

SmartCard-HSM XKEK Key Domain HowTo

Author	Andreas Schwier
Version	1.0
Date	2020-09-04

SmartCard-HSM XKEK Key Domain HowTo

# **Table of Contents**

1	Introduction	4
2	Prereguisite	4
	2.1 Installing Software	4
	2.2 Starting the Smart Card Shell	4
3	Setting up a XKEK Key Domain	6
	3.1 Initialize Device for using Key Domains	6
	3.2 Create the Group Signer.	7
	3.3 Adding a Device to the Group	8
4	Migrating Kevs in a XKEK Kev Domain	.14
5	Summary	18
-		-

## **1** Introduction

This document is a step-by-step description for configuring XKEK Key Domains with a SmartCard-HSM 4K.

# 2 Prerequisite

### 2.1 Installing Software

You need to install the Smart Card Shell at least in version 3.16 available from www.openscdp.org/scsh3.

### 2.2 Starting the Smart Card Shell

Start the Smart Card Shell and select a workspace directory. This can be any folder on your system, for example the workspace folder from the SmartCard-HSM Starterkit.

Please select your workspace directory or initial configuration script.		
When selecting a configuration script, then the workspace is determined by the directory location of the script.		
/home/asc/tmp/howto/smartcardhsm-workspace 🔻 Browse		
Use this as the default and do not ask again.		
ОК		

Run the KeyManager with "File/Key Manager (CTRL+M)".



The shell shows some status information of the device, in particular the device id (here DECC0401002).

# 3 Setting up a XKEK Key Domain

### 3.1 Initialize Device for using Key Domains

During SmartCard-HSM initialization you will need to specify how many key domains you want to configure. Select "Initialize Device" from the context menu attached to the "SmartCard-HSM" node in the outline.

File Edit Options Help         SmartCard-HSM (DECC0401)         User PIN not verified, 3 t         SO PIN not verified, 15 t         Export Device ID	<pre>Running setup script config.js Smart Card Shell Scripting Engine (scdp4j) 3.17.365 (c) 2005-2016 CardContact Systems GmbH, Minden, Germany (www.cardcontact.de) Enter 'help' for a command overview or 'quit' to close the shell &gt;load("keymanager/keymanager.js"); SmartCard-HSM Version 3.4 on JCOP 3 Free memory 85896 byte Issuer : CVC id-SC-HSM DEVIC CAR=DESRCACC100001 CHR=DEDICC0400001 CED=22. Okto Device : CVC id-SC-HSM Device CAR=DEDICC0400001 CHR=DECC040100200000 CED=19. Please right-click on nodes in the outline to see possible actions. For most operations you will need to authenticate first using a mechanism from the User PIN context menu. &gt; </pre>

Continue to the "Select Device Key Encryption Scheme" and select "Key Domains.

	Smart Card Shell V 🗸 🚫
File Edit Options Help	
SmartCard-HSM (DECC0401002)	Running setup script config.js
User PIN not verified, 3 tries remaining (63C3)	Smart Card Shell Scripting Engine (scdp4j) 3.17.365
$\sim$ SO PIN II $\sim$ Configuration Option $\sim$ $\sim$	(a) 2005 2016 Condentant Custom Cable Minden, Company (usu condentant de)
Select Device Key Encryption scheme	Enter 'help' for a command overview or 'quit' to close the shell
Select Device Key End yption scheme	>load("kevmanager/kevmanager.is"):
Key Domains	
OK Cancel	SmartCard-HSM Version 3.4 on JCOP 3 Free memory 85896 byte Issuer : CVC id-SC-HSM DICA CAR=DESRCACC100001 CHR=DEDICC0400001 CED=22. Oktob Device : CVC id-SC-HSM Device CAR=DEDICC0400001 CHR=DECC040100200000 CED=19. A
	Please right-click on nodes in the outline to see possible actions. For most operations you will need to authenticate first using a mechanism from the User PIN context menu. >
[ <b>]</b>	Shell Trace Tasks

On the next dialog confirm 1 key domain. The device is initialized and shows one key domain slot that is not yet created.

File Edit Options Help         SmartCard-HSM (DECC0401002)         User PIN authenticated (9000)         SO PIN not verified, 15 tries remaining (63CF)         Key domain 0 not created	<pre>Running setup script config.js Smart Card Shell Scripting Engine (scdp4j) 3.17.365 (c) 2005-2016 CardContact Systems GmbH, Minden, Germany (www.cardcontact.de) Enter 'help' for a command overview or 'quit' to close the shell &gt;load("keymanager/keymanager.js"); SmartCard-HSM Version 3.4 on JCOP 3 Free memory 85896 byte Issuer : CVC id-SC-HSM DICA CAR-DESRCACC100001 CHR-DEDICC0400001 CED=22. Okto Device : CVC id-SC-HSM Device CAR=DEDICC0400001 CHR=DECC040100200000 CED=19. </pre>
	Shell Trace Tasks

You can create up to 255 key domains and each one can be either a DKEK or XKEK key domain.

### 3.2 Create the Group Signer

The Group Signer is the certification instance that issues Key Domain Membership certificates for members of the key domain group. The signer is an ECDSA key on the brainpoolP256r1 curve. It can be created on any SmartCard-HSM. For demonstration purpose we create the key on the same device, using "Generate ECC key" from the SmartCard-HSM context menu. Choose "brainpoolP256r1" and a label "Group Signer". Leave algorithms empty or enter "70" for ECDSA.

If you now open the AT-CVREQ structure with the newly generated public key, then you can see the Public Point y 86, which starts with '0462AEA0'. The 32 bytes following the '04' (the x-coordinate of the public point) are the key domain UID that uniquely identifies the group signer and a key domain instance on a SmartCard-HSM.



### 3.3 Adding a Device to the Group

To add a SmartCard-HSM to the group controlled by the group signer, the device authentication public key must be certified in a Key Domain Membership (KDM) certificate. In order to do that, you will first need to export the Device ID of the SmartCard-HSM you want to add. The data written to the file is actually a concatenation of the device certificate and the device issuer's CA certificate. Choose "Export Device ID" from the context menu and save with the file name proposed in the dialog.

File Edit Options Help	
SmartCard-HSM (DECC0401002)	
Smart Card Shell Scripting Engine (scdp4i) 3.17.	5
- SO PIN no logical solution of the solution o	rmany (www.cardcontact.de)
← ← Group Si Enter file name for device id	lose the shell
/home/asc/share/projects/workspace_scsh/DECC040100200000.id Browse	
Сапсеі Free СС100001 С СС0400001	memory 85896 byte R=DEDICC0400001 CED=22. Oktober R=DECC040100200000 CED=19. Aug
Please right-click on nodes in the outline to ser For most operations you will need to authenticate mechanism from the User PIN context menu. >Initializing, please wait Initializing complete Generating key can take up to 60 seconds, please Key generated	possible actions. first using a Mait

SmartCard-HSM XKEK Key Domain HowTo

Next you will need to issue the KDM for the device. Right-click on the AT-CVREQ node under the group signer and select "Group Signer Operations".



You can now select, if you want to issue a KDM for a device with version less than 3.4 or above.

	Smart Card Shell V ^	$\otimes$
File Edit Options Help	7	
SmartCard-HSM (DECC0401002)	ning setup script config.js	
- So PIN networking 15 trice remaining (6005)	rt Card Shell Scripting Engine (scdp4j) 3.17.365	
	005-2016 CardContact Systems GmbH, Minden, Germany (www.cardcontact.de)	
Select signature format	netp for a command overview or quit to close the shell	
► 🔄 AT-CV Static key domain membership (>=3.4)	("keymanager/keymanager.js");	
OK Cancel	Card-HSM Version 3.4 on JCOP 3 Free memory 85896 byte r : CVC id-SC-HSM DICA CAR=DESRCACC100001 CHR=DEDICC0400001 CED=22. Okt e : CVC id-SC-HSM Device CAR=DEDICC0400001 CHR=DECC040100200000 CED=19.	ober Aug
Ple For mec >In Ini Gen Key Dev	ase right-click on nodes in the outline to see possible actions. most operations you will need to authenticate first using a hanism from the User PIN context menu. itializing, please wait tializing complete erating key can take up to 60 seconds, please wait generated ice Id exported to /home/asc/share/projects/workspace_scsh/DECC0401002000	00.i
S	mell Trace Tasks	

After "OK" you are prompted for the file containing the device id of the device you want to add.



The script will now ask, if you really want to add that device to the key domain controlled by the selected group signer. You can see that the key domain UID matches the public key of the group signer.

	Smart Card Shell	~ ^ 😣
File Edit Options Help		
SmartCard-HSM (DECC0401002)  Duser PIN authenticated (9000)  Duser PIN not verified, 15 tries remaining (63CF)  Duser PIN not created  Composigner(1)	Running setup script config.js Smart Card Shell Scripting Engine (scdp4j) 3.17.365 (c) 2005-2016 CardContact Systems GmbH, Minden, Germany (www.cardcontact Sys	ontact.de)
dd device DECC040100200000 to key domain 62AEA0	ED39ABF09F12F9EE1FC2BB35EFAC4DC5E51677BBA429F95DE5DA9170EC {C	ancel for No
	OK Cancel	
	For most operations you will need to authenticate first using a mechanism from the User PIN context menu. >Initialigng, please wait Initializing complete Generating key can take up to 60 seconds, please wait	-
	Key generated Device Id exported to /home/asc/share/projects/workspace_scsh/DECC	040100200000.
	Image: Shell state     Trace state	

The script automatically writes a file with extension ".kdm" in the workspace (here DECC040100200000-62AEA0ED39ABF09F12F9EE1FC2BB35EFAC4DC5E51677BBA429F95DE5DA9170EC.kdm).

Now you can create the key domain instance for that device in the key domain slot allocated during initialization. Right click on the key domain entry and select "Create XKEK Key Domain".

	Smart Card Shell V 🔨 🔨
File Edit Options Help	
SmartCard-HSM (DECC0401002)	Running setup script config.js
<ul> <li>         Output: Output to the second seco</li></ul>	Smart Card Shell Scripting Engine (scdp4j) 3.17.365
← Key domain 0 not c ← ← Group Signer(1) Create DKEK Key Domain	(c) 2005-2016 CardContact Systems GmbH, Minden, Germany (www.cardcontact.de) Enter 'help' for a command overview or 'quit' to close the shell
<ul> <li>Subject Key Ider Clear Action Control Contro Control Control Control Control Control Control Control Con</li></ul>	>load("keymanager/keymanager.js");
	SmartCard-HSM Version 3.4 on JCOP 3 Free memory 85896 byte Issuer : CVC id-SC-HSM DICA CAR=DESRCACC100001 CHR=DEDICC0400001 CED=22. Oktober Device : CVC id-SC-HSM Device CAR=DEDICC0400001 CHR=DECC040100200000 CED=19. Aug
	Please right-click on nodes in the outline to see possible actions. For most operations you will need to authenticate first using a mechanism from the User PIN context menu. >Initializing, please wait Initializing complete
	Generating key can take up to 60 seconds, please wait Key generated Device Id exported to /home/asc/share/projects/workspace scsh/DECC040100200000.i
	Input : 54 0D 9C 3F 76 E8 71 5F 0F 96 19 53 23 8B CD 76 3F 3C 09 4C ED 66 A4 9B Key Domain Membership written to /home/asc/share/projects/workspace_scsh/DECC04C
	Shell Trace Tasks

In the next step you need to select the .kdm file that contains the key domain membership certificate issued by the group signer.

	Smart Card Shell	$\sim$ $\sim$ $\otimes$
File Edit Optio	ons Help	
SmartCard-HSN	4 (DECC0401002) thenticated (9000) Smart Card Shell Scripting E	js ngine (scdp4i) 3.17.365
- 🗋 SU PIN hor	Configuration Option	∧ ⊗ , Minden, Germany (www.cardcontact.de)
oroup Si → Subje	Enter name for file containing key domain membership	'quit' to close the shell
- 📑 AT-CV	12F9EE1FC2BB35EFAC4DC5E51677BBA429F95DE5DA9170EC.kdm	vse
	OK Cancel	Free memory 85396 byte CC100001 CHR=DEDICC0400001 CED=22. Oktober CC0400001 CHR=DECC040100200000 CED=19. Aug
Please right-click on nodes in the outline to see possible actions. For most operations you will need to authenticate first using a mechanism from the User PIN context menu. >Initializing, please wait Initializing complete Generating key can take up to 60 seconds, please wait Key generated Device Id exported to /home/asc/share/projects/workspace_scsh/DECC0400 Input : 54 0D 9C 3F 76 E8 71 5F 0F 96 19 53 23 8B CD 76 3F 3C 09 4C EI Key Domain Membership written to /home/asc/share/projects/workspace_scsh/ Vorkspace_scsh/DECC0400		in the outline to see possible actions. med to authenticate first using a context menu. o 60 seconds, please wait asc/share/projects/workspace_scsh/DECC040100200000.i 5F 0F 96 19 53 23 8B CD 76 3F 3C 09 4C ED 66 A4 9B n to /home/asc/share/projects/workspace_scsh/DECC04C
	siten riace lasks	

After pressing "OK" the key domain instance is created. You can again see the key domain UID shown on the node.

Smart Card Shell 🗸 🔨 🛇			
File Edit Options Help			
SmartCard-HSM (DECC0401002)         □ User PIN authenticated (9000)         □ S0 PIN not verified, 15 tries remaining (63CF)         □ XKEK with KCV 00000000000000 in key domain 62AEA0ED39A         ♥ → Group Signer(1)         □ Subject Key Identifier: F778F3D9EE83F866EDAA351D1C5DCE         ♥ AT-CVREQ CAR=DECC040100200000 CHR=DECC0401002000	Running setup script config.js Smart Card Shell Scripting Engine (scdp4j) 3.17.365 (c) 2005-2016 CardContact Systems GmbH, Minden, Germany (www.cardcontac Enter 'help' for a command overview or 'quit' to close the shell >load("keymanager/keymanager.js"); SmartCard-HSM Version 3.4 on JCOP 3 Free memory 85896 byte Issuer CVC idesc.HSM DICA CARPERER ACC100001 CHR-DEDICCO400001 CED=22		
	Device : CVC id-SC-HSM Device CAR=DEDICC0400001 CHR=DECC040100200000 C Please right-click on nodes in the outline to see possible actions. For most operations you will need to authenticate first using a mechanism from the User PIN context menu. >Initializing, please wait Initializing complete Generating key can take up to 60 seconds, please wait Key generated Device Id exported to /home/asc/share/projects/workspace_scsh/DECC04010 Input : 54 0D 9C 3F 76 E8 71 5F 0F 96 19 53 23 8B CD 76 3F 3C 09 4C ED Key Domain Membership written to /home/asc/share/projects/workspace_scs <b>Membership Trace Tasks</b>		

Now that we have created a XKEK Key Domain, we can start generating keys in it. Let's right-click on the key domain and select "Generate AES Key".

🗲 Smart Card Shell 🗸 🔨 😒			
File Edit Options Help			
File Edit Options Help SmartCard-HSM (DECC0401002) Ouser PIN authenticated (9000) SO PIN not verified, 15 tries remain Strike with KCV 00000000000000000000000000000000000	Running setup script co Smart Card Shell Script: 2005-2016 CardConta ar 'help' for a comm: 2005-2016 CardConta ar 'help' for a comm: 2005-2016 CardConta ar 'help' for a comm: Create Exchange Key Associate XKEK Key Domain Delete Key Encryption Key Delete Key Domain Ptease right-click on nu For most operations you mechanism from the User >Initializing, please wi Initializing complete Generated Device Id exported to // Input : 54 OD 9C 3F 76 I Key Domain Membership wi	nfig.js ing Engine (scdp4j) 3.17.365 ct Systems GmbH, Minden, Germany (www.cardcontac and overview or 'quit' to close the shell nager.js"); .4 on JCOP 3 Free memory 85896 byte DICA CAR=DESRCACC100001 CHR=DEDICC0400001 CED=22 Device CAR=DEDICC0400001 CHR=DECC040100200000 CE odes in the outline to see possible actions. will need to authenticate first using a PIN context menu. ait up to 60 seconds, please wait home/asc/share/projects/workspace_scs/DECC04016 E8 71 SF 0 9 09 19 53 23 88 CD 76 3F 3C 09 4C ED ritten to /home/asc/share/projects/workspace_scs	
	Shell Trace Tasks	s	

For the new key we select "WRAP" to allow exporting the key from the key domain.



The new key is shown as a member of the key domain. It can be used just like any other key. It shows up normally in PKCS#11 or the Java Key Store.

🔨 Smart Card Shell 🗸 🔨 😒			
File Edit Options Help			
SmartCard-HSM (DECC0401002) User PIN authenticated (9000) SO PIN not verified, 15 tries remaining (63CF) XKEK with KCV 00000000000000 in key domain 62AEA0ED39A → AES-Key(2) Subject Key Identifier: 02 Algorithms: DERIVE_SP800_56C, WRAP → Group Signer(1) Subject Key Identifier: F778F3D9EE83F866EDAA351D1C5DCE AT-CVREQ CAR=DECC04010020000 CHR=DECC0401002000	Running setup script config.js Smart Card Shell Scripting Engine (scdp4j) 3.17.365 (c) 2005-2016 CardContact Systems GmbH, Minden, Germany (www.cardcont Enter 'help' for a command overview or 'quit' to close the shell >load("keymanager/keymanager.js"); SmartCard-HSM Version 3.4 on JCOP 3 Free memory 85896 byte Issuer : CVC id-SC-HSM DICA CAR=DESRCACC100001 CHR=DEDICC0400001 CED- Device : CVC id-SC-HSM Device CAR=DEDICC0400001 CHR=DECC040100200000 Please right-click on nodes in the outline to see possible actions. For most operations you will need to authenticate first using a mechanism from the User PIN context menu. Initializing complete Generating key can take up to 60 seconds, please wait Key generated Device Id exported to /home/asc/share/projects/workspace_sch/DECC040 Input : 54 0D 9C 3F 76 E8 71 5F 0F 96 19 53 23 8B CD 76 3F 3C 09 4C E Key generated		
	Shell Trace Tasks		

The KCV of 00..00 indicates that currently no XKEK is present to wrap or unwrap keys.

# 4 Migrating Keys in a XKEK Key Domain

Now that we have created a XKEK Key Domain and a contained AES key, we want to migrate that key to another SmartCard-HSM that is also part of the key domain. You will need to repeat step 3.3 for each device you would like to add.

For the purpose of this demonstration, I will use a single device. The steps are identical if you export and import into the different devices.

For exporting a key from the key domain we first need to have a Key Encryption Key (KEK). Unlike with DKEKs that are imported using key shares, a XKEK is dynamically created as the result of performing a key agreement with EC Diffie-Hellman. Both, sender and receiver must generate an EC key pair and make the public key available to the other side. Each side then performs an ECDH operation with it's private key and the other side's public key. The result is a shared XKEK on both side. This XKEK is then used to wrap keys for export or to unwrap key after import.

The first step is to create the EC key pair for the key agreement operation. Choose "Generate ECC Key" from the key domain's context menu. It is important to select the "ECDH\_XKEK" algorithm (and only this algorithm).



If you now look into the AT-CVREQ of the newly generated public key, then you can see that it contains an extension id-key-domain-uid that indicates the key domain in which this key was generated. The id-key-domain-uid extension is only added for ECC keys with ECDH\_XKEK set in the algorithm list.



We save the AT-CVREQ and it's certificate chain into a file using "Export Public Key".

	Smart Card Sł	iell v ^ 😒
File Edit Options Help		
SmartCard-HSM (DECC0401002)		Smart Card Shell Scripting Engine (scdp4j) 3.17.365
– 🗋 User PIN authenticated (9000)		(a) 2005 2016 Condentest Systems CmbH Minden, Cormony (ww
🛛 — 🗋 SO PIN not verified, 15 tries remaining (63CF)		Enter 'help' for a command overview or 'quit' to close the
← 📑 XKEK with KCV 0000000000000000 in key dom	ain 62AEA0ED39ABF09F12F	
AES-Key(2)		>load("keymanager/keymanager.js");
Containge Rey(3)     D Subject Key Identifier: 9E69AC30AECA5C	050EA751CA69E759EE9B6A	SmartCard-HSM Version 3.4 on JCOP 3 Free memory 85
- Algorithms; ECDH XKEK		Issuer : CVC id-SC-HSM DICA CAR=DESRCACC100001 CHR=DEDICC04
- 📮 AT-CVREQ CAR=DECC040100200000 CH	P-DECC040100200000 oCA	Device : CVC IU-SC-HSM Device CAR=DEDICC0400001 CHR=DECC040
🕈 🗝 Group Signer(1)	Export Public Key	Lease right-click on nodes in the outline to see possible
Subject Key Identifier: F778F3D9EE83F866E	Group Signer Operation	s")r most operations you will need to authenticate first usi uchanism from the User PIN context menu
AT-CVREQ CAR=DECC040100200000 CHR=L	Derive XKEK	initializing, please wait
	Import Certificate	itializing complete
	Dumn Certificate	merating key can take up to bu seconds, please wait
	Dump Certificate	pevice Id exported to /home/asc/share/projects/workspace_sc
		Input : 54 0D 9C 3F 76 E8 71 5F 0F 96 19 53 23 8B CD 76 3F
		Generating key can take up to 60 seconds, please wait
		Key generated
		Generating key can take up to 60 seconds, please wait
	Þ	Shell Trace Tasks

Having done that on both sides, we are ready to agree a XKEK. For that we choose "Derive XKEK" and select the file containing the public key of the other side (our own public key will work as well).

As a result the XKEK node shows the Key Check Value (KCV) of the freshly agreed XKEK.



Now that we have a XKEK, we can wrap and export our AES-Key. Select "Export Key and Certificate" on the AES-Key node (AES keys of course don't have a certificate, so you can ignore the warning).



After selecting the file name, the key is wrapped under the XKEK, exported from the device and written into the file.

We can now delete the AES key and clear the XKEK with "Delete Key Encryption Key" from the key domain's context menu.



The XKEK is cleared and the KCV is reset to 00..00

	Smart Card Shell	~ ^ 😣
File Edit Options Help		
SmartCard-HSM (DECC0401002) User PIN authenticated (9000) SO PIN not verified, 15 tries remaining (63CF) C XKEK with KCV 000000000000000 in key domain 62AEA0ED C → Exchange Key C → Group Signer(1)	39ABF09F12F	73 [ APPLICATION 19 ] IMPLICIT SEQUENCE SIZE[ 4 _ OBJECT IDENTIFIER = { id-key-domain-uid } 80 [ CONTEXT 0 ] SIZE[ 32 ) 00000 62 AE A0 ED 39 AB F0 9F 12 F9 EE 1F C 0010 AC 4D C5 E5 16 77 BB A4 29 F9 5D E5 D 5F37 [ APPLICATION 55 ] SIZE[ 64 ] 0000 15 AB 14 D6 42 54 AB 24 AD 80 33 12 8F 70 7 0010 A7 72 5C F2 E2 47 EE 10 43 A1 BF C7 B3 6A F 0020 A5 3E 92 D1 C7 76 22 96 BA F4 CD 2D 36 92 A 0030 28 E8 A1 E5 E9 3C 7F 2A 8B 90 3F E2 3D 42 2 42 [ APPLICATION 55 ] SIZE[ 64 ] 0000 14 45 43 43 30 34 30 31 30 30 32 30 30 30 30 5F37 [ APPLICATION 55 ] SIZE[ 64 ] 0000 11 A7 F8 F2 99 70 C8 F8 1C 15 0B 01 CD FF 1F 0010 75 A7 4A 34 27 5F EB 1C 2B 15 F0 1A CA FD 04 0020 6F 0C DF CF 7E E4 60 21 E7 FB 67 B8 15 8C 0F 0030 35 17 29 60 83 46 27 C6 A7 25 5C 54 5D D1 A1 noring meta data or certificate: Card (CARD_INVALID_SW/27 at /home/asc/share/projects/workspace_scsh/scsh/sc-hsm/ at /home/asc/share/projects/workspace_scsh/scsh/sc-hsm/ at /home/asc/opt/CardContact/scsh3/keymanager/keymanage at /home/asc/opt/CardContact/scsh3/keymanager/keymanage at /home/asc/share/projects/workspace_scsh/AES
		snell Trace Tasks

To import the key we need to repeat the XKEK key agreement using the public key of the sender in the "Derive XKEK" function. That generates the XKEK, whose KCV is show at the node.

For importing the AES-Key you need to select "Import Key and Certificate" from the SmartCard-HSM node. No need to select the right key domain – that is done automatically, as the wrapped key contains the KCV of the XKEK in it's meta data.

	Smart Card Shell	· · · · 8
File Edit Options Help		
<ul> <li>SmartCard-HSM (DECC0401</li> <li>Generate RSA Key Generate ECC Key Generate ECC Key Generate ECC Key Generate AES Key</li> <li>★ → Exchange Key(3)</li> <li>▲ Algorithms: ECDH</li> <li>▲ AlgorKEQ CAR=D</li> <li>★ → Group Signer(1)</li> </ul>	IED39ABF09F12F CA69E759FE9B6A 0100200000 oCA	0030 38 97 63 68 28 D9 4A 20 9F 88 85 E1 75 0040 F3 87 [ CONTEXT 7 ] SIZE( 1 ) 0000 01 5F20 [ APPLICATION 32 ] SIZE( 16 ) 0000 44 45 43 43 30 34 30 31 30 30 32 30 30 30 65 [ APPLICATION 5 ] IMPLICIT SEQUENCE SIZE( 49 ) 73 [ APPLICATION 19 ] IMPLICIT SEQUENCE SIZE( 49 ) 73 [ APPLICATION 19 ] IMPLICIT SEQUENCE SIZE( 40 08 ] CONTEXT 0 ] SIZE( 32 ) 0000 62 AE A0 ED 39 AB F0 9F 12 F9 EE 1F C 0010 AC 4D C5 E5 16 77 BB A4 29 F9 5D E5 D 5F37 [ APPLICATION 55 ] SIZE( 64 ) 0000 15 AB 14 D6 42 54 AB 24 AD 80 33 12 8F 70 7 0010 A7 72 5C F2 E2 47 EE 10 43 A1 BF C7 B3 6A F 0020 A5 3E 92 D1 C7 76 22 96 BA F4 CD 2D 36 92 A 0030 28 EB A1 E5 E9 3C 7F 2A 8B 90 3F B2 3D 42 2 42 [ APPLICATION 2 ] SIZE( 16 ) 0000 11 A7 F8 F2 59 70 C8 FB 1C 15 0B 01 CD FF 1F 0010 75 A7 4A 34 27 5F EB 1C 2B 15 F0 1A CA FD 04 0020 6F 0C DF CF 7E E4 60 21 E7 FB 67 BB 15 8C 0F 0030 35 17 29 60 83 46 27 C6 A7 25 5C 54 5D D1 A1
		Snell Trace Tasks

Et voila, the key is back in the device.

![](_page_17_Picture_3.jpeg)

## 5 Summary

Key Domains are a powerful feature of the SmartCard-HSM to implement secure key management.

While DKEK key domains with imported key shares are easy to set-up, they put a high burden on the organizational overhead to securely generate and control DKEK shares. If you loose control over your DKEK shares, then confidentially of your keys is at risk. DKEK key shares allow to recover the Key Encryption Key

that wraps your key material exported from the SmartCard-HSM.

XKEK Key Domains are different and there is a guarantee, that a key can never leave a XKEK key domain. The Key Encryption Key is always the result of an ECDH operation and the private key can not leave the key domain either. Effectively, there is no way to establish the XKEK outside of the SmartCard-HSM, so it is impossible to decrypt the wrapped key material.

Implementing key backup with a XKEK Key Domain means adding a SmartCard-HSM to the key domain and migration keys into that device. This is controlled by the group signer that you need to keep under control. If you need to replace a broken device, just add a new device to the group and you are ready to migration key material into that device.

As you've seen from the steps above, managing a XKEK manually is a complex task. Our suggestion is to write key management scripts using the SmartCard-HSM. You can take a look at the key manager script (keymanager/keymanager.js in the Smart Card Shell installation) to learn how that can be done.