows network ethernet 0 configuration for HSM.
	
	`CSLAN Administration -> Configuration -> Network ->`
	`IP Address -> IPv4 -> eth0`
	
	Expected output: 192.168.4.204/24		

Configure the root HSM

Connect the HSM Management machine to the root HSM

Step	Activity	Initials	Time
2.10	The CA connects the HSM management laptop into the root HSM LON18-HSM01 port eth0		

Validate Root HSM state

Step	Activity	Initials	Time
2.11	**The CA validates the initial state of the Root HSM**		
	CA Commentary "I will now validate the state of the root HSM"		
	CA presents CSL Time		
	`csadm Dev=root CSLGetTime`		
	If does not match wall clock time set using (20020602115500 is an example)		
	`csadm Dev=root CSLSetTime=ask,$(date +"%Y%m%d%H%M%S")`		
	CA presents CS component time again (if necessary)		
	`csadm Dev=192.168.4.203 GetTime`		
	Set time to GMT		
	`csadm Dev=root LogonSign=ADMIN,/tcn/hsm/key/ADMIN.key SetTime=GMT`		
	The CA presents current state of audit log		
	`csadm Dev=192.168.4.203 GetAuditConfig`		
	expect a line with "Events=0x00000007"		
	CA to explain what this means		
	`csadm Dev=192.168.4.203 GetAuditLog`		
	Demonstrate new contents		
	The CA presents the Boot Log		
	`csadm Dev=192.168.4.203 GetBootLog`		
	The CA presents current version of firmware and other general information about HSM state		
	`csadm Dev=192.168.4.203 ListFirmware`		
	Expect:		
	ID name type version initialization level		

Step	Activity	Initials	Time
	0 SMOS C64 5.5.9.1 INIT_OK		
	4 POST C64 1.0.0.2 INIT_OK		
	a HCE C64 2.2.2.3 INIT_INACTIVE		
	d EXAR C64 2.2.1.1 INIT_INACTIVE		
	68 CXI C64 2.3.0.5 INIT_OK		
	81 VDES C64 1.0.9.3 INIT_OK		
	82 PP C64 1.3.1.7 INIT_OK		
	83 CMDS C64 3.6.2.0 INIT_OK		
	84 VRSA C64 1.3.6.1 INIT_OK		
	85 SC C64 1.2.0.3 INIT_OK		
	86 UTIL C64 3.0.5.1 INIT_OK		
	87 ADM C64 3.0.25.5 INIT_OK		
	88 DB C64 1.3.2.2 INIT_OK		
	89 HASH C64 1.0.11.2 INIT_OK		
	8b AES C64 1.4.1.4 INIT_OK		
	8d DSA C64 1.2.3.3 INIT_OK		
	8e LNA C64 1.2.4.2 INIT_OK		
	8f ECA C64 1.1.12.4 INIT_OK		
	91 ASN1 C64 1.0.3.6 INIT_OK		
	96 MBK C64 2.2.8.2 INIT_OK		
	9a NTP C64 1.2.0.9 INIT_OK		
	9c ECDSA C64 1.1.16.1 INIT_OK		
	CA confirms Mode is Operational and there is no alarm		
	`csadm Dev=root GetState`		
	CA confirms the version of CSLan OS		
	`csadm Dev =root CSLGetVersion`		
	expect version 4.5.5		
	The CA present the list of users		
	`csadm Dev=root ListUser`		
	expect only default ADMIN user (add action take if not the case)		
	The CA presents the Listed Master Backup Keys – if none present there will be no keys loaded to the HSM		

Step	Activity	Initials	Time
	`csadm Dev=root MBKListKeys` no key expected (add action to take if not the case)		

Install root HSM master backup key

Step	Activity	Initials	Time
2.12	**The SO provides the CA with a set of smartcards to store the root HSM MBK** The CA requests a set of 6 smartcards from SO1. The SO provides the CA with a set of 6 smartcards to contain the sharded root HSM Master Backup key. The SO will label these cards ROOTMBK1 – 6. The CA places the cards in front of him in a position visible to the cameras and the participants.		
2.13	**The CA, with assistance from CA2, prepares the cards by resetting the PIN for the cards** CA Commentary: "I will now prepare six smartcards to store the sharded master backup key for the root HSM " The following steps are repeated six times The CA takes the top card from the deck reading the serial aloud IW records serial of the card CA2 generates a new PIN for the card in KeePassXC on secondary laptop Select the pins group Entries → Add new entry Title: <serial number of KeyCard> Password: Click the Generate 'dice' icon Length: 6 Character Types: 0-9 (digits only) Click OK CA2 will be ready to display the PIN to the CA when requested to do so. **Note that if at anytime the pin pad times out (which it will do during the ceremony), the CA will unplug and reinsert the pin pad USB cable at the HSM management laptop.** Using a card shield The CA changes PIN on the card:		

Step	Activity	Initials	Time
	`csadm MBKPINChange=:cs2:cjo:USB0`		
	inserts card to PIN Pad		
	provides default PIN (123456)		
	Provides and repeats the corresponding PIN from KeePassXC (/pins/<serial>)		
	Card 1 serial:		
	Card 2 serial:		
	Card 3 serial:		
	Card 4 serial:		
	Card 5 serial:		
	Card 6 serial:		
2.14	**The CA creates a new MBK sharded across six smartcards.**		
	CA Commentary: "I will now create the master backup key on the smartcards"		
	The CA creates MBK on 6 cards with 3 required.		
	`csadm Dev=root LogonSign=ADMIN,/tcn/hsm/key/ADMIN.key Key=:cs2:cjo:USB0,15 MBKGenerateKey=AES,32,6,3,tCN-Root`		
	CA2 will provide and PINs from the KeePass for each card serial as required		
	(If Pin Pad goes to sleep, unplug then re-plug)		
2.15	**The CA imports MBK from cards into HSM**		
	CA Commentary: "I will import the MBK from the smartcards into the HSM"		
	The CA imports the MBK using the management laptop		
	`csadm Dev=root LogonSign=ADMIN,/tcn/hsm/key/ADMIN.key Key=:cs2,cjo:USB0,15 MBKImportKey=3`		
	and follows instruction on Pin Pad		
	The CA2 supplies PINs as necessary for the cards used.		

Step	Activity	Initials	Time
	(If Pin Pad goes to sleep, unplug then re-plug)		
	The CA Lists MBK from HSM using the management laptop:		
	`csadm Dev=root MBKListKey`		
	CA and IW record and confirm the key check value		
	Key check value:		
	Date and time:		
	IW Signature		

Prepare root HSM master Backup Key for offsite storage

Step	Activity	Initials	Time
2.16	**The Cards are prepared for offsite storage.**		
	CA Commentary: "will now ask the SO to prepare the MBK smartcards for offsite storage"		
	1. The CA provides the six smartcards to the SO 2. The SO takes two cards and puts them into a card case in a TEB calling out the card serials and bag numbers as he does. 3. The SO closes the TEB 4. The SO puts the TEB into a transport pelicase ready for storage at the root offsite storage location. 5. The IW records the card label, TEB and pelicase box 6. The SO and IW repeat the previous steps for the four remaining cards 7. the SO places the envelopes into separate TEBs, calls out the bag numbers to the IW, who records them and places the bags on the desk in clear view		
	Box 1		
	TEB number:		
	Card 1 serial:		
	Card 2 serial:		

Box 2 TEB number: Card 1 serial: Card 2 serial: Box 3 TEB number: Card 1 serial: Card 2 serial: IW signature:			

Disconnect HSM management laptop from root HSM

Step	Activity	Initials	Time
2.17	**The CA disconnects the HSM Management machine from the root HSM LON18-HSM01 port eth0**		

Configure the subordinate HSM

Connect HSM management machine to subordinate HSM

Step	Activity	Initials	Time
2.18	**The CA connects the HSM Management machine to the subordinate HSM LON18-HSM02 port eth0**		

Validate subordinate HSM state

Step	Activity	Initials	Time
2.19	**The CA validates the initial state of the subordinate HSM**		
	CA Commentary "I will now validate the state of the subordinate HSM"		
	CA presents CSL Time		
	`csadm Dev=subs CSLGetTime`		
	If does not match wall clock time set using (20020602115500 is an example)		
	`csadm Dev=subs CSLSetTime=ask,$(date +"%Y%m%d%H%M%S")`		
	CA presents CS component time again (if necessary)		
	`csadm Dev=subs GetTime`		
	Set time to GMT		
	`csadm Dev=subs LogonSign=ADMIN,/tcn/hsm/key/ADMIN.key SetTime=GMT`		
	The CA presents current state of audit log		
	`csadm Dev=subs GetAuditConfig`		
	expect a line with "Events=0x00000007"		
	CA to explain what this means		
	`csadm Dev=subs GetAuditLog`		
	Demonstrate new contents		
	The CA presents the Boot Log		
	`csadm Dev=subs GetBootLog`		
	The CA presents current version of firmware and other general information about HSM state		
	`csadm Dev=subs ListFirmware`		
	Expect:		
	ID name type version initialization level		
	---		
	0 SMOS C64 5.5.9.1 INIT_OK		
	4 POST C64 1.0.0.2 INIT_OK		
	a HCE C64 2.2.2.3 INIT_INACTIVE		
	d EXAR C64 2.2.1.1 INIT_INACTIVE		
	68 CXI C64 2.3.0.5 INIT_OK		
	81 VDES C64 1.0.9.3 INIT_OK		
	82 PP C64 1.3.1.7 INIT_OK		

Step	Activity	Initials	Time
	83 CMDS C64 3.6.2.0 INIT_OK		
	84 VRSA C64 1.3.6.1 INIT_OK		
	85 SC C64 1.2.0.3 INIT_OK		
	86 UTIL C64 3.0.5.1 INIT_OK		
	87 ADM C64 3.0.25.5 INIT_OK		
	88 DB C64 1.3.2.2 INIT_OK		
	89 HASH C64 1.0.11.2 INIT_OK		
	8b AES C64 1.4.1.4 INIT_OK		
	8d DSA C64 1.2.3.3 INIT_OK		
	8e LNA C64 1.2.4.2 INIT_OK		
	8f ECA C64 1.1.12.4 INIT_OK		
	91 ASN1 C64 1.0.3.6 INIT_OK		
	96 MBK C64 2.2.8.2 INIT_OK		
	9a NTP C64 1.2.0.9 INIT_OK		
	9c ECDSA C64 1.1.16.1 INIT_OK		
	CA confirms Mode is Operational and there is no alarm		
	`csadm Dev=subs GetState`		
	CA confirms the version of CSLan OS		
	`csadm Dev=subs CSLGetVersion`		
	expect version 4.5.5		
	The CA present the list of users		
	`csadm Dev=subs ListUser`		
	expect only default ADMIN user (add action take if not the case)		
	The CA presents the Listed Master Backup Keys – if none present there will be no keys loaded to the HSM		
	`csadm Dev=subs MBKListKeys`		
	no key expected		

Change the subordinate HSM root password

Step	Activity	Initials	Time
2.20	**The CA Changes the subordinate HSM root password**		
	CA Commentary: "I will now change the default root password on the subordinate HSM"		

Step	Activity	Initials	Time
	CA2 generates a new password for the HSM root user using keepass		
	ssh subs -l cslagent su passwd exit		
	Change cslagent password		
	CA2 generates a new password for the HSM root user using keePass		
	`passwd` `exit`		

Replace the subordinate HSM default ADMIN user

Step	Activity	Initials	Time
2.21	**The CA creates a temporary SUPER user**		
	CA commentary: "I will now create a super user on the subordinate HSM to facilitate the creation of additional permanent users and guard against loss of access to the HSM when applying the restricted security config later in the ceremony"		
	The CA requests a new smartcard from the SO		
	The SO provides the smartcard to the CA and calls out the serial. The IW notes the serial in the script.		
	The CA changes the PIN for the smartcard and creates a new user using the card. This user will be destroyed as part of the clean-up process, so the CA is free to use a PIN of his choice. There is no need to record the PIN in KeePass.		
	`csadm ChangePin=:cs2:cjo:USB0`		
	CA enters old PIN 123456		
	CA enters new PIN twice		
	The CA creates the new user:		

	`csadm Dev=subs LogonSign=ADMIN,/tcn/hsm/key/ADMIN.key`
	`AddUser=SUPER,FFFFFFFF{CXI_GROUP=*},rsasign,:cs2,cjo:USB0`
	
	SUPER user smartcard serial:		

2.22	The CA deletes the default ADMIN user using the newly created SUPER user		
	CA Commentary: "I will now remove the default ADMIN user from the subordinate HSM"		

Step	Activity	Initials	Time
	`csadm Dev=subs LogonSign=SUPER,:cs2:cjo:USB0 DeleteUser=ADMIN`		
	CA checks user has gone		
	`csadm Dev=subs ListUsers`		

Install subordinate HSM master backup key

Step	Activity	Initials	Time
2.23	**The SO provides the CA with a set of smartcards to store the subordinate HSM MBK**		
	The CA requests a set of 6 smartcards from SO.		
	The SO provides the CA with a set of 6 smartcards to contain the sharded subordinate HSM Master Backup key. The SO will label these cards SUBSMBK1 – 6.		
	The CA places the cards in front of him in a position visible to the cameras and the participants.		
2.24	**The CA, with assistance from CA2, prepares the cards by resetting the PIN for the cards**		
	CA Commentary: "I will now prepare six smartcards to store the sharded master backup key for the subordinate HSM''		
	The following steps are repeated six times		
	The CA takes the top card from the deck reading the serial aloud		
	IW records serial of the card		
	CA2 generates a new PIN for the card in KeePassXC on secondary laptop		
	Select the pins group		
	Entries → Add new entry		
	Title: <serial number of KeyCard>		
	Password: Click the Generate 'dice' icon		
	Length: 6		
	Character Types: 0-9 (digits only)		
	Click OK		
	CA2 will be ready to display the PIN to the CA when requested to do so.		
	Note that if at any time the pin pad times out (which it will do during the ceremony), the CA will unplug and reinsert the pin pad USB cable at the HSM management laptop.		
	Using a card shield The CA changes PIN on the card:		

Step	Activity	Initials	Time
	`csadm MBKPINChange=:cs2:cjo:USB0`		
	inserts card to PIN Pad		
	provides default PIN (123456)		
	Provides and repeats the corresponding PIN from KeePassXC (/pins/<serial>)		
	Card 1 serial:		
	Card 2 serial:		
	Card 3 serial:		
	Card 4 serial:		
	Card 5 serial:		
	Card 6 serial:		
2.25	**The CA creates a new MBK sharded across six smartcards.**		
	CA Commentary: "I will now create the master backup key on the smartcards"		
	The CA creates MBK on 6 cards with 3 required		
	`csadm Dev=subs LogonSign=SUPER,:cs2:cjo:USB0 Key=:cs2:cjo:USB0,15 MBKGenerateKey=AES,32,6,3,tCN-subs`		
	and follows instruction on Pin Pad		
	CA2 hands cards to CA as required.		
	SUPER card followed by MBK 1-6 as required (If Pin Pad goes to sleep, unplug then re-plug)		
2.26	**The CA imports MBK from cards into HSM**		
	CA Commentary: "I will now import the MBK into the subordinate HSM"		
	The CA imports the MBK using the management laptop		
	`csadm Dev=subs LogonSign=SUPER,:cs2:cjo:USB0 Key=:cs2,cjo:USB0,15 MBKImportKey=3`		
	and follows instruction on PinPad		
	CA2 hands cards to CA as required.		

Step	Activity	Initials	Time
	SUPER card followed by 3 of MBK 1-6 as required (If PinPad goes to sleep, unplug then re-plug)		
	The CA2 supplies PINs as necessary for the cards used.		
	The CA Lists MBK from HSM using the management laptop:		
	`csadm Dev=192.168.4.204 MBKListKey`		
	CA and IW record and confirm the key check value		
	Key check value:		
	Date and time:		
	IW Signature		

Prepare subordinate HSM master Backup Key for offsite storage

Step	Activity	Initials	Time
2.27	**The CA prepares the subordinate HSM MBK smartcards for offsite storage**		
	CA Commentary: "I will now ask the SO to prepare the MBK smartcards for offsite storage"		
	The CA provides the six smartcards to the SO		
	For each location of the cards		
	The SO takes two cards and puts them into a card case in a TEB calling out the card serials and bag numbers as he does.		
	The SO closes the TEB		
	The SO puts the TEB into the appropriate container ready for storage at the subordinate storage location.		
	Location 1		
	TEB number:		
	Card 1 serial:		
	Card 2 serial:		

	Location 2		
	TEB number:		
	Card 1 serial:		
	Card 2 serial:		
	Location 3		
	TEB number:		
	Card 1 serial:		
	Card 2 serial:		

Configure subordinate HSM operational users

Step	Activity	Initials	Time

2.28	**The CA creates operational users for the subordinate HSM**
	
	CA Commentary: "The CA explains why he is going to create HSM Administrator and HSM Oversight (users) on subordinate, but not root HSM. Root is going to be delete, Subordinate is going to be used to create Doorman and NetworkMap certificates soon."
	
	The CA requests 4 smartcards from the SO.
	
	The SO takes 4 cards and labels them HSMAdmin1, HSMAdmin2, HSMOversight1, HSMOversight2.
	
	The SO hands the cards to the CA and calls the labels and serials to the IW as he does so. The IW records the labels and serials in the script.
	
	The CA creates the first HSM Administrator users
	
	    The CA inserts the HSMAdmin1 card into the smartcard reader
	
	    `csadm ChangePin=:cs2:cjo:USB0`
	
	    CA enters old PIN 123456
	
	    CA2 creates a new PIN in the KeePass database		

Step	Activity	Initials	Time
	Title = <serial number> User = HSMAdmin1 Password = 6 digits		
	The CA enters the new PIN		
	The CA creates the new user		
	`csadm Dev=subs LogonSign=SUPER,:cs2:cjo:USB0`		
	`AddUser=HSMAdmin1,22022000{CXI_GROUP=*},rsasign,:cs2,cjo:USB0`		
	The CA creates the second HSM Administrator user		
	The CA inserts the HSMAdmin2 card into the smartcard reader		
	`csadm ChangePin=:cs2:cjo:USB0`		
	CA enters old PIN 123456		
	CA2 creates a new PIN in the KeePass database		
	Title = <serial number> User = HSMAdmin2 Password = 6 digits		
	The CA enters the new PIN		
	The CA creates the new user		
	`csadm Dev=subs LogonSign=SUPER,:cs2:cjo:USB0` `AddUser=HSMAdmin2,22022200{CXI_GROUP=*},rsasign,:cs2,cjo:USB0`		
	The CA creates the first HSM Oversight users		
	The CA inserts the HSMOversight1 card into the smartcard reader		
	`csadm ChangePin=:cs2:cjo:USB0`		
	CA enters old PIN 123456		
	CA2 creates a new PIN in the KeePass database		
	Title = <serial number> User = HSMOversight1 Password = 6 digits		
	The CA enters the new PIN		
	The CA creates the new user		
	`csadm Dev=subs LogonSign=SUPER,:cs2:cjo:USB0` `AddUser=HSMOversight1,22020020{CXI_GROUP=*},rsasign,:cs2,cjo:USB0`		
	The CA creates the second HSM oversight user		
	The CA inserts the HSMOversight2 card into the smartcard reader		
	`csadm ChangePin=:cs2:cjo:USB0`		
	CA enters old PIN 123456		
	CA2 creates a new PIN in the KeePass database		

Step	Activity	Initials	Time

	Title = <serial number>
	User = HSMOversight2
	Password = 6 digits
	
	The CA enters the new PIN
	
	The CA creates the new user
	
	`csadm Dev=subs LogonSign=SUPER,:cs2:cjo:USB0`
	`AddUser=HSMOversight2,22020020{CXI_GROUP=*},rsasign,:cs2,cjo:USB0`
	
	The CA lists users to verify the four operational users have been created
	
	`csadm Dev=subs ListUsers`
	
	HSMAdmin1 Card Serial:
	
	HSMAdmin2 Card Serial:
	
	HSMOversight1 Card Serial:
	
	HSMOversight2 Card Serial:		

Load a restricted security config on the subordinate HSM

Step	Activity	Initials	Time

2.29	**The CA loads a restricted config for the HSM**
	
	CA Commentary: "I will now load an alternate restricted configuration of the HSM that requires on each of two separate groups of users are present to carry out significant operations on the HSM. This enforces a maker-checker capability for the subordinate HSM."
	
	The restricted security config is included in the appendix to this doc and is used to provide capability separation between Admin and Oversight HSM users.
	
	The CA loads an alternate signing module key for the HSM
	
	`csadm Dev=subs LogonSign=SUPER,:cs2:cjo:USB0`
	`LoadAltMdlSigKey=/tcn/hsm/MDL_PUB.key`
	
	The CA uploads the restricted config
	
	`csadm Dev=subs LogonSign=SUPER,:cs2:cjo:USB0`
	`LoadFile=/tcn/hsm/cmds.scf`		

Restart the subordinate HSM

Step	Activity	Initials	Time
2.30	**The CA restarts the Crypto Server to activate the restricted config** CA Commentary: "I will restart the crypto server in order to activate the restricted config on the subordinate HSM" `csadm Dev=subs Restart`		

Verify that the new users have survived the restart

Step	Activity	Initials	Time
2.31	**The CA verifies the users after reboot** CA Commentary: "I will now check users that I have created survived the cryptoserver reboot" `csadm Dev=subs ListUsers`		

Disconnect HSM management laptop from subordinate HSM

Step	Activity	Initials	Time
2.32	The CA disconnects the HSM Management machine from the subordinate HSM LON18-HSM02 port eth0		

Act 3 - Activate HSMs and generate Keys and Certificates

Generate root key and Certificate

Connect HSM management machine to root HSM

Step	Activity	Initials	Time
3.1	The CA connects the HSM Management machine to the root HSM LON18-HSM02 port eth0		

Create commissioner user on the root HSM

Step	Activity	Initials	Time
3.2	**The CA creates a single commissioner user that will be used to manipulate the root key during the ceremony**		
	CA Commentary: "I will now create a single commissioner user to enable manipulation of the root key during the ceremony"		
	The CA connects the keypad to the management laptop if it is not already		
	The CA requests a new blank smartcard from the SO.		
	SO labels the card ROOTCOMM, calls out the label and serial number and provides new card to CA		
	The IW records new card serial number and label in the script		
	The CA changes the PIN for the smartcard and creates a new user using the card. This user will be destroyed as part of the clean-up process, so the CA is free to use a PIN of his choice. There is no need to record the PIN in KeePass.		
	`csadm ChangePin=:cs2:cjo:USB0`		
	CA enters old PIN 123456		
	CA enters new PIN twice (CA2 - might as well write it down, it's going to be deleted)		
	The CA creates COMMISSIONER user with Authentication mechanism: Smartcard (RSA Signature), Permission levels: 0000 0002 and Attributes:		
	`csadm Dev=root LogonSign=ADMIN,/tcn/hsm/key/ADMIN.key AddUser=COMMISSIONER,00000002{CXI_GROUP=TCN.*},rsasign,:cs2:cjo:USB0`		
	The CA inserts COMMISSIONER user card into Pin Pad and follows instructions.		
	Will silently succeed.		
	The CA lists the current users:		
	`csadm Dev=root ListUsers`		
	expect ADMIN and COMMISSIONER only		
	ROOT COMMISSIONER user card serial:		

Step	Activity	Initials	Time
	ROOT COMMISSIONER user card label:		

Generate root key and certificate

Step	Activity	Initials	Time

3.3	**The CA creates the root key and certificate**
	
	CA commentary: "I will now create the root key and certificate"
	
	The CA prepares to generate root key and matching certificate:
	
	    The CA plugs in keypad if it isn't already.
	
	    The CA change current directory to /tcn/sign
	
	    `cd /tcn/sign`
	
	    The CA presents the content of root-hsm-operations.sh script
	
	    `cat root-hsm-operations.sh`
	
	    The CA presents the content of root-key-csr-generator.conf file
	
	    `cat root-key-csr-generator.conf`
	
	The CA generates the root cert
	
	    The CA executes root hsm operations script with option to initialize key and CSR.
	
	    unplug pinpad
	
	    `./root-hsm-operations.sh key-csr-generator`
	
	    hit Enter on management laptop
	
	    On prompt on by laptop, plug card reader in to laptop
	
	    when prompted by pin pad enter COMMISSIONER card and pin details as required
	
	The CA lists the keys present in the HSM
	
	    `cxitool Dev=root`
	    `LogonSign=COMMISSIONER,:cs2:cjo:USB0 ListKeys`
	
	    will display key named 'rootca'
	
	    Expect:
	
	    Algo: ECDSA
	    size: 256
	    type: pub+priv
	    group: TCN:ROOT
	    name: rootca
	    spec: 1
	
	    The CA confirms algorithm and size match expectations		

Step	Activity	Initials	Time

	The CA presents content of config for signing csr:
	
	`cat root-csr-signer.conf`
	
	The CA executes root hsm operations script for signing csr:
	
	    Unplug pinpad
	
	    `./root-hsm-operations.sh csr-signer`
	
	    Plug in PIN pad
	
	    hit Enter on management laptop
	
	    On prompt on by laptop, plug card reader in to laptop
	
	    when prompted by PIN pad, insert the COMMISSIONER card and enter pin details as required
	
	The CA presents content of config for updating certificate:
	
	`cat root-cert-updater.conf`
	
	    The CA executes root hsm operations script with option to initialise key and CSR.
	
	    Unplug pin pad
	
	    `./root-hsm-operations.sh cert-updater`
	
	    hit Enter on management laptop
	
	    On prompt on by laptop, plug card reader in to laptop
	
	    when prompted by pin pad plug in COMMISSIONER card and enter pin details as required		

Create CRL using root HSM

Step	Activity	Initials	Time

3.4	**The CA generates an empty CRL and signs it with the root key**
	
	CA Commentary: "I will now create the initial certificate revocation list"
	
	The CA presents content of config for generating empty CRL:
	
	`cat root-crl-generator.conf`
	
	The CA executes root hsm operations script with option to generate the CRL signed by root key
	
	`./root-hsm-operations.sh crl-generator`
	
	CA shows the generated CRL:
	
	`keytool -printcrl -file cnrc.crl`		

3.5	The CA prepare USB flash drives for CRL. The CA requests two USB flash devices from the SO		

Step	Activity	Initials	Time
	Each step repeated twice		
	Connect a USB stick (this will be destroyed later):		

	Identify device, format and mount USB flash drive:
	
	```lsblk```
	```mkfs.vfat /dev/sdX```
	```mount /dev/sdX /mnt```
	
	       where X matches selected device		

	CA copies CRL file onto USB flash drive:
	
	```cp cnrc.crl /mnt```
	```ls –la /mnt```		

	The CA umount USB disk: ```umount /mnt```		
	CA1 hands USB flash devices to CA2		
	CA2 uses one USB device to copy CRL to Egnyte		
	The CA2 hands both USB drives to SO who places seals them in a TEB, labelled as ROOT CRL and marked for storage in the London office safe.		
	The SO calls out the TEB number to the IW who records it in the script: CRL TEB number:		

Backup root key and verify backup integrity

CA Commentary: "Due to the criticality of the root key and certificate, we are going to validate that the cryptographic material can be backed up and restored to an alternate name in the root HSM"

Backup root key from HSM

Step	Activity	Initials	Time
3.6	**The CA prepare a USB memory stick for the backup of the root key using user COMMISSIONER** CA commentary "I am now going to create an encrypted backup of the root key" The CA requests a USB flash drive from the SO. The CA Connects the USB flash drive to the management laptop: Identify device, format and mount USB flash drive:		

Step	Activity	Initials	Time
| | ```
lsblk
mkfs.vfat /dev/sdX
mount /dev/sdX /mnt
``` <br><br> where X matches selected device <br><br> CA creates the backup: <br><br> ```
cxitool Dev=root LogonSign=COMMISSIONER,:cs2:cjo:USB0
Name=rootca BackupKey
``` <br><br> The CA renames file with root key backup <br><br> ```
mkdir /tmp/keys
mv TCN%2E%ROOT.kbk /tmp/keys/root_key.kbk
``` <br><br> The CA encrypts backup of root key using gpg using the passphrase root backup passphrase created earlier: <br><br> ```
gpg --cipher-algo aes256 --output
/tmp/keys/root_key.kbk.enc --passphrase-file
/tmp/gpg/root_backup_pass.txt --batch --yes --armour
--symmetric /tmp/keys/root_key.kbk
``` <br><br> The CA copies the encrypted backup to the USB flash drive: <br><br> ```
cp /tmp/keys/root_key.kbk.enc /mnt/
``` | | |

Remove backup from HSM

Step	Activity	Initials	Time
| 3.7 | **The CA removes the USB flash drive from the HSM management laptop** <br><br> The CA unmounts the USB drive <br><br> ```
umount /mnt
``` <br><br> The CA removes the USB drive from laptop | | |

Reintroduce backup from HSM

Step	Activity	Initials	Time
| 3.8 | **The CA reconnects and remounts the USB flash drive on the HSM management laptop** <br><br> The CA mounts backup disk to confirm restore procedure <br><br> Reconnect the USB flash device from the previous step <br><br> Identify device, mount USB flash drive: <br><br> ```
lsblk
mount /dev/sdX /mnt
``` <br><br> where X matches selected device | | |

Restore backup to HSM

Step	Activity	Initials	Time
3.9	**The CA restores the root key to the HSM**		
	CA commentary: "I will now test that I can restore the root key to an alternate path"		
	The CA copy key into /tmp directory (temporary filesystem)		
	`mkdir /tmp/keys-bak/` `cp /mnt/root_key.kbk.enc /tmp/keys-bak`		
	The CA decrypts the encrypted backup of root key root backup passphrase created earlier:		
	`gpg --decrypt --cipher-algo aes256 --output /tmp/keys-bak/root_key.kbk --passphrase-file /tmp/gpg/root_backup_pass.txt --batch --yes  /tmp/keys-bak/root_key.kbk.enc`		
	The CA Restores the key:		
	CA runs:		
	`cxitool Dev=root LogonSign=COMMISSIONER,:cs2:cjo:USB0 Name=rootca-bak RestoreKey=/tmp/keys-bak/root_key.kbk`		
	Verify the root key was restored:		
	`cxitool Dev=192.168.4.203 LogonSign=COMMISSIONER,:cs2:cjo:USB0 Name=* ListKeys`		
	There will now appear a rootca-bak key in the key list		

Format test USB flash drive and return to SO for destruction

Step	Activity	Initials	Time
3.10	The CA formats the USB key and returns it to the SO for destruction		
	CA commentary: "we secure the backup test media for destruction, even though it is strongly encrypted"		
	`umount /mnt` `mkfs.vfat /dev/sdX`		
	CA hands test backup key to SO.		
	The SO puts the key in the red "destruction" box.		

Backup the root key and package for offsite storage

Prepare multiple copies of the root key backup for archival on USB flash drives

Step	Activity	Initials	Time
3.11	**The CA prepares three copies of the master backup key encrypted root key for storage**		
	CA Commentary: "I will now copy the encrypted backup of the root key material for transport to secure storage"		
	The SO provides the CA with three USB flash drives for the root backup (from varied manufactures if possible		
	For each backup:		
	The CA the copies encrypted backup to the USB flash device		
	Insert USB flash drive		
	Identify device, format and mount USB flash drive:		
	``` lsblk mkfs.vfat /dev/sdX mount /dev/sdX /mnt ```		
	where X matches selected device		
	Copy root key (encrypted with GPG) copy onto a backup USB flash drive		
	`cp /tmp/keys/root_key.kbk.enc /mnt/`		
	`The CA lists the USB device contents`		
	`ls -la /mnt`		
	The CA unmounts USB flash drive		
	`umount /mnt`		
	The CA remove USB flash drive from laptop		
	The CA sets the USB disk in plain sight of the audience and cameras.		
	The end state will be three USB sticks		

Package the root key USB flash drives for offsite storage

Step	Activity	Initials	Time
3.12	**The Encrypted root keys are prepared for offsite storage** The CA returns the three USB flash drives containing the encrypted root key backup to the SO The SO places each USB drive into a TEB and seals it, then places a single TEB in each root offsite storage box, calling out the TEB and box locations to the IW logs them in the script. Box 1 TEB number: Box 2 TEB number: Box 3 TEB number: IW signature:		

Export the root certificate for distribution

Step	Activity	Initials	Time

3.13	**Export the root certificate and verify that the SHA1 fingerprint of the exported version is the same as that of the certificate on the management laptop.**
	
	CA Commentary: "I will now prepare an export of the root certificate"
	
	The CA exports the root certificate
	
	`cxitool Dev=root LogonSign=COMMISSIONER,:cs2:cjo:USB0 Name=rootca Group=TCN.ROOT Spec=1 ExportCert`
	
	`this will write the certificate to TCN.ROOT_rootca.der`
	
	The CA uses oepnssl to display the certificate		

Step	Activity	Initials	Time
	`openssl x509 –inform der –noout –in TCN.ROOT_rootca.der –text` The certificate is exported ready for distribution. It is now available on the HSM management laptop for future use.		

Disconnect HSM management laptop from root HSM

Step	Activity	Initials	Time
3.14	The CA disconnects the HSM Management machine from the root HSM LON18-HSM01 port eth0		

Generate Subordinate keys and certificates

Connect HSM management machine to subordinate HSM

Step	Activity	Initials	Time
3.15	The CA connects the HSM Management machine to the subordinate HSM LON18-HSM02 port eth0		

Create commissioner user

Step	Activity	Initials	Time

3.16	**The CA creates a single commissioner user that may be used to manipulate the subordinate HSM during the ceremony.**
	
	CA Commentary: "I will now create a commissioner user to enable the manipulation of keys on the subordinate HSM during the ceremony"
	
	The CA connects the keypad to the management laptop if it is not already
	
	The CA requests a new blank smartcard from the SO.
	
	The SO labels the card "SUBCOMM"
	
	The SO calls out the label and the serial number to IW who records them in the script
	
	The SO hands the card to the CA.
	
	The CA changes the PIN for the smartcard and creates a new user using the card. This user will be destroyed as part of the clean-up process, so the CA is free to use a PIN of his choice. There is no need to record the PIN in KeePass.
	
	`csadm ChangePin=:cs2:cjo:USB0`
	
	CA enters old PIN 123456
	
	CA enters new PIN twice (user will be deleted; PIN confidentiality is not critical)
	
	The CA creates COMMISSIONER user.
	
	`csadm Dev=subs LogonSign=HSMAdmin1,:cs2:cjo:USB0 LogonSign=HSMOversight1,:cs2:cjo:USB0 AddUser=COMMISSIONER,00000002{CXI_GROUP=TCN.*},rsasign,:cs2:cjo:USB0`
	
	The CA inserts the HSMAdmin1 and HSMOversiget1 user cards into PinPad and follows instructions.
	
	Will silently succeed.
	
	The CA lists the current users:
	
	`csadm Dev=root ListUsers`
	
	expect SUPER, COMMISSIONER, HSMAdmin1 and HSMOversight1
	
	COMMISSIONER user card serial:		

Generate subordinate keys and CSRs

Step	Activity	Initials	Time
3.17	**Generate subordinate keys and CSRs**		
	CA commentary: "I will now generate subordinate keys and CSRs"		
	The CA prepares to generate subordinate key and matching certificate:		
	The CA plugs in keypad if it isn't already.		
	The CA change current directory to /tcn/sign		
	cd /tcn/sign		
	The CA presents the content of subordinate-hsm-operations.sh script		
	cat subordinate-hsm-operations.sh		
	The CA generates the key and CSR for each subordinate key. The following steps are repeated 8 times (once for each key)		
	unplug pin pad		
	`./subordinate-hsm-operations.sh key-csr-generator N`		
	where "N" is the subordinate key number between 1 and 8, starting at 1 and counting upwards.		
	insert card into pin pad leaving USB disconnected from laptop		
	hit Enter on management laptop		
	On prompt on by laptop, plug card reader in to laptop		
	when prompted by pin pad enter pin details as required		
	The CA uses COMMISSIONER card and PIN to authenticate to HSM.		
	If multiple users are required by the script the CA will invite Crypto Officers to input their details as required. This is defined by the configuration file for the tool		
	The CA validates the keys created		
	`cxitool Dev=subs` `LogonSign=COMMISSIONER,:cs2:cjo:USB0 ListKeys`		
	expecting results similar to:		
	Algo: ECDSA size: 256 type: pub+priv group: TCN:SUBORDINATE name: subordinateN spec: 1		
	For each key generated		

Sign Subordinate CSRs

Disconnect HSM management laptop from subordinate HSM

Step	Activity	Initials	Time
3.18	The CA disconnects the HSM Management machine from the subordinate HSM LON18-HSM02 port eth0		

Connect HSM management machine to root HSM

Step	Activity	Initials	Time
3.19	The CA connects the HSM Management machine to the root HSM LON18-HSM02 port eth0		

Sign subordinate CSRs using the root HSM

Step	Activity	Initials	Time

3.20	**The CA signs the subordinate CSRs and issue certificates using the root key**
	
	CA commentary: "I will now sign the subordinate CSRs and issue certificates using the root key"
	
	The CA presents content of config for signing csr to the room
	
	`cat subordinate-csr-signer.conf`
	
	The CA signes each subordinate CSR to generate root-signed certificates. The following steps are repeated 8 times, once for each key:
	
	`./subordinate-hsm-operations.sh csr-signer <number>`
	
	where <number> is a subordinate key number (between 1 and 10)
	
	insert card into pin pad leaving usb disconnected from laptop
	
	hit Enter on management laptop
	
	On prompt on by laptop, plug card reader in to laptop
	
	when prompted by pin pad enter pin details as required
	
	The CA uses the COMMISSIONER user card and PIN to authenticate to the HSM		

Disconnect HSM management laptop from root HSM

Step	Activity	Initials	Time
3.21	The CA disconnects the HSM Management machine from the root HSM LON18-HSM01 port eth0		

Connect HSM management machine to subordinate HSM

Step	Activity	Initials	Time
3.22	The CA connects the HSM Management machine to the subordinate HSM LON18-HSM02 port eth0		

Write subordinate Certificates to the subordinate HSM

Step	Activity	Initials	Time
3.23	**The CA rewrites the subordinate CA certificates to the subordinate HSM**		
	CA Commentary: I am now writing the subordinate certificates to the HSM		
	The CA presents content of config for updating certificate:		
	`cat subordinate-cert-updater.conf`		
	For each subordinate key the CA executes subordinate-hsm-operations script with option to update CSR		
	`./subordinate-hsm-operations.sh cert-updater <number>`		
	where <number> is a number of a subordinate key		
	insert card into pin pad leaving USB disconnected from laptop		
	hit Enter on management laptop		
	When prompted by the updater select the user you wish to authenticate as to update the HSM with the certificate		
	On prompt on by laptop, plug card reader in to laptop		
	when prompted by pin pad enter pin details as required		
	The CA inserts the COMMISSIONER user card and authenticate with PIN		

Validate certificate path for all subordinate certificates

Step	Activity	Initials	Time
3.24	**The CA will demonstrate that all the certificates produced can be resolved to the root**		
	CA Commentary "I am now going to verify that the exported certificates can be resolved to the trust root"		
	All subordinate certificates are stored in a single Java certificate trust store (JKS). All certificates and their paths can be validated using the KeyTool program		
	CA validates the certificate store		
	`keytool -list -v -keystore certificateStore.jks -storetype jks`		

Step	Activity	Initials	Time
	The CA identifies the Subject Key Identifier for root certificate:		
	Alias name = rootca SubjecKeyIdentifier / KeyIdentifier = < record the hex number here >		
	The IW records the root key identifier in his copy of the script		
	For every Subordinate Certificate, check that the "Authority Key Identifier" is the same as the root key's "Subject Key Identifier"		
	Alias = cna1..x AuthorityKeyIdentifier / KeyIdentifier = < same hex number as recorded in previous step >		
	Root KeyIdentifier		

Export Certificates from the subordinate HSM

The CA will export the finalised certificate store from the HSM management laptop to the secondary laptop.

CA Commentary: "I will now export the final certificates from the HSM management laptop to the secondary laptop.

Step	Activity	Initials	Time
3.25	**The CA creates a PKCS12 format certificate store**		
	CA Commentary: "I will now export the certificate store containing the root and subordinate certificates to PKCS12 format"		
	The keytool will prompt for the certificate store password		
	The CA converts the certificate trust store to PKCS12 format:		
	The keytool will prompt for the certificate store password		
	```keytool –importkeystore –srckeystore certificateStore.jks –destkeystore certificateStore.p12 –srcstoretype JKS – deststoretype PKCS12```		
	The CA lists the certificate store contents with the associated fingerprints		
	```keytool –list –keystore certificateStore.p12 – storetype pkcs12```		
	The IW records the root fingerprints in the script: Root Cert fingerprint		

Step	Activity	Initials	Time
3.26	**The CA will Export the JKS and PKCS12 certificate stores to media that can be shared with the world.**		
	CA Commentary: "I will now copy the certificate stores to external media"		
	The CA requests two USB flash drives of different vendors from SO		
	SO hands USB devices to CA		
	The CA formats and writes the certificateStore to the flash drives		

	Identify device, format and mount USB flash drive:
	
	`lsblk`
	`mkfs.vfat /dev/sdX`
	`mount /dev/sdX /mnt`
	
	where X matches selected device		

	`cp certificateStore.* /mnt/`		
	`cp TCN.ROOT_rootca.der /mnt/`		

	The CA will demonstrate that the media contains the relevant certificate stores only
	
	`ls -al /mnt`
	
	`umount /mnt`		

3.27	**The CA2 will import that certificate on the secondary laptop (linux example)**		

	CA Commentary: "I will now copy the exported certificates to a secondary machine and show their contents"
	
	`lsblk`
	
	`mount /dev/sdX /mnt (where X is from the lsblk step)`
	
	`cp /mnt/certificateStore.jks Egnyte`
	
	`cp /mnt/certificateStore.pk12 Egnyte`
	
	`cp /mnt/TCN.ROOT_rootca.der Egnyte`		

	The CA2 will display the certificate fingerprints.
	
	`keytool -list -keystore <Egnyte path>/certificateStore.jks -storetype jks`
	
	`keytool -list -keystore <Egnyte path>/certificateStore.p12> -storetype pkcs12`		

	Alternate fingerprint display technique (windows) `C:\Program Files (x86)\Java\jre1.8.0_181\bin\keytool.exe -list -keystore "<Egnyte path>/certificateStore.jks"`		

Step	Activity	Initials	Time
	The IW will confirm that the root fingerprint matches that created on the management laptop in the previous step		
3.28	**The CA2 copies the certificates to Egnyte (cloud file storage) from the secondary laptop.** CA Commentary: We will copy the certificate stores to cloud storage		
3.29	**The SO prepares the completed certificate export for storage in the London office Safe** CA Commentary: "We will retain the media used to transfer the certificates for safekeeping" The CA2 returns the USB flash drives to the SO The SO seals the USB flash drives in a TEB, labelled TCN Certificate Backup and calls out the TEB number and label to the IW who records them in this log Certificate backup TEB number: Certificate backup TEB label:		

Cleanup Backup and shutdown subordinate HSM

Delete the SUPER user

Step	Activity	Initials	Time

3.30	**The CA deletes the SUPER user from the subordinate HSM**
	
	CA commentary: "The SUPER user is no longer required, I shall remove it"
	
	`csadm Dev=subs LogonSign=HSMAdmin1,:cs2:cjo:USB0`
	`LogonSign:HSMOversight1,:cs2:cjo:USB0 DeleteUser=SUPER`
	
	The CA hands the SO the SUPER smartcard who marks if for destruction		

Delete the COMMISSIONER user

Step	Activity	Initials	Time

3.31	**The CA deletes the commissioner user from the subordinate HSM**
	
	CA commentary: "The commissioner user is no longer required, I shall remove it"
	
	`csadm Dev=subs LogonSign=HSMAdmin1,:cs2:cjo:USB0`
	`LogonSign:HSMOversight1,:cs2:cjo:USB0`
	`DeleteUser=COMMISSIONER`		

Step	Activity	Initials	Time
	The CA hands the SO the COMMISSIONER smartcard who marks if for destruction		

Backup the subordinate HSM

Step	Activity	Initials	Time
3.32	**The CA backs up the subordinate HSM user database and key files:** CA Commentary: "I will now backup the subordinate HSM user database and key files and copy to the HSM management for export The CA backs up the user and key database from the HSM using the HSMAdmin1 user: ``` mkdir /tmp/backup csadm Dev=subs LogonSign=HSMAdmin1,:cs2:cjo:USB0 BackupDatabase=/tmp/backup/user.db csadm Dev=subs LogonSign=HSMAdmin1,:cs2:cjo:USB0 BackupDatabase=/tmp/backup/CXIKEY.db ``` The CA encrypts both backup using gpg with the subordinate backup passphrase created earlier: the encryption from the previous step ``` gpg --cipher-algo aes256 --output /tmp/backup/CXIKEY.db.enc --passphrase-file /tmp/gpg/subs_backup_pass.txt --batch --yes --armour --symmetric /tmp/backup/CXIKEY.db gpg --cipher-algo aes256 --output /tmp/backup/user.db.enc --passphrase-file /tmp/gpg/subs_backup_pass.txt --batch --yes --armour --symmetric /tmp/backup/user.db ``` The SO provides the CA with three blank flash memory drives (from separate vendors, if possible) The CA writes the encrypted backups to six flash memory sticks: Repeat for each USB flash drive: Identify device, format and mount USB flash drive: ``` lsblk mkfs.vfat /dev/sdX mount /dev/sdX /mnt ``` where X matches selected device ``` cp /tmp/backup/*.enc /mnt/ ``` ``` umount /mnt ```		

Package the subordinate HSM backups USB flash drives for offsite storage

Step	Activity	Initials	Time
3.33	**The subordinate HSM back up USB flash drives are packaged for offsite storage** CA commentary: "we will now package the subordinate HSM backups for external storage" The CA hands the flash memory to the SO The SO seals the three USB flash memory in three separate TEBs and stores them in the appropriate offsite storage location, labelling them as "subordinate HSM backup" and, calling out the TEB and location number to the IW, who writes them down in his copy of the script TEB 1 number: TEB 1 Location: TEB 2 number: TEB 2 location: TEB 3 number: TEB 3 location:		

Backup the KeePass database

The CA2 backs up the KeePass database and provides the backup material to the SO for safekeeping

CA commentary: "I will now backup the keePass database and provide the backups to the SO for safekeeping"

Step	Activity	Initials	Time
3.34	The CA2 requests four USB memory sticks from SO. (one will be used to copy the KeePass database to the secondary laptop and on to Egnyte. three others will be used to backup the keePass secret key)		
3.35	The CA2 copies the KeePass database (rkg-passwords.kbdx) onto one USB stick		

Step	Activity	Initials	Time
	The CA2 copies the database to Egnyte		
3.36	CA2 validates that the Egnyte copy of the database functions as expected.		
	the KeePass database is tested in Egnyte using the rkg-passwords key on one of the USB sticks SO - provides the KeePass database password from the envelope CA2 - When successful, the rkg-passwords.key file is securely deleted from the secondary laptop		
3.37	The KeePass database and keys are stored in TEB ready for storage		
	The SO seals the KeePass database USB flash drive in a TEB, labelled TCN KeePass database and calls out the label and bag number to the IW, who records it in the script		
	The SO seals the KeePass keys USB flash drives in separate TEBs, labelled TCN KeePass key 1, TCN KeePass Key 2, and TCN KeePass key 3 and calls out the label and bag number to the IW, who records them in the script.		
	The SO sets the four bags aside for storage in the London office safe. The keys will be lodged with R3 directors in the near future		
	KeePass database TEB number:		
	KeePass key 1 TEB number:		
	KeePass key 2 TEB number:		
	KeePass key 3 TEB number:		

Export the audit logs from the subordinate HSM

Step	Activity	Initials	Time

3.38	**The CA exports the logs from the subordinate HSM LON18-HSM02 to the HSM management laptop and takes a SHA256 hash of the result.**
	
	CA Commentary: "I am now going to take a copy of the subordinate HSM audit log and copy to the HSM management laptop
	
	The Ca create a temporary directory to store logs:
	
	`mkdir /tmp/audit`
	
	The CA carries out a process to export the subordinate HSM audit and cslan logs
	
	`cd /tcn/hsm`
	
	`csadm Dev=subs LogonSign:HSMAdmin1,:cs2:cjo:USB0 GetAuditLog > /tmp/audit/hsm02_audit.log`
	
	`csadm Dev=sub LogonSign:HSMAdmin1,:cs2:cjo:USB0 CSLGetLogFile > /tmp/audit/hsm02_cslan.log`
	
	The CA creates a SHA256 hash of the subordinate HSM audit logs and the IW records it in the script below
	
	`cd /tmp/audit`
	`openssl dgst -sha256 hsm02_audit.log > hsm02_audit.log.sha256`
	
	`openssl dgst -sha256 hsm02_cslan.log > hsm02_cslan.log.sha256`		

Shutdown the subordinate HSM

Step	Activity	Initials	Time
3.39	**The CA shuts the subordinate HSM down** CA Commentary "I will now shutdown the subordinate HSM" `csadm Dev=subs CSLShutdown=ask`		

Disconnect HSM management laptop from subordinate HSM

Step	Activity	Initials	Time
3.40	The CA disconnects the HSM Management machine from the subordinate HSM LON18-HSM02 port eth0		

Cleanup, purge and shutdown root HSM

Connect HSM management machine to root HSM

Step	Activity	Initials	Time
3.41	The CA connects the HSM Management machine to the root HSM LON18-HSM02 port eth0		

Export the audit logs from the root HSM

Step	Activity	Initials	Time

3.42	**The CA exports the logs from the root HSM LON18-HSM01 to the HSM management laptop and takes a SHA256 hash of the result.** CA Commentary: "I am now going to take a copy of the root HSM audit log and copy to the HSM management laptop The CA carries out a process to export the root HSM audit and cslan logs
	
	`cd /tcn/hsm`
	
	`csadm Dev=root LogonSign=ADMIN,/tcn/hsm/key/ADMIN.key GetAuditLog > /tmp/audit/hsm01_audit.log`
	
	`csadm Dev=root LogonSign=ADMIN,/tcn/hsm/key/ADMIN.key CSLGetLogFile > /tmp/audit/hsm01_cslan.log`
	
	The CA creates a SHA256 hash of the subordinate HSM audit logs and the IW records it in the script below
	
	`cd /tmp/audit/`
	`openssl dgst -sha256 hsm01_audit.log > hsm01_audit.log.sha256`
	
	`openssl dgst -sha256 hsm01_cslan.log > hsm01_cslan.log.sha256`		

Purge the root HSM

Step	Activity	Initials	Time
3.43	**The CA Purges the Crypto Server via an External Erase** CA Commentary: "I am now purging the HSM crypto server by carrying out an external erase" The CA opens front panel door of the HSM and presses the ERASE CS button The CA uses csadm to reset the crypto server to factory default `csadm Dev=root Clear=Defaults`		

Step	Activity	Initials	Time

	The CA resets the Alarm using csadmin and then restarts the Crypto server
	
	`        csadm Dev=root`
	`        LogonSign=Admin,/tcn/hsm/key/ADMIN.key ResetAlarm`		

3.44	**The CA loads firmware modules in order to bring the HSM to an operational state again**
	
	CA Commentary: "I am now Loading the Firmware Modules for CryptoServer to bring the device to Operational Mode"
	
	`csadm Dev=root LogonSign=Admin,/tcn/hsm/key/ADMIN.key`
	`LoadPKG=/tcn/hsm/Firmware/SecurityServer-Se2-`
	`Series/SecurityServer-Se2-Series-4.21.0.3.mpkg`
	
	The CA waits until the HSM state returns to Operational on the front panel		

3.45	**The CA verifies the result of the external erase** CA Commentary "I am now verifying the external erase. The only existing user should be the default admin" The CA present the list of users `        csadm Dev=192.168.4.203 ListUser` expect only default ADMIN user, if not re-purge		

Shutdown the root HSM

Step	Activity	Initials	Time
3.46	**The CA shuts the root HSM down** CA commentary: "I am now shutting the root HSM down" `csadm Dev=root CSLShutdown=ask`		

Disconnect HSM management laptop from root HSM

Step	Activity	Initials	Time
3.47	The CA disconnects the HSM Management machine from the root HSM LON18-HSM01 port eth0		

Act 4- Secure Key, audit and hardware material

Subordinate key material is made ready for transport to secure storage

Step	Activity	Initials	Time
4.1	**The SO ensures that subordinate key material is packaged in an appropriate manner and prepared for transport to its secure storage location.** CD Commentary: The SO will now ensure that the subordinate key material is packaged in an appropriate manner and prepared for transport to our secure storage locations. 2 locked pelicases containing tamper evident bags will be carried by 2 media couriers to 2 external secure locations where the tamper evident bags will be stored in safety deposit boxes. Additionally, one tamper evident bag will be placed in the r3 safe inside the secure server room. Each tamper evident bag will contain 2 identical usb sticks containing a backup of the subordinate key, as well as 2 sharded HSM cards which can be used to access the subordinate HSM on a future occasion. In order to access the HSM on a future occasion, one will require 3 of the 6 HSM cards. The SO ensures that the subordinate key material is stored in the following manner (this may have already been accomplished) SO splits the 6 HSM cards with the subordinate MBK backups into 3 pairs of 2. SO split the 6 USB sticks with the subordinate HSM backup into 3 pairs of 2. SO seals each pair of HSM cards and USB sticks into a TEB SO records all serial numbers of HSM cards, usb sticks, TEBs and zip ties in the asset register, along with the destination they are intended for. SO places 2 secured TEB in the pelicase labelled with the destination SL04. SO places 2 secured TEB in the pelicase labelled with the destination SL05. SO places London Office back ups on the table ready for collection by business services for storage in the safe at the end of the ceremony. 2 pelicases are place on the table ready for collection by media couriers in section 4.6.		

Root Key material is made ready for transport to secure storage

Step	Activity	Initials	Time
4.2	**The SO ensures that root key material is packaged in an appropriate manner and prepared for transport to its secure storage location.**		
	CD Commentary: The SO will now ensure that the root key material is packaged in an appropriate manner and prepared for transport to its secure storage location. 3 pelicases will be collected by a secure data courier and each will be distributed to a different fire proof and secure location specialising in secure data storage. Each pelicase will contain 2 identical USB sticks containing a backup of the root key, as well as 2 sharded HSM cards which can be used to recreate the root HSM on a future occasion. In order to recreate the root HSM on a future occasion, one will require 3 of the 6 HSM cards.		
	The SO ensures that		
	The six smart cards with the root MBK backups are split into 3 pairs of 2 split across storage locations and secured in TEBs.		
	The three USB backups are split across storage locations and secured in TEBs		
	The secured TEBs are placed into the appropriate pelicase.		
	The pelicases are locked.		
	SO places pelicases to the side, which will only leave the room when the courier arrives to collect them.		

Transport root key material to secure storage

Step	Activity	Initials	Time
4.4	**The root key pelicases will be collected by the secure data courier when they arrive (currently a 4pm-8pm window). Following the ceremony, they will be transferred to the server room and locked away waiting for the secure data couriers.**		
	Explain how they are protected in the meantime		

Transport subordinate key material to secure storage

Step	Activity	Initials	Time
4.5	**The SO arranges for the secure storage controllers to transport the subordinate key material to the safety deposit and to the London Office safe.**		
	CD Commentary: There are 2 media couriers who will each take one pelicase and one key, with the key being for the pelicase carried by the		

Step	Activity	Initials	Time
	other media courier. They will deliver the contents of each pelicase to a different secure storage location.		
	The SO calls the Media Couriers into the room.		
	SO calls to doorman to fetch Media Couriers.		
	In the presence of the Media Couriers SO will lock the 2 pelicases according to the colour coded stickers with one lock on the left-hand slot and a zip tie on the right hand slot. The media couriers will witness the securing of the contents they will be delivering.		
	Media Courier 1 collects the pelicase marked SL04 and the key marked SL05.		
	Media Courier 2 collects the pelicase marked SL05 and the key marked SL04.		
	The Media Couriers leave by taxi to the secure locations SL04 and SL05. where they will remove the TEB from each box and place the TEB in to the safety deposit boxes.		

Export logs

Step	Activity	Initials	Time
4.6	**The CA exports logs where necessary.**		
	CA Commentary: "I will now export and print logs where necessary for inclusion in the audit bundle"		
	The SO provides the CA with two USB flash drives for the audit logs.		
	Identify device, format and mount USB flash drive:		
	``` lsblk mkfs.vfat /dev/sdX mount /dev/sdX /mnt ```		
	where X matches selected device.		
	``` ls –lA /mnt umount /mnt ```		
	The CA collects the audit logs and hashes together and copies them to a temporary directory before copying them to the USB flash drives:		
	The CA stops the console log on the HSM management laptop.		
	`exit`		
	The CA carries out a SHA 256 hash of the console log		
	`openssl dgst –sha256 /tmp/audit/tcn–rkg-2018-09-14.log > /tmp/audit/tcn–rkg-2018-09-14.log.sha256`		

Step	Activity	Initials	Time
	The CA calls the hash to the IW, who records it in the script		
	The CA copies the console log to the external storage		
	`cp /root/tcn-rkg-2018-09-14.log /tmp/audit/`		
	The CA exports the console log hash to external storage		
	`cp root/tcn-rkg-2018-09-14.log.sha256 /tmp/audit/`		
	For each USB flash drive:		
	The CA mounts the flash drive:		

	`lsblk`
	        `mkfs.vfat /dev/sdX`
	        `mount /dev/sdX /mnt`		

	where X is the device shown by lsblk		
	The CA copies the contents of the temporary audit directory		
	`cp /tmp/audit/* /mnt`		
	The CA demonstrates that the external storage contains only the audit logs specified		
	First that require and only required files are on the USB flash drive and that none has unexpected size.		
	`ls -l /mnt`		
	expected results is:		
	hsm01_auditlog.log		
	hsm01_auditlog.log.sha256		
	hsm01_cslanlog.log		
	hsm01_cslanlog.log.sha256		
	hsm02_auditlog.log		
	hsm02_auditlog.log.sha256		
	hsm02_cslanlog.log		
	hsm02_cslanlog.log.sha256		
	tcn-rkg-2018-09-14.log		
	tcn-rkg-2018-09-14.log.sha256		
	The CA ejects the external storage from the HSM management laptop		
	`umount /mnt`		

Step	Activity	Initials	Time
	The CA2 imports the audit logs and preserves them for safekeeping		
	The CA2 imports the audit logs and the console logs onto the secondary laptop		
	The CA2 demonstrates that the log hashes match the imported files		
	The CA2 uploads the logs to cloud storage (Egnyte) for review		
	The CA2 supplies the two audit log flash drives to the SO		
	The SO seals the audit flash drives in a TEB and prepares to store it in the London office safe		
	The IW records the TEB number in the script		
	Console log hash:		
	Audit flash drive TEB #:		

Shutdown HSM management laptop

Step	Activity	Initials	Time
4.7	**The CA shuts the HSM management laptop down**		

Secure hardware scheduled for destruction

Step	Activity	Initials	Time
4.8	**The SO seals all USB flash drives and other hardware scheduled for destruction in a TEB**		
	CD Commentary: "The SO will secure any media or other hardware scheduled for destruction. A certificate of destruction will be added to the audit bundles after completion"		
	Hardware scheduled for destruction is sealed in a TEB by the SO.		
	This must include the backup encryption passphrase USB flash drive.		
	The SO will acquire a certificate of destruction for this equipment when the ceremony is complete and the IW will ensure that it is included in the subsequent audit bundle.		

Step	Activity	Initials	Time
	SO calls out the TEB number to the IW who records it in the log		
	Destruction TEB number:		

Secure ceremony hardware

Step	Activity	Initials	Time
4.9	**SO secures the hardware and other material used in the ceremony to the computer room rack or office safe.**		
	CD Commentary: "We will now secure the equipment and media resulting from the ceremony"		
	The following equipment is stored in the London office computer room rack:		
	Management Laptop		
	HSM1		
	HSM2		
	The following equipment is stored in the London office Safe:		
	TEB containing audit log USB flash drive.		
	TEB containing 2 certificate backup		
	2x TEB containing KeePass keys		
	TEB containing KeePass database		
	TEB containing 2 subordinate HSM backup USB flash drives		
	TEB containing 2 subordinate HSM MBK shard smartcards		
	TEB containing hardware scheduled for destruction		

Act 5 – Close the Key signing ceremony

Participants sign internal witness' script

Step	Activity	Initials	Time
5.1	The Participants sign the IW's script		

Stop recording

Step	Activity	Initials	Time
5.2	The AVO stops recording		

Prepare audit materials

Step	Activity	Initials	Time
5.3	The IW prepares audit material according to Appendix A. Audit bundle checklist. Material for the audit bundle available during the ceremony will be compiled during the ceremony. Some material, such as AV footage and scanned copies of the IW's annotated script may be provided after the ceremony has completed. A subsequent attestation will be provided by the IW with regards the accuracy of any material assembled after the ceremony close.		

IW signs Appendix B attestation

Step	Activity	Initials	Time
5.4	The IW signs and dates the Appendix B attestation in his copy of the script		

Post ceremony information and close

Step	Activity	Initials	Time
5.5	The CD brings the ceremony to close and the participants depart. If any secure material remains in the room, dual occupancy applies, and the IW is required to remain until it is collected by the MC.		

Appendix A. Audit bundle checklist

The IW, with the assistance of the SO1 or SO2 will assemble the audit bundle during the ceremony. The bundle components will be duplicated to provide two physical copies and electronic copies stored to Egnyte.

Available during the ceremony

Ceremony Script

Hard copies of the IW's ceremony script, including notes taken during the ceremony and attestation. The SO1 or SO2 will assist the IW in ensuring will ensure that the script is scanned, and a second hard copy of preserved and electronic copies made and uploaded to Egnyte.

System logs from the ceremony

Two USB flash drives containing the audit and cslan logs from the HSMs and the script output from the management laptop (prepared in previous step):

hsm01_auditlog.log

hsm01_auditlog.log.sha256

hsm01_cslanlog.log

hsm01_cslanlog.log.sha256

hsm02_auditlog.log

hsm02_auditlog.log.sha256

hsm02_cslanlog.log

hsm02_cslanlog.log.sha256

tcn-rkg-2018-09-14.log

tcn-rkg-2018-09-14.log.sha256

These will be placed in separate TEBs with the numbers recorded in the IW's script. The audit materials will also be uploaded by the IW to Egnyte.

Assembled after the ceremony

Video Capture device recording from admin laptop

Two sets of the video capture device output from the HSM management laptop from the ceremony, provided by the AVO.

Camera recordings from the ceremony venue

Two sets of the camera recordings from the ceremony venue, provided by the AVO.

Appendix B. Key Ceremony Script (by IW)

I hereby attest that the Root Key Generation Ceremony was conducted in accordance with this script. Any exceptions that may have occurred were accurately and properly documented.

IW:

Signature:

Date:

Appendix C: Script Variation

If it becomes necessary to modify the script during the ceremony, and it is not possible to simply correct inline, the IW will record such changes in additional steps in pages such as this and attach them to his copy of the script. The script will be clearly annotated to describe which step this occurs in:

Additional Step	Activity	Initials	Time

Appendix D. HSM Management Machine recovery

Should the HSM management machine suffer a hardware failure in any way, it will be replaced by the standby machine and the ceremony will proceed as before. The following are the steps necessary to migrate the ceremony to the secondary management machine:

Recovery Step	Activity	Initials	Time
	The CA will copy the following files from the primary HSM management machine via USB drive: ``` lsblk mkfs.vfat /dev/sdX mnt /dev/sdX /mnt mkdir /mnt/recovery cp -r /tmp /mnt/recovery/ cp -r /tcn/sign /mnt/recovery/ umount /mnt ``` Where X matches selected device Remove the USB drive and connect to the backup management laptop: ``` lsblk mnt /dev/sdX /mnt cp -r /mnt/recovery/tmp /tmp/ mv /tcn/sign /tcn/sign.bak cp -r /mnt/recovery/tcn/sign /tcn/sign/ umount /mnt ``` Continue from failed step		

Appendix E. Subordinate HSM restricted config

```
# Version 2.0
# Remove permissions 0x068:8 'BackupKey' and 0x65:9 'RestoreKey' permission
locks as it breaks the Java key api and require ADMIN cards for normal crypto
operations.
#
# Version 1.0
# Original two admin version of config

[DisableSFC]

[Permissions]
# ADM Push up admin level to 4 except for backup, which needs a new backup
group too
0x087 = 3:04002020,\
        4:04002020,\
        7:04002020,\
        11:04002020,\
        19:04002020,\
        20:04002020,\
        21:04002020,\
        22:04002020,\
        23:22020000,\
        24:22020000,\
        25:44002020,\
        26:44002020

#CMDS Push up admin level to 4 except for backup, which needs a new backup
group too
0x083 = 3:40002020,\
        5:40002020,\
        6:40002020,\
        12:20020000,\
        13:40002020,\
        14:40002020,\
        17:40002020,\
        19:04002020,\
        20:04002020

#CXI Push up admin level to 4
0x068 = 16:22020000,\
        17:22020000,\
        33:40002020,\
        34:40002020

#MBK Push up admin to level 4
0x096 = 4:04002020,\
        5:04002020



# Summary of command codes below
```

```
#---------------------------------------------------------------------------
-----------------------
# ADM (FC = 0x087)
#---------------------------------------------------------------------------
-----------------------
# List of external functions (denoted by SFC and function name) in the ADM
module that can be disabled

#IMPORTANT NOTES:
# - Functions denoted by !!! should not be disabled in order to keep essen-
tial functionality available.
# - EXCEPTION (related to SFC = 3):
#   To load the Audit Log configuration file (auditlog.cfg) or an Alternative
Module Signature Key (mdlsigalt.key),
#   user permission 22000000 is required.

# SFC = 0    Echo                                    (default) required
permissions = 00000000
# SFC = 1    !!! GetState !!!                        (default) required
permissions = 00000000
# SFC = 2    ListFiles                               (default) required
permissions = 00000000
# SFC = 3    LoadFile                                (default) required
permissions = 02000000
# SFC = 4    DeleteFile                              (default) required
permissions = 02000000
# SFC = 5    ListModulesActive                       (default) required
permissions = 00000000
# SFC = 6    !!! GetTime !!!                         (default) required
permissions = 00000000
# SFC = 7    SetTime                                 (default) required
permissions = 02000000
# SFC = 10   !!! GetAuditLog !!!                     (default) required
permissions = 00000000
# SFC = 11   !!! ClearAuditLog !!!                   (default) required
permissions = 02000000 or 20000000
# SFC = 14   MemInfo                                 (default) required
permissions = 00000000
# SFC = 18   CheckBootCode                           (default) required
permissions = 00000000
# SFC = 19   LoadFirmwareDecryptionKey               (default) required
permissions = 02000000
# SFC = 20   !!! ResetAlarm !!!                      (default) required
permissions = 02000000 or 20000000
# SFC = 21   Clear                                   (default) required
permissions = 02000000
# SFC = 22   SetTimeRel                              (default) required
permissions = 02000000
# SFC = 23   ListDBSearchKeys                        (default) required
permissions = 22000000
# SFC = 24   ExportDBEntry                           (default) required
permissions = 22000000
# SFC = 25   ImportDBEntry                           (default) required
permissions = 22000000
```

```
# SFC = 26   ConfigParamSet                               (default) required
permissions = 22000000
# SFC = 27   ConfigParamGet                               (default) required
permissions = 00000000
# SFC = 28   RamMemInfo                                   (default) required
permissions = 00000000


#------------------------------------------------------------------------------
---------------------
# CMDS (FC = 0x083)
#------------------------------------------------------------------------------
---------------------
# List of external functions (denoted by SFC and function name) in the CMDS
module that can be disabled

# IMPORTANT NOTES:
# Functions denoted by !!! should not be disabled in order to keep essential
functionality available.
# *   To create a user (SFC = 14) with permission > 0 in user group 6 the
permissions
#     required by default = 21000000

# SFC = 0    Echo                                         (default) required
permissions = 00000000
# SFC = 1    Reverse Echo                                 (default) required
permissions = 00000000
# SFC = 2    List Registered Functions                    (default) required
permissions = 00000000
# SFC = 3    Add User                                     (default) required
permissions = 20000000 *
# SFC = 4    List All Users                               (default) required
permissions = 00000000
# SFC = 5    Delete User                                  (default) required
permissions = 20000000
# SFC = 6    Change User                                  (default) required
permissions = 20000000
# SFC = 7    Get Authentication State                     (default) required
permissions = 00000000
# SFC = 8    Get Boot Log                                 (default) required
permissions = 00000000
# SFC = 9    List PIN Pad Applications                    (default) required
permissions = 00000000
# SFC = 10   !!! Get Session Key !!!                      (default) required
permissions = 00000000
# SFC = 11   !!! End Session !!!                          (default) required
permissions = 00000000
# SFC = 12   Backup User                                  (default) required
permissions = 20000000
# SFC = 13   Restore User                                 (default) required
permissions = 20000000
# SFC = 14   Add User Extended                            (default) required
permissions = 20000000 *
# SFC = 15   !!! Get User Info !!!                        (default) required
permissions = 00000000
```

```
# SFC = 17  Set Maximum Authentication Failures       (default) required
permissions = 20000000
# SFC = 18  Get Maximum Authentication Failures       (default) required
permissions = 00000000
# SFC = 19  Set Administration-Only Mode              (default) required
permissions = 02000000
# SFC = 20  Set Startup Mode                          (default) required
permissions = 02000000
# SFC = 21  Get Startup Mode                          (default) required
permissions = 00000000


#---------------------------------------------------------------------------
-----------------------
# CXI (FC = 0x068)
#---------------------------------------------------------------------------
-----------------------
# List of external functions (denoted by SFC and function name) in the CXI
module that can be disabled

# IMPORTANT NOTES:
# - Functions denoted by !!! should not be disabled in order to keep essen-
tial functionality available.
# - The key management functions, denoted below by ***, require different
permissions depending on the
#   storage object/the object to be returned (see chapter 3.10.4 of csadm
Manual for System Administrators)
# - The user management functions, denoted below by *'*, require different
permissions:
#   20000000 on creation/deletion of an SO or 00000200 on creation/deletion
of a User

# SFC = 0   VerifyGenuineness                         (default) required
permissions = 00000000
# SFC = 1   GetInfo                                   (default) required
permissions = 00000000
# SFC = 2   GetPersKey                                (default) required
permissions = 00000000
# SFC = 5   InitKeyGroup                              (default) required
permissions = 00000200
# SFC = 7   GenerateDSAParam                          (default) required
permissions = 00000002
# SFC = 8   BackupKey ***                             (default) required
permissions = 00000002 or 00000200 or 20000000
# SFC = 9   RestoreKey ***                            (default) required
permissions = 00000002 or 00000200 or 20000000
# SFC = 10  !!! ListKeys *** !!!                      (default) required
permissions = 00000002 or 00000200 or 20000000
# SFC = 11  !!! GenerateKey !!!                       (default) required
permissions = 00000002
# SFC = 12  OpenKey ***                               (default) required
permissions = 00000002 or 00000200 or 20000000
# SFC = 13  DeleteKey ***                             (default) required
permissions = 00000002 or 00000200 or 20000000
# SFC = 14  GetKeyProp ***                            (default) required
permissions = 00000002 or 00000200 or 20000000
```

```
# SFC = 15  !!! SetKeyProp ***!!!                          (default) required
permissions = 00000002 or 00000200 or 20000000
# SFC = 16  ExportKey                                      (default) required
permissions = 00000002
# SFC = 17  ImportKey                                      (default) required
permissions = 00000002
# SFC = 18  ComputeHash                                    (default) required
permissions = 00000002
# SFC = 20  Crypt                                          (default) required
permissions = 00000002
# SFC = 21  Sign                                           (default) required
permissions = 00000002
# SFC = 22  Verify                                         (default) required
permissions = 00000002
# SFC = 23  GenerateRandom                                 (default) required
permissions = 00000002
# SFC = 24  AgreeSecret                                    (default) required
permissions = 00000002
# SFC = 25  CreateObject      (for PKCS#11 support)        (default) required
permissions = 00000002
# SFC = 26  GenerateKeyPair   (for PKCS#11 support)        (default) required
permissions = 00000002
# SFC = 27  CopyObject        (for PKCS#11 support)        (default) required
permissions = 00000002
# SFC = 28  DeriveKey         (for PKCS#11 support)        (default) required
permissions = 00000002
# SFC = 29  WrapKey           (for PKCS#11 support)        (default) required
permissions = 00000002
# SFC = 30  UnwrapKey         (for PKCS#11 support)        (default) required
permissions = 00000002
# SFC = 33  AddUser *'*       (for PKCS#11 support)        (default) required
permissions = 20000000 or 00000200
# SFC = 34  DeleteUser *'*    (for PKCS#11 support)        (default) required
permissions = 20000000 or 00000200
# SFC = 35  GenerateDSAParam PQ                            (default) required
permissions = 00000002
# SFC = 36  GenerateDSAParam G                             (default) required
permissions = 00000002

#----------------------------------------------------------------------------
-----------------------
# MBK (FC = 0x096)
#----------------------------------------------------------------------------
-----------------------
# List of SFCs and the corresponding external functions in the MBK module
that can be disabled
# IMPORTANT: Functions denoted by !!! should not be disabled in order to keep
essential functionality available.

# SFC = 4   !!! Generate MBK !!!                           (default) required
permissions = 02000000
# SFC = 5   !!! Import MBK !!!                             (default) required
permissions = 02000000
# SFC = 6   List Keys                                      (default) required
permissions = 00000000
```

```
#-------------------------------------------------------------------------------
-----------------------
# NTP (FC = 0x09a)
#-------------------------------------------------------------------------------
------------------------
# List of external functions (denoted by SFC and function name) in the NTP
module that can be disabled

# SFC = 1    Set Time Delay                               (default) required
permissions = 00000000
# SFC = 2    Change Activation State                      (default) required
permissions = 00200000
# SFC = 3    Get Settings                                 (default) required
permissions = 00000000
# SFC = 4    Set Settings                                 (default) required
permissions = 00200000
# SFC = 5    Set Time                                     (default) required
permissions = 00200000


#--------------------------------------------------------------------------------
-----------------------
# PP (FC = 0x082)
#--------------------------------------------------------------------------------
-----------------------
# List of external functions (denoted by SFC and function name) in the PP
module that can be disabled

# SFC = 0 Set/Get PIN Pad Type                            (default) required
permissions = 00000000
```