Copyright © STS Association

STS Association

STS600-9-1
Edition 1.0
Jan 2019

Key management - Worked example with deterministic test vectors

Contents

1 Introduction

This document is a worked example of the STS600-4-2 Key Management Process including Security Module (SM) Manufacturer key generation, KMC key generation, SM initialisation, SM PUBKEY transfer from the SM Manufacturer to the KMC, the Vending Key Load Request from the SM to the KLF, the Vending Key Load Response and Key Load File (KLF) from the KMC to the SM, and the import of Vending Keys from the KLF into the SM.

Several references are made to sections in the STS600-4-2 specification throughout this document. All values shown are deterministic in nature and may be used as test vectors for the implementation of STS600-4-2 key agreement processes.

In a live situation, KMC and SM public keys values are ephemeral during generation but will remain static after initial generation, for the specified validity of these keys. The ECDH keys used in VKLOADREQ are ephemeral for each VKLOADREQ and VKLOADRESP.

An independent software program has been implemented to verify these static vectors using the openssl cryptographic library.

This document is based on the Prism document: *Worked example of STS600-4-2 Key Agreement*
TV Name: TV_STSA1 (Key Agreement Static Test Vectors: STS 600-4-2 Edition 1.2).

2 Normative References

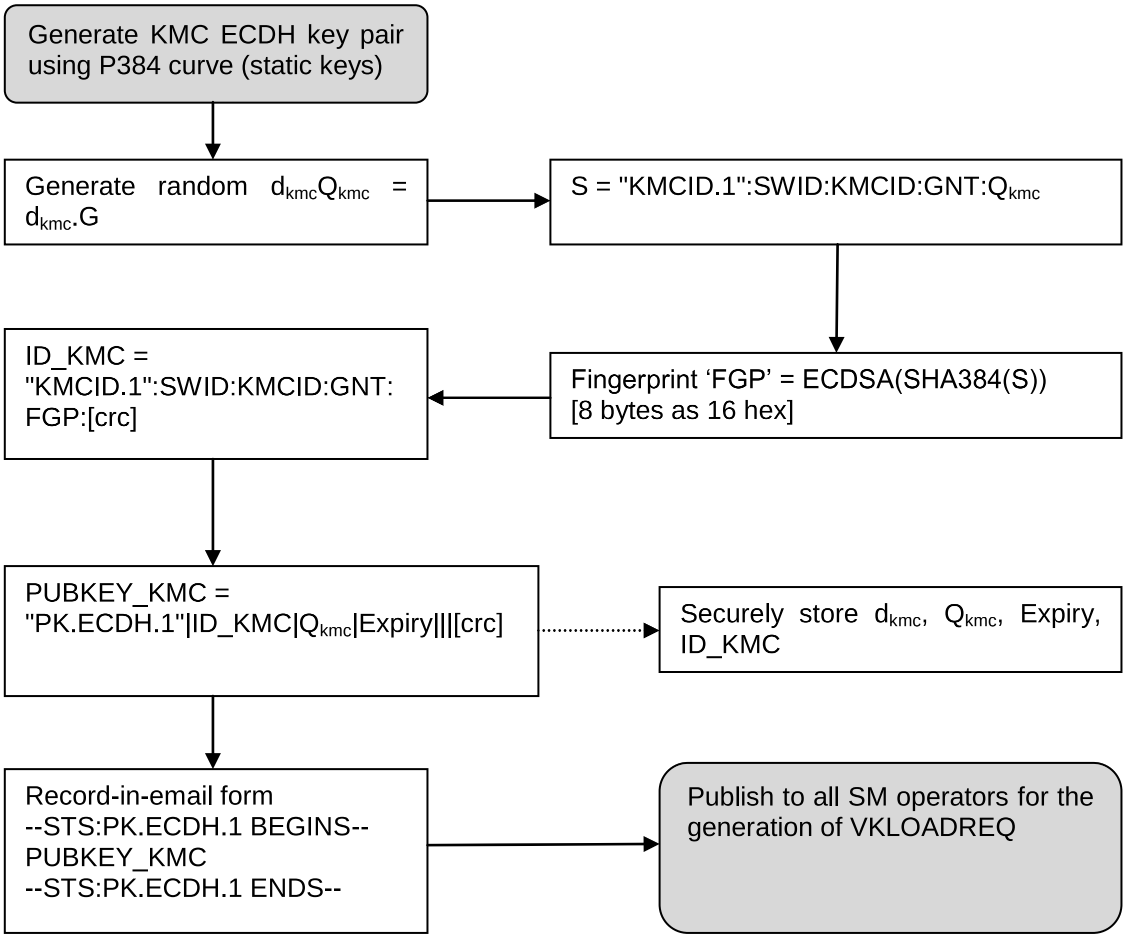
STS600-4-2 *STANDARD TRANSFER SPECIFICATION – Companion Specification – Key Management System*

3 Definitions

For definitions used in this document, please refer to STS600-4-2.

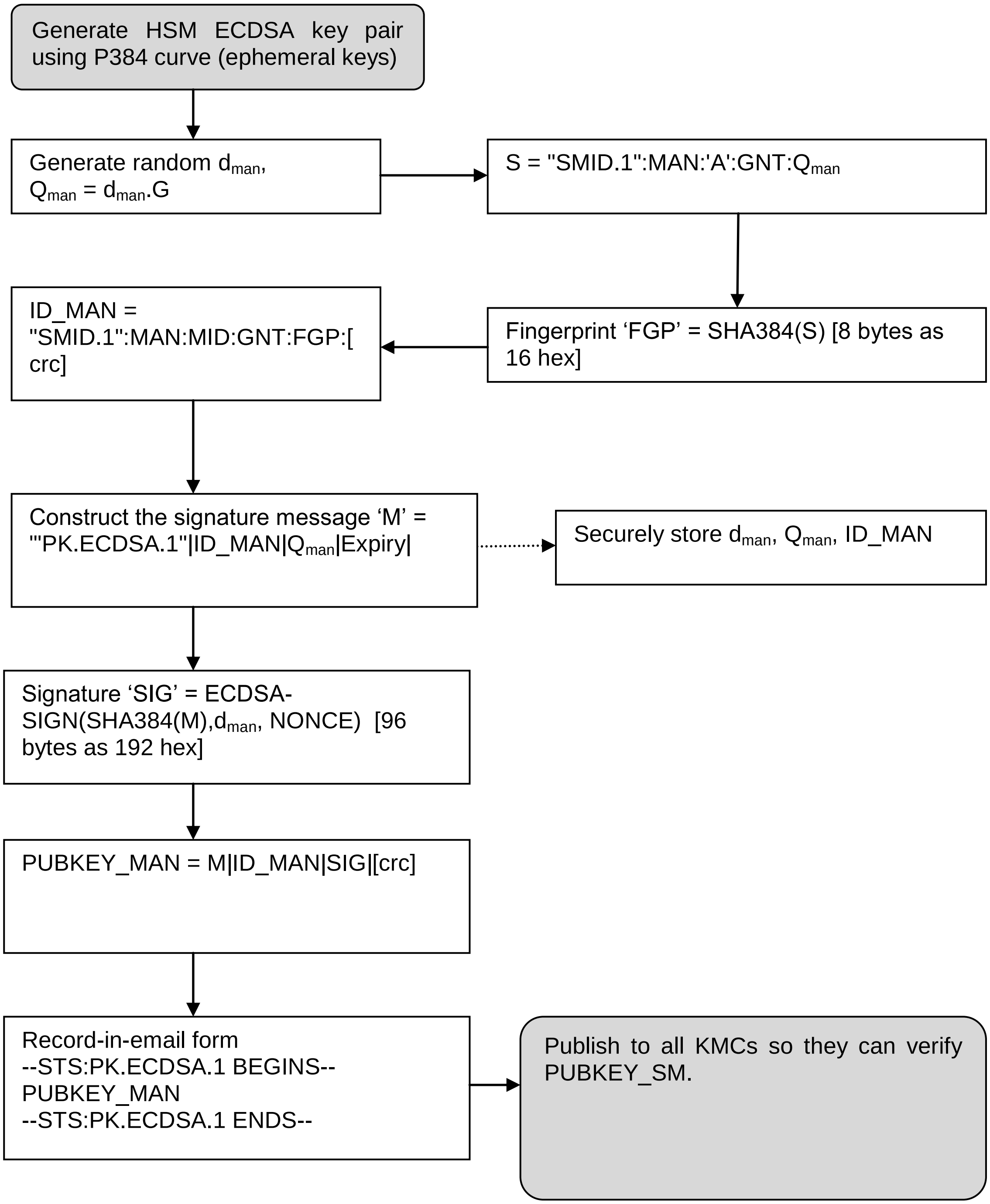
4 KMC ECDH PUBKEY generation process (static)

This is the KMC public key used for key agreement between the KMC and the SM (when the VKLOADREQ is generated).


Generate KMC ECDH key pair using P384 curve (static keys)
Generate random $d_{kmc}Q_{kmc} = d_{kmc}.G$
S = "KMCID.1":SWID:KMCID:GNT:Q_{kmc}
ID_KMC = "KMCID.1":SWID:KMCID:GNT: FGP:[crc]
Fingerprint 'FGP' = ECDSA(SHA384(S)) [8 bytes as 16 hex]
PUBKEY_KMC = "PK.ECDH.1"|ID_KMC|Q_{kmc}|Expiry|||[crc]
Securely store d_{kmc}, Q_{kmc}, Expiry, ID_KMC
Record-in-email form
--STS:PK.ECDH.1 BEGINS--
PUBKEY_KMC
--STS:PK.ECDH.1 ENDS--
Publish to all SM operators for the generation of VKLOADREQ

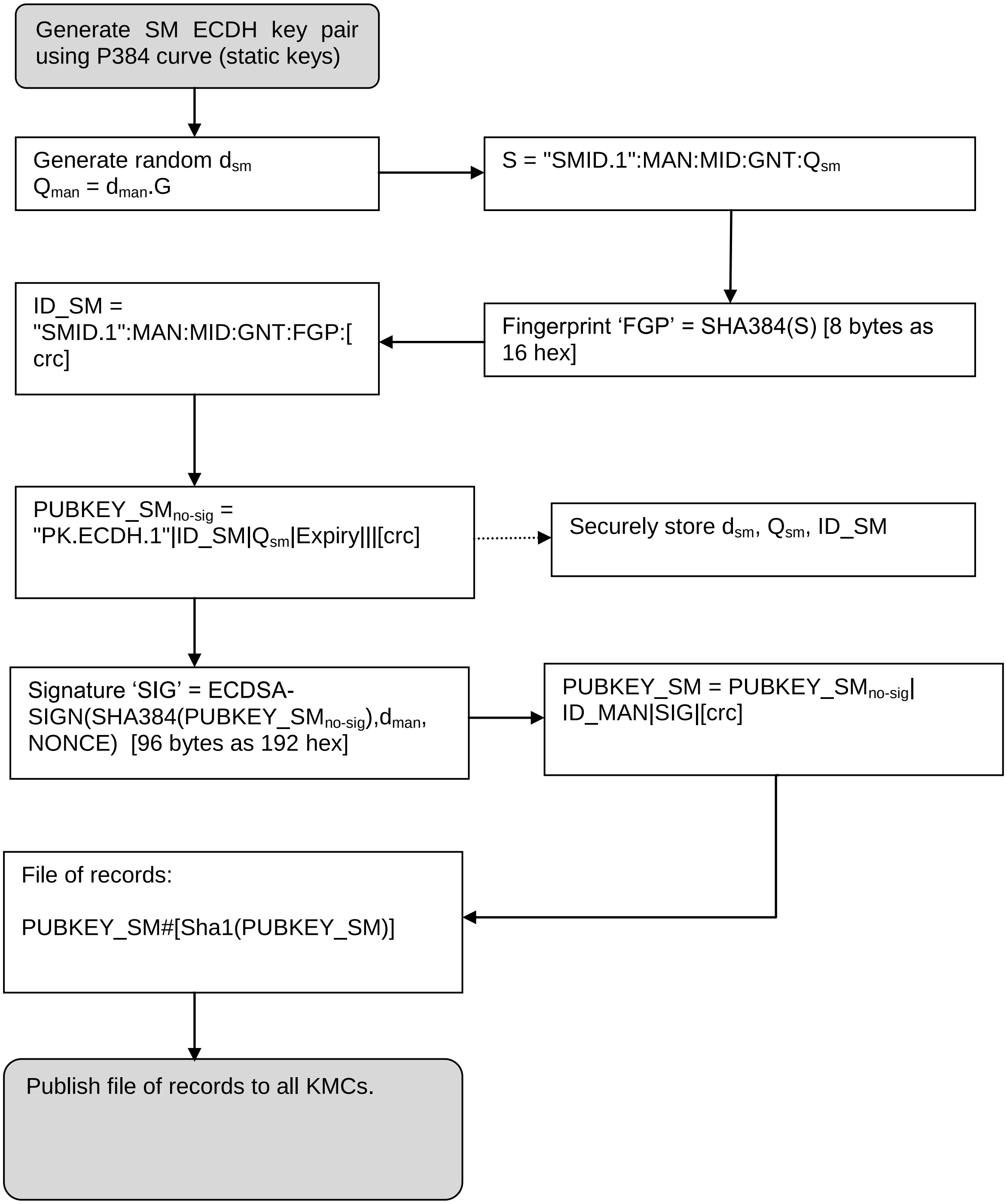
5 SM Manufacturer ECDSA PUBKEY_MAN generation process

This is the SM Manufacturer public key used to transfer SM public keys from the SM Manufacturer to the KMC.


Generate HSM ECDSA key pair using P384 curve (ephemeral keys)
Generate random d_{man}, Q_{man} = d_{man}.G
S = "SMID.1":MAN:'A':GNT:Q_{man}
Fingerprint 'FGP' = SHA384(S) [8 bytes as 16 hex]
ID_MAN = "SMID.1":MAN:MID:GNT:FGP:[crc]
Construct the signature message 'M' = '"PK.ECDSA.1"|ID_MAN|Q_{man}|Expiry|
Securely store d_{man}, Q_{man}, ID_MAN
Signature 'SIG' = ECDSA-SIGN(SHA384(M),d_{man}, NONCE) [96 bytes as 192 hex]
PUBKEY_MAN = M|ID_MAN|SIG|[crc]
Record-in-email form
--STS:PK.ECDSA.1 BEGINS--
PUBKEY_MAN
--STS:PK.ECDSA.1 ENDS--
Publish to all KMCs so they can verify PUBKEY_SM.

6 SM Manufacturer ECDH PUBKEY_SM generation process

This is the SM public key used for key agreement between the KMC and the SM; it is transferred from the SM Manufacturer to the KMC in a message that is digitally signed by the SM Manufacturer.


Generate SM ECDH key pair using P384 curve (static keys)
Generate random d_{sm}
$Q_{man} = d_{man}.G$
S = "SMID.1":MAN:MID:GNT:Q_{sm}
Fingerprint ‘FGP’ = SHA384(S) [8 bytes as 16 hex]
ID_SM = "SMID.1":MAN:MID:GNT:FGP:[crc]
PUBKEY_SM$_{no-sig}$ = "PK.ECDH.1"|ID_SM|Q_{sm}|Expiry|||[crc]
Securely store d_{sm}, Q_{sm}, ID_SM
Signature ‘SIG’ = ECDSA-SIGN(SHA384(PUBKEY_SM$_{no-sig}$),d_{man}, NONCE) [96 bytes as 192 hex]
PUBKEY_SM = PUBKEY_SM$_{no-sig}$|ID_MAN|SIG|[crc]
File of records:
PUBKEY_SM#[Sha1(PUBKEY_SM)]
Publish file of records to all KMCs.

7 PUBKEY_KMC: Generate KMC ECDH Key Pair

This section describes the generation of the KMC public key for distribution to all SM operators. This public key is used for key agreement between the KMC and the SM (when the VKLOADREQ is generated).

STEP	Attribute	Value
1	KMCID	'TEST1'
	SWID	'sts-KeyAgreement-1.2'
2	Generate PUBKEY_KMC (STS600-4-2 10.3 & 10.3.1)	
2.1	Generate ECDH key pair using GENERATE-KEY() in curve P-384 (STS600-4-2 6.8)	
2.1.1	KMC Static Key (private portion) - Randomly select d in the range 1 to (curve.n - 1)	
	D	x'A6531F356BD1DAC52C62ED2DBF3A6FB2CE9CDC06C55D07E93507E90774FE664BCB281C939DE5678F5FB007298D422F50
2.1.2	Static Key (public portion) Q in uncompressed affine coordinates (STS600-4-2 6.5)	Compute Q = d.G
	Q.x	x'4DED24DCA96783C3B240CEEBBB1D69EA36F96F15ACCB13D2EA68B698DDA34443A465E85531904F36F387F5C8908F7DFA
	Q.y	x'4EF8CE0065F6EA5CEC23578EC1C96E4662F2B74184F91A552F9AFB96F99F3EEAFC8C1B5A800857E5B2AC3F0CB2197BD5
	Qoctetstr	x'044DED24DCA96783C3B240CEEBBB1D69EA36F96F15ACCB13D2EA68B698DDA34443A465E85531904F36F387F5C8908F7DFA4EF8CE0065F6EA5CEC23578EC1C96E4662F2B74184F91A552F9AFB96F99F3EEAFC8C1B5A800857E5B2AC3F0CB2197BD5
2.2	Construct ID_KMC	SWID and KMCID given above
	GNT	'20180110T120000Z'
	Fingerprint (STS600-4-2 7.1):	
	Hash input S	'KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:044DED24DCA96783C3B240CEEBBB1D69EA36F96F15ACCB13D2EA68B698DDA34443A465E85531904F36F387F5C8908F7DFA4EF8CE0065F6EA5CEC23578EC1C96E4662F2B74184F91A552F9AFB96F99F3EEAFC8C1B5A800857E5B2AC3F0CB2197BD5:'
	SHA384 hash output	x'4712CFF444570C8AF67017644733D18E12E4932BB5597608AEBC36147D88DDF9BCADBCD571B5B27BBE4B7AD2FC0D333B
	Fingerprint	'4712CFF444570C8A'
	ID_KMC	'KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:4712CFF444570C8A:

		4C31'
2.3	Construct PUBKEY_KMC	ID_KMC and Q are given above
	Expiry	'20210110T120000Z'
	PUBKEY	'PK.ECDH.1|KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:4712CFF444570C8A:4C31|044DED24DCA96783C3B240CEEBBB1D69EA36F96F15ACCB13D2EA68B698DDA34443A465E85531904F36F387F5C8908F7DFA4EF8CE0065F6EA5CEC23578EC1C96E4662F2B74184F91A552F9AFB96F99F3EEAFC8C1B5A800857E5B2AC3F0CB2197BD5|20210110T120000Z|||B8F9'
3	Securely store d, Q, Expiry, ID_KMC	
4	Outputs	
	Fingerprint	'4712CFF444570C8A'
	Record-in-email format	--STS:PK.ECDH.1 BEGINS-- PK.ECDH.1|KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:4712CFF444570C8A:4C31|044DED24DCA96783C3B240CEEBBB1D69EA36F96F15ACCB13D2EA68B698DDA34443A465E85531904F36F387F5C8908F7DFA4EF8CE0065F6EA5CEC23578EC1C96E4662F2B74184F91A552F9AFB96F99F3EEAFC8C1B5A800857E5B2AC3F0CB2197BD5|20210110T120000Z|||B8F9 --STS:PK.ECDH.1 ENDS--

8 PUBKEY_MAN: Generate ECDSA Key Pair

This section describes the generation of the SM ECDSA public key for distribution to the KMC. This public key is used to transfer SM public keys from the SM Manufacturer to the KMC.

STEP	Attribute	Value
1	MANUFACTURER	'Prism'
2	Generate PUBKEY_MAN (STS600-4-2 8 & 8.1)	
2.1	Generate ECDSA key pair using GENERATE-KEY() in curve P-384 (STS600-4-2 6.8)	
2.1.1	Manufacturer Key (private portion)	Randomly select d in the range 1 to (curve.n - 1)
	D	x'DA3E238A54D908957A8BD30DD1110A764CB09DBF7FFB753010190F44D172FF7051B562504FFD60C373A1FD22CE0323CF
2.1.2	Manufacturer Key (public portion)	Compute Q = d.G Q in uncompressed affine coordinates (per Section 6.5)
	Q.x	x'ECDC178D70C8B495ED5F5A05A68393D710A6BB9DACDEDAD28EA5886E5AE683FF287A26FD23ABFBF20682F218DF76D724
	Q.y	x'CD56C1E85A19F65C5DD7AAA590F342D95503CDE643AB7E0691B9725BABEEF15C7D26857A0416D96F61682FA83A14991A
	Qoctetstr	x'04ECDC178D70C8B495ED5F5A05A68393D710A6BB9DACDEDAD28EA5886E5AE683FF287A26FD23ABFBF20682F218DF76D724CD56C1E85A19F65C5DD7AAA590F342D95503CDE643AB7E0691B9725BABEEF15C7D26857A0416D96F61682FA83A14991A
2.2	Construct ID_MAN	
	MANUFACTURER and Q are given above	
	GNT	'20180115T140000Z'
	Fingerprint (STS600-4-2 7.1)	
	Hash input (S)	'SMMAN.1:Prism:A:20180115T140000Z:04ECDC178D70C8B495ED5F5A05A68393D710A6BB9DACDEDAD28EA5886E5AE683FF287A26FD23ABFBF20682F218DF76D724CD56C1E85A19F65C5DD7AAA590F342D95503CDE643AB7E0691B9725BABEEF15C7D26857A0416D96F61682FA83A14991A:'
	SHA384 hash output	x'105717ACA4A508529DF381AE435D8CAD43315CECB4C81A3C945DBFCE01AC6851D0F763AD63149012336A0FF2C5F4F0A0
	Fingerprint	'105717ACA4A50852'
	PKID (this is a static value used in the ECDH ephemeral keys)	'SMMAN.1:Prism:A:20180115T140000Z:105717ACA4A50852:8B04'
2.3	Construct PUBKEY_MAN (self-signed)	PKID and Q are given above
	Expiry	'20210115T140000Z'
	Signature (STS600-4-2 7.2)	
	Input message (M)	'PK.ECDSA.1|SMMAN.1:Prism:A:20180115T140000Z:105717ACA4A50852:8B04|04ECDC178D70C8B495ED5F5A05A68393D710A6BB9DACDEDAD28EA5886E5AE683FF287A26FD23ABFBF20682F218DF76D724

<table>
<tr><td></td><td></td><td>CD56C1E85A19F65C5DD7AAA590F342D95503CDE643AB7E0691B9725BABEEF15C7D26857A0416D96F61682FA83A14991A|20210115T140000Z|'</td></tr>
<tr><td></td><td>Randomly generated NONCE</td><td>x'B899E85100941DC34E070668CBD9AFDE55B346D000AD582B3E1E9BBC3DCF4217DC020F37FAAA5C0EC3814D38E122F6A6'</td></tr>
<tr><td></td><td>Take SHA384(M) to get e</td><td>x'F0D770091886A1B1F34FAF8598597D90D8D9F3DBDAD150522AEEC4C5300C0CCB3B8456858D3BA0D108D86E520B8D49BF'</td></tr>
<tr><td></td><td>Signature Using manufacturer private key in 2.1.1 above.</td><td>x'5F82E6B0FB0837F383D3E1E7D7061CC0A42DDA3530DB68E21F03F185271D85A46A9D4369FC1507B233C1CAFDA61D01020D6A649289CFBC05F919E3EFCBCDF8CFFFD756AB9B3DD63D66C3ED6D99BE3184124CDECA5A59FC7A14316A2DD0265AA4'</td></tr>
<tr><td></td><td>PUBKEY</td><td>'PK.ECDSA.1|SMMAN.1:Prism:A:20180115T140000Z:105717ACA4A50852:8B04|04ECDC178D70C8B495ED5F5A05A68393D710A6BB9DACDEDAD28EA5886E5AE683FF287A26FD23ABFBF20682F218DF76D724CD56C1E85A19F65C5DD7AAA590F342D95503CDE643AB7E0691B9725BABEEF15C7D26857A0416D96F61682FA83A14991A|20210115T140000Z|SMMAN.1:Prism:A:20180115T140000Z:105717ACA4A50852:8B04|5F82E6B0FB0837F383D3E1E7D7061CC0A42DDA3530DB68E21F03F185271D85A46A9D4369FC1507B233C1CAFDA61D01020D6A649289CFBC05F919E3EFCBCDF8CFFFD756AB9B3DD63D66C3ED6D99BE3184124CDECA5A59FC7A14316A2DD0265AA4|1F11'</td></tr>
<tr><td>3</td><td>Outputs</td><td></td></tr>
<tr><td></td><td>Fingerprint</td><td>'105717ACA4A50852'</td></tr>
<tr><td></td><td>Record-in-email format</td><td>--STS:PK.ECDSA.1 BEGINS--
PK.ECDSA.1|SMMAN.1:Prism:A:20180115T140000Z:105717ACA4A50852:8B04|04ECDC178D70C8B495ED5F5A05A68393D710A6BB9DACDEDAD28EA5886E5AE683FF287A26FD23ABFBF20682F218DF76D724CD56C1E85A19F65C5DD7AAA590F342D95503CDE643AB7E0691B9725BABEEF15C7D26857A0416D96F61682FA83A14991A|20210115T140000Z|SMMAN.1:Prism:A:20180115T140000Z:105717ACA4A50852:8B04|5F82E6B0FB0837F383D3E1E7D7061CC0A42DDA3530DB68E21F03F185271D85A46A9D4369FC1507B233C1CAFDA61D01020D6A649289CFBC05F919E3EFCBCDF8CFFFD756AB9B3DD63D66C3ED6D99BE3184124CDECA5A59FC7A14316A2DD0265AA4|1F11
--STS:PK.ECDSA.1 ENDS--</td></tr>
</table>

9 PUBKEY_SM: Generate ECDH KEY PAIR

This section describes the generation of the SM ECDH public key for distribution to the KMC. This public key is used used for key agreement between the KMC and the SM (when the VKLOADREQ is generated); it is transferred from the SM Manufacturer to the KMC in a message that is digitally signed by the SM Manufacturer.

STEP	Attribute	Value
1	Prerequisites: SM (STS600-4-2 9.1)	
	HWID	'Prism-VSM-1'
	FWID	'STS6-001'
	MID	'06000001'
2	Generate PUBKEY_SM (STS600-4-2 9.2 & 9.2.1)	
2.1	Set GNT to the current date according to the RTC	
	GNT	'20180120T090000Z'
2.2	Generate ECDH key pair using GENERATE-KEY() in curve P-384 (STS600-4-2 6.8)	
2.2.1	SM Static Key (private portion)	Randomly select d in the range 1 to (curve.n - 1)
	d	x'62EB5B3F0C35325D14C31423717870773F9FD6C767CDD9088013512F3FB08186698F2F2B1298049E944346554664869B'
2.2.2	SM Static Key (public portion)	Compute Q = d.G
	Q.x	x'795CF0B4D74920C64A6879504A6DE9CA788076D946D2F70F8981C01137752C7C050DC6FA61C2CB3D77EFE4275826857C'
	Q.y	x'C286805608F43C2E9AC8752600D99FE92CCFB7E146742F0DC9C74CEF6568CBB75AB075D2DFED2EE5531554FA844B8DBC'
	Qoctetstr Q in uncompressed affine coordinates (STS600-4-2 6.5)	x'04795CF0B4D74920C64A6879504A6DE9CA788076D946D2F70F8981C01137752C7C050DC6FA61C2CB3D77EFE4275826857CC286805608F43C2E9AC8752600D99FE92CCFB7E146742F0DC9C74CEF6568CBB75AB075D2DFED2EE5531554FA844B8DBC'
2.3	Construct ID_SM	
	MID, GNT, and Q are given above	
	MANUFACTURER (from SMMAN)	'Prism'
	Fingerprint (STS600-4-2 7.1):	
	Hash input (S)	'SMID.1:Prism:06000001:20180120T090000Z:04795CF0B4D74920C64A6879504A6DE9CA788076D946D2F70F8981C01137752C7C050DC6FA61C2CB3D77EFE4275826857CC286805608F43C2E9AC8752600D99FE92CCFB7E146742F0DC9C74CEF6568CBB75AB075D2DFED2EE5531554FA844B8DBC:'
	SHA384 hash output	x'320C265FDC769D3E13D4AD85AF38DE1A33C4BD6CEDEDE61FCA9DCBD3FD77B17F50E185930818CEE921CB160CBBBBB359'
	Fingerprint	'320C265FDC769D3E'
	ID_SM	'SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF'
2.4	Securely store d, Q, and ID_SM	
2.5	Construct PUBKEY_SM-NOSIG	

	ID_SM and Q are given above	
	Expiry	'99991231T115959Z'
	PUBKEY	'PK.ECDH.1|SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF|04795CF0B4D74920C64A6879504A6DE9CA788076D946D2F70F8981C01137752C7C050DC6FA61C2CB3D77EFE4275826857CC286805608F43C2E9AC8752600D99FE92CCFB7E146742F0DC9C74CEF6568CBB75AB075D2DFED2EE5531554FA844B8DBC|99991231T115959Z|||ECD4'
3	Manufacturer signs PUBKEY_SM using Manufacturer ECDSA private key (STS600-4-2 7.2) to create PUBKEY_SM	
	SM MANUFACTURER (Issuer of PUBKEY_SM):	
	PKID Note: 105717ACA4A50852 is the ECDSA fingerprint	'SMMAN.1:Prism:A:20180115T140000Z:105717ACA4A50852:8B04'
	SMMAN.d	x'DA3E238A54D908957A8BD30DD1110A764CB09DBF7FFB753010190F44D172FF7051B562504FFD60C373A1FD22CE0323CF'
	Input message (M)	DFCONCAT('|', rectype, Subject, Q_HEX, Expiry)
	M	'PK.ECDH.1|SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF|04795CF0B4D74920C64A6879504A6DE9CA788076D946D2F70F8981C01137752C7C050DC6FA61C2CB3D77EFE4275826857CC286805608F43C2E9AC8752600D99FE92CCFB7E146742F0DC9C74CEF6568CBB75AB075D2DFED2EE5531554FA844B8DBC|99991231T115959Z|'
	Randomly generated Nonce	x'ABA0F8FAA9A7EEA31390AB846F1E81C85720C99776010170611608D2AA7680B488FCA958053348369A9F60F2852A32A2'
	Take SHA384(M) to get e	x'A7B96FED92F0A26428F6F792E50E109FE1D5D983FA3ED25246A8227BB9A1BBE2DE34F2B79CF179D1996AAE8DD7E8D073'
	Signature	x'8E36C1BBF029875C57985D107DE41293FE2ACC8526C33CD7056AC4F4595F4768569E0560A9C85FC6310F77FC8A0C7E5839B12D5A3498E3AAF1C9E8DEA974B554EC64DCAA546709697D67695770EAE4CF9937CD62E6AB726DB35BD5C7CAD23774'
	PUBKEY	'PK.ECDH.1|SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF|04795CF0B4D74920C64A6879504A6DE9CA788076D946D2F70F8981C01137752C7C050DC6FA61C2CB3D77EFE4275826857CC286805608F43C2E9AC8752600D99FE92CCFB7E146742F0DC9C74CEF6568CBB75AB075D2DFED2EE5531554FA844B8DBC|99991231T115959Z|SMMAN.1:Prism:A:20180115T140000Z:105717ACA4A50852:8B04|8E36C1BBF029875C57985D107DE41293FE2ACC8526C33CD7056AC4F4595F4768569E0560A9C85FC6310F77FC8A0C7E5839B12D5A3498E3AAF1C9E8DEA974B554EC64DCAA546709697D67695770EAE4CF9937CD62E6AB726DB35BD5C7CAD23774|4971'

<table>
<tr><th>4</th><th>Outputs</th><th></th></tr>
<tr><td></td><td>PUBKEY_SM in file-of-records format
Note: the number after the # is a SHA1 hash used as a checksum</td><td>PK.ECDH.1|SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF|04795CF0B4D74920C64A6879504A6DE9CA788076D946D2F70F8981C01137752C7C050DC6FA61C2CB3D77EFE4275826857CC286805608F43C2E9AC8752600D99FE92CCFB7E146742F0DC9C74CEF6568CBB75AB075D2DFED2EE5531554FA844B8DBC|99991231T115959Z|SMMAN.1:Prism:A:20180115T140000Z:105717ACA4A50852:8B04|8E36C1BBF029875C57985D107DE41293FE2ACC8526C33CD7056AC4F4595F4768569E0560A9C85FC6310F77FC8A0C7E5839B12D5A3498E3AAF1C9E8DEA974B554EC64DCAA546709697D67695770EAE4CF9937CD62E6AB726DB35BD5C7CAD23774|4971#03DEF08D021CE970F7A87E2DD999FE7970B67B5F'</td></tr>
</table>

10 KEY AGREEMENT, PHASE 1: SM generates VKLOADREQ

SM Vending Key Load Request (STS600-4-2 Section 11)

(For demonstration purposes we skip zeroisation and load request speed limit enforcement)

STEP	Attribute	Value
1	Parse PUBKEY_KMC	
	Input PUBKEY_KMC	'PK.ECDH.1|KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:4712CFF444570C8A:4C31|044DED24DCA96783C3B240CEEBBB1D69EA36F96F15ACCB13D2EA68B698DDA34443A465E85531904F36F387F5C8908F7DFA4EF8CE0065F6EA5CEC23578EC1C96E4662F2B74184F91A552F9AFB96F99F3EEAFC8C1B5A800857E5B2AC3F0CB2197BD5|20210110T120000Z|||B8F9'
	Parse OK	
	KMC.ID_KMC	'KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:4712CFF444570C8A:4C31'
	KMC.Qoctetstr	x'044DED24DCA96783C3B240CEEBBB1D69EA36F96F15ACCB13D2EA68B698DDA34443A465E85531904F36F387F5C8908F7DFA4EF8CE0065F6EA5CEC23578EC1C96E4662F2B74184F91A552F9AFB96F99F3EEAFC8C1B5A800857E5B2AC3F0CB2197BD5'
	KMC.Expiry	'20210110T120000Z'
	KMC.Issuer	''
	KMC.Signature	x'
	RTC	'20180218T112233Z'
	Key is not expired	Key is not expired
2	Retrieve NIST P-384 domain parameters and check their integrity	
	Domain parameters OK	
3	Obtain KMC.Q (point) from Q_HEX (in PUBKEY_KMC)	
	KMC.Q.x	x'4DED24DCA96783C3B240CEEBBB1D69EA36F96F15ACCB13D2EA68B698DDA34443A465E85531904F36F387F5C8908F7DFA'
	KMC.Q.y	x'4EF8CE0065F6EA5CEC23578EC1C96E4662F2B74184F91A552F9AFB96F99F3EEAFC8C1B5A800857E5B2AC3F0CB2197BD5'
4	Parse ID_KMC and verify fingerprint	
	Parse OK	
	Actual fingerprint from ID_KMC	'4712CFF444570C8A'
	Computed fingerprint (STS600-4-2 7.1):	
	Hash input (S)	'KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:044DED24DCA96783C3B240CEEBBB1D69EA36F96F15ACCB13D2EA68B698DDA34443A465E85531904F36F387F5C8908F7DFA4EF8CE0065F6EA5CEC23578EC1C96E4662F2B

		74184F91A552F9AFB96F99F3EEAFC8C1B5A800857E5B2AC3F0CB2197BD5:'
	SHA384 hash output	'4712CFF444570C8AF67017644733D18E12E4932BB5597608AEBC36147D88DDF9BCADBCD571B5B27BBE4B7AD2FC0D333B'
	Computed fingerprint	'4712CFF444570C8A'
	Fingerprint OK	
5	Retrieve SM values from storage: d, Q, ID	
	SM.ID_SM	'SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF'
	SM.d	x'62EB5B3F0C35325D14C31423717870773F9FD6C767CDD9088013512F3FB08186698F2F2B1298049E944346554664869B'
	SM.Q.x	x'795CF0B4D74920C64A6879504A6DE9CA788076D946D2F70F8981C01137752C7C050DC6FA61C2CB3D77EFE4275826857C'
	SM.Q.y	x'C286805608F43C2E9AC8752600D99FE92CCFB7E146742F0DC9C74CEF6568CBB75AB075D2DFED2EE5531554FA844B8DBC'
	SM.Qoctetstr Q in uncompressed affine coordinates (STS600-4-2 6.5)	x'04795CF0B4D74920C64A6879504A6DE9CA788076D946D2F70F8981C01137752C7C050DC6FA61C2CB3D77EFE4275826857CC286805608F43C2E9AC8752600D99FE92CCFB7E146742F0DC9C74CEF6568CBB75AB075D2DFED2EE5531554FA844B8DBC'
6	Parse ID_SM and verify fingerprint	
	Parse OK	
	Actual fingerprint from ID_SM	'320C265FDC769D3E'
	Computed fingerprint (STS600-4-2 7.1):	
	Hash input (S)	'SMID.1:Prism:06000001:20180120T090000Z:04795CF0B4D74920C64A6879504A6DE9CA788076D946D2F70F8981C01137752C7C050DC6FA61C2CB3D77EFE4275826857CC286805608F43C2E9AC8752600D99FE92CCFB7E146742F0DC9C74CEF6568CBB75AB075D2DFED2EE5531554FA844B8DBC:'
	SHA384 hash output	x'320C265FDC769D3E13D4AD85AF38DE1A33C4BD6CEDEDE61FCA9DCBD3FD77B17F50E185930818CEE921CB160CBBBBB359'
	Computed fingerprint	'320C265FDC769D3E'
	Fingerprint OK	
7	Validate KMC.Q	
	KMC.Q is given above	
		Q is not the point at Infinity
		Q.x and Q.y are in range
		Q is on the curve
		Q has the correct order for a public key (P=nQ is the point at Infinity)
8	Validate SM.Q	
	SM.Q is given above	
		Q is not the point at Infinity
		Q.x and Q.y are in range
		Q is on the curve
		Q has the correct order for a public key (P=nQ is the point at Infinity)

9	Check that SM has the correct value for its private key (SM.Q = SM.d . G)	
	SM.d, SM.Q are given above	
		Private key is in range [1,n-1]
		SM has correct private key
10	Set TVP_KMC to a TIMESTAMP the current time according to the SM's RTC	
	TvpKmc	'20180125T150000Z'
11	Generate an ephemeral key pair (dE, QE) using GENERATE-KEY() in curve P-384 (STS600-4-2 6.8)	
11.1	SM Ephemeral Key (private portion)	Randomly select d in the range 1 to (curve.n - 1)
	dE	x'5CE87AE7BD200159C7671A35C7084724311F883BEF9E04D7826E0208D77622B9038E34BD4259973E49D60EDD3A531043'
11.2	SM Ephemeral Key (public portion)	Compute Q = d.G
	QE.x	x'73E2C294EE44A17A5668ABE67C1F93CBDBCE38DEF4848584C279047A8DDCFFBAE8857C2CCC101A50E4ADB1ECDE9E1473'
	QE.y	x'5B8CBFA88D18BD25F247DF0014298F48BB11CA8415320E7AF7172B0B20D5C00D57D04E33D07343EDE185299CF2CA1E10'
11.3	Set QEStr to Point-to-Octet-String(QE)	QE in uncompressed affine coordinates (per Section 6.5)
	QEStr	x'0473E2C294EE44A17A5668ABE67C1F93CBDBCE38DEF4848584C279047A8DDCFFBAE8857C2CCC101A50E4ADB1ECDE9E14735B8CBFA88D18BD25F247DF0014298F48BB11CA8415320E7AF7172B0B20D5C00D57D04E33D07343EDE185299CF2CA1E10'
12	Set ZE = ECC-CDH(dE, KMC.Q)	
	KMC.Q and dE are given above	
	ZE	x'B08EA35D0CDBD085C22D20C76F2EC65B69F4E562B9392CA5141994464C3C7EC9574477D06AC81F0619C85DAE87E6D359'
13	Set ZS = ECC-CDH(SM.d, KMC.Q)	
	KMC.Q and SM.d are given above	
	ZS	x'2BB3E105662B9241A3190EF60F79C72BC1EF11C1F9E67220375B951CE908DD47F564109CA163C59BA94A3813A79EFEA0
14	Set Z = ZE || ZS	
	Z	x'B08EA35D0CDBD085C22D20C76F2EC65B69F4E562B9392CA5141994464C3C7EC9574477D06AC81F0619C85DAE87E6D3592BB3E105662B9241A3190EF60F79C72BC1EF11C1F9E67220375B951CE908DD47F564109CA163C59BA94A3813A79EFEA0'
15	Construct SharedInfo	LVCONCAT('STS.KAA.1', ID_SM , ID_KMC , TvpKmc)
	ID_SM, ID_KMC, TvpKmc are given above	
	SharedInfo Note: the length values here are shown as readable text, but they should be binary, i.e. "0409" must be passed as 0x04 0x09	x'04095354532E4B41412E313C534D49442E313A507269736D3A30363030303030313A32303138303132305430393030305A3A3332304332363546444337363944433453A38454646494B4D4349442E313A7374732D4B657941677265656D656E742D312E323A5445

		5354313A3230313830313130543132303030305A3A34373132434646343434353730433841 3A3443333110323031383031323554313530303030305A'
16	Set DKM	KDF-X963-SHA-384(Z, SharedInfo, 384)
	Z, SharedInfo are given above	
	Hash input is 'Z || Counter || SharedInfo', where Counter=x'00000001	
	Hash input	x'B08EA35D0CDBD085C22D20C76F2EC65B69F4E562B9392CA5141994464C3C7EC9574477D06AC81F0619C85DAE87E6D3592BB3E105662B9241A3190EF60F79C72BC1EF11C1F9E67220375B951CE908DD47F564109CA163C59BA94A3813A79EFEA00000000104095354532E4B41412E313C534D49442E313A507269736D3A30363030303030313A3230313830313230543039303030305A3A33323043323635464443373639443345 3A38454646494B4D4349442E313A7374732D4B657941677265656D656E742D312E323A54455354313A3230313830313130543132303030305A3A34373132434646343434353730433841 3A34433331103230313830313235543135303030305A'
	SHA384 hash output	x'82334CBC2FC7C893D4A86BE7AAA574F6C0B50F299B44186F99812E6BD366579CC811108E08E614856DE323F9399FE92C'
	DKM	x'82334CBC2FC7C893D4A86BE7AAA574F6C0B50F299B44186F99812E6BD366579CC811108E08E614856DE323F9399FE92C'
17	Set MacKey[192] || KEK[192] = DKM[384]	
	MacKey	x'82334CBC2FC7C893D4A86BE7AAA574F6C0B50F299B44186F'
	KEK	x'99812E6BD366579CC811108E08E614856DE323F9399FE92C'
18	Compute MacTag_SM	
18.1	Construct MacData_SM	LVCONCAT('U_2', ID_SM, ID_KMC, QEStr, TvpKmc, HWID, FWID)
	ID_SM, ID_KMC, QEStr, TvpKmc are given above	
	SM.HWID	'Prism-VSM-1'
	SM.FWID	'STS6-001'
	MacData_SM	x'0703555F323C534D49442E313A507269736D3A303630303030303031 3A3230313830313230543039303030305A3A333230433236354644433736394433453A38454646494B4D4349442E313A7374732D4B657941677265656D656E742D312E323A54455354313A3230313830313130543132303030305A3A3437313243464634343435373043384 13A34433331610473E2C294EE44A17A5668ABE67C1F93CBDBCE38DEF4848584C279047A8DDCFFBAE8857C2CCC101A50E4ADB1ECDE9E14735B8CBFA88D18BD25F247DF0014298F48BB11CA8415320E7AF7172B0B20D5C00D57D04E33D07343EDE185299CF2CA1E1010323031383031323554313530303030305A0B507269736D2D56534D2D3108535453362D303031'
18.2	Compute MacTag_SM	HMAC-SHA-384-192(MacKey, MacData_SM)

<table>
<tr><td></td><td>MacTag_SM</td><td>x'BE6CB4AC631E12EEB5D3F85496042A3274FEAB0477935778'</td></tr>
<tr><td>19</td><td>Compute ExpMacTag_KMC</td><td></td></tr>
<tr><td>19.1</td><td>Construct MacData_KMC</td><td>LVCONCAT('V2', ID_KMC, ID_SM, TvpKmc, QEStr)</td></tr>
<tr><td></td><td>ID_KMC, ID_SM, TvpKmc, QEStr are given above</td><td></td></tr>
<tr><td></td><td>MacData_KMC</td><td>x'05025632494B4D4349442E313A7374732D4B657941677265656D656E742D312E323A54455354313A3230313830313130543132303030305A3A3437313243464634343435373043384134433331C534D49442E313A507269736D3A30363030303030313A323031383031323054303930303030305A3A33323043323635464443373639443345A38454646103230313830313235543135303030305A610473E2C294EE44A17A5668ABE67C1F93CBDBCE38DEF4848584C279047A8DDCFFBAE8857C2CCC101A50E4ADB1ECDE9E14735B8CBFA88D18BD25F247DF0014298F48BB11CA8415320E7AF7172B0B20D5C00D57D04E33D07343EDE185299CF2CA1E10'</td></tr>
<tr><td>19.2</td><td>Compute ExpMacTag_KMC</td><td>HMAC-SHA-384-192(MacKey, MacData_KMC)</td></tr>
<tr><td></td><td>ExpMacTag_KMC</td><td>x'7E6DEC39AFE13B846C59B26EB059186BC521BCAD63718467'</td></tr>
<tr><td>20</td><td>Convert binary strings to hex</td><td></td></tr>
<tr><td>20.1</td><td>Set QEHex = BASE16(QEStr)</td><td></td></tr>
<tr><td></td><td>QEHex</td><td>'0473E2C294EE44A17A5668ABE67C1F93CBDBCE38DEF4848584C279047A8DDCFFBAE8857C2CCC101A50E4ADB1ECDE9E14735B8CBFA88D18BD25F247DF0014298F48BB11CA8415320E7AF7172B0B20D5C00D57D04E33D07343EDE185299CF2CA1E10'</td></tr>
<tr><td>20.2</td><td>Set MacTag_SMHex = BASE16(MacTag_SM)</td><td></td></tr>
<tr><td></td><td>MacTag_SMHex</td><td>'BE6CB4AC631E12EEB5D3F85496042A3274FEAB0477935778'</td></tr>
<tr><td>21</td><td>Construct the Vending Key Load Request (STS600-4-2 7.3):</td><td></td></tr>
<tr><td></td><td>VKLOADREQ</td><td>BUILD-RECORD('VKLOAD.REQ.1', '|', 7, ID_SM, ID_KMC, TvpKmc, HWID, FWID, QEHex, MacTag_SMHex)</td></tr>
<tr><td></td><td>VKLOADREQ</td><td>'VKLOAD.REQ.1|SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF|KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:4712CFF444570C8A:4C31|20180125T150000Z|Prism-VSM-1|STS6-001|0473E2C294EE44A17A5668ABE67C1F93CBDBCE38DEF4848584C279047A8DDCFFBAE8857C2CCC101A50E4ADB1ECDE9E14735B8CBFA88D18BD25F247DF0014298F48BB11CA8415320E7AF7172B0B20D5C00D57D04E33D07343EDE185299CF2CA1E10|BE6CB4AC631E12EEB5D3F85496042A3274FEAB0477935778|F6B3'</td></tr>
<tr><td>22</td><td>Securely store KEK, Fingerprint_KMC, TvpKmc, and ExpMacTag_KMC</td><td></td></tr>
<tr><td>23</td><td>Outputs</td><td></td></tr>
<tr><td></td><td>Record-in-email format</td><td>--STS:VKLOAD.REQ.1 BEGINS--
VKLOAD.REQ.1|SMID.1:Prism:06000001:20180120T</td></tr>
</table>

		090000Z:320C265FDC769D3E:8EFF|KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:47 12CFF444570C8A:4C31|20180125T150000Z|Prism-VSM-1|STS6-001|0473E2 C294EE44A17A5668ABE67C1F93CBDBCE38DEF48 48584C279047A8DDCFFBAE885 7C2CCC101A50E4ADB1ECDE9E14735B8CBFA88D 18BD25F247DF0014298F48BB11 CA8415320E7AF7172B0B20D5C00D57D04E33D073 43EDE185299CF2CA1E10|BE6 CB4AC631E12EEB5D3F85496042A3274FEAB04779 35778|F6B3 --STS:VKLOAD.REQ.1 ENDS--

11 KEY AGREEMENT, PHASE 2: KMC generates VKLOADRESP

KMC Vending Key Load Response (STS600-4-2 Section 12)

(For demonstration purposes we skip audit logging, HWID/FWID checks, PUBKEY expiry checks, and TVP window)

STEP	Attribute/action	Value/result
1	Parse VKLOADREQ	
	Input	
	VKLOADREQ	'VKLOAD.REQ.1|SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF|KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:4712CFF444570C8A:4C31|20180125T150000Z|Prism-VSM-1|STS6-001|0473E2C294EE44A17A5668ABE67C1F93CBDBCE38DEF4848584C279047A8DDCFFBAE8857C2CCC101A50E4ADB1ECDE9E14735B8CBFA88D18BD25F247DF0014298F48BB11CA8415320E7AF7172B0B20D5C00D57D04E33D07343EDE185299CF2CA1E10|BE6CB4AC631E12EEB5D3F85496042A3274FEAB0477935778|F6B3'
	REQ.ID_SM	'SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF'
	REQ.ID_KMC	'KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:4712CFF444570C8A:4C31'
	REQ.TvpKmc	'20180125T150000Z'
	REQ.HWID	'Prism-VSM-1'
	REQ.FWID	'STS6-001'
	REQ.QEHex	'0473E2C294EE44A17A5668ABE67C1F93CBDBCE38DEF4848584C279047A8DDCFFBAE8857C2CCC101A50E4ADB1ECDE9E14735B8CBFA88D18BD25F247DF0014298F48BB11CA8415320E7AF7172B0B20D5C00D57D04E33D07343EDE185299CF2CA1E10'
	REQ.MacTag_SMHex	'BE6CB4AC631E12EEB5D3F85496042A3274FEAB0477935778'
2	Retrieve NIST P-384 domain parameters and check their integrity	Domain parameters OK
3	Retrieve KMC values from storage: d, Q, Expiry, ID	
	KMC.ID_KMC	'KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:4712CFF444570C8A:4C31'
	KMC.Expiry	'20210110T120000Z'
	KMC.d	x'A6531F356BD1DAC52C62ED2DBF3A6FB2CE9CDC06C55D07E93507E90774FE664BCB281C939DE5678F5FB007298D422F50'
	KMC.Q.x	x'4DED24DCA96783C3B240CEEBBB1D69EA36F96F15ACCB13D2EA68B698DDA34443A465E85531904F36F387F5C8908F7DFA'

	KMC.Q.y	x'4EF8CE0065F6EA5CEC23578EC1C96E4662F2B74184F91A552F9AFB96F99F3EEAFC8C1B5A800857E5B2AC3F0CB2197BD5'
	KMC.Qoctetstr Q in uncompressed affine coordinates (STS600-4-2 6.5)	x'044DED24DCA96783C3B240CEEBBB1D69EA36F96F15ACCB13D2EA68B698DDA34443A465E85531904F36F387F5C8908F7DFA4EF8CE0065F6EA5CEC23578EC1C96E4662F2B74184F91A552F9AFB96F99F3EEAFC8C1B5A800857E5B2AC3F0CB2197BD5'
4	Parse REQ.ID_KMC and check target KMC	Parse OK
	REQ.ID_KMC.Manufacturer	'sts-KeyAgreement-1.2'
	REQ.ID_KMC.MID	'TEST1'
	REQ.ID_KMC.GNT	'20180110T120000Z'
	REQ.ID_KMC.Fingerprint	'4712CFF444570C8A'
	VKLOADREQ has been sent to the correct KMC	Ok
	VKLOADREQ has used the correct PUBKEY_KMC	Ok
5	Parse REQ.ID_SM	Parse OK
	REQ.ID_SM.Manufacturer	'Prism'
	REQ.ID_SM.MID	'06000001'
	REQ.ID_SM.GNT	'20180120T090000Z'
	REQ.ID_SM.Fingerprint	'320C265FDC769D3E'
6	Fetch PUBKEY_SM, PUBKEY_MAN, and LastTVP for SM	
	Find PUBKEY_SM for MANUFACTURER='Prism', MID='06000001'	Found OK
	PUBKEY_SM	'PK.ECDH.1|SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF|04795CF0B4D74920C64A6879504A6DE9CA788076D946D2F70F8981C01137752C7C050DC6FA61C2CB3D77EFE4275826857CC286805608F43C2E9AC8752600D99FE92CCFB7E146742F0DC9C74CEF6568CBB75AB075D2DFED2EE5531554FA844B8DBC|99991231T115959Z|SMMAN.1:Prism:A:20180115T140000Z:105717ACA4A50852:8B04|8E36C1BBF029875C57985D107DE41293FE2ACC8526C33CD7056AC4F4595F4768569E0560A9C85FC6310F77FC8A0C7E5839B12D5A3498E3AAF1C9E8DEA974B554EC64DCAA546709697D67695770EAE4CF9937CD62E6AB726DB35BD5C7CAD23774|4971'
	LastTvp	'19930101T000000Z'
7	Parse PUBKEY_SM	
	PUBKEY_SM is given above	Parse OK
	PUBKEY_SM.ID_SM	'SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF'
	PUBKEY_SM.Qoctetstr	x'04795CF0B4D74920C64A6879504A6DE9CA788076D946D2F70F8981C01137752C7C050DC6FA61C2CB3D77EFE4275826857CC286805608F43C2E9AC8752600D99FE92CCFB7E146742F0DC9C74CEF6568CBB75AB075D2DFED2EE5531554FA844B8DBC'
	PUBKEY_SM.Expiry	'99991231T115959Z'
	PUBKEY_SM.Issuer	'SMMAN.1:Prism:A:20180115T140000Z:105717ACA4A50852:8B04'

	PUBKEY_SM.Signature	x'8E36C1BBF029875C57985D107DE41293FE2ACC8526C33CD7056AC4F4595F4768569E0560A9C85FC6310F77FC8A0C7E5839B12D5A3498E3AAF1C9E8DEA974B554EC64DCAA546709697D67695770EAE4CF9937CD62E6AB726DB35BD5C7CAD23774'
8	Check that KMC has PUBKEY_SM used in VKLOADREQ PUBKEY_SM, REQ.ID_SM are given above	KMC has PUBKEY_SM used in VKLOADREQ
9	Check that PUBKEY_MAN is available	
	Find PUBKEY_MAN for Issuer 'SMMAN.1:Prism:A:20180115T140000Z:105717ACA4A50852:8B04'	Found OK
	PUBKEY_MAN	'PK.ECDSA.1|SMMAN.1:Prism:A:20180115T140000Z:105717ACA4A50852:8B04|04ECDC178D70C8B495ED5F5A05A68393D710A6BB9DACDEDAD28EA5886E5AE683FF287A26FD23ABFBF20682F218DF76D724CD56C1E85A19F65C5DD7AAA590F342D95503CDE643AB7E0691B9725BABEEF15C7D26857A0416D96F61682FA83A14991A|20210115T140000Z|SMMAN.1:Prism:A:20180115T140000Z:105717ACA4A50852:8B04|5F82E6B0FB0837F383D3E1E7D7061CC0A42DDA3530DB68E21F03F185271D85A46A9D4369FC1507B233C1CAFDA61D01020D6A649289CFBC05F919E3EFCBCDF8CFFFD756AB9B3DD63D66C3ED6D99BE3184124CDECA5A59FC7A14316A2DD0265AA4|1F11'
10	Check for replay (REQ.TvpKmc is fresh compared to LastTvp) REQ.TvpKmc, LastTvp are given above	REQ.TvpKmc is fresh
11	Check that TvpKmc is recent (compared to KMC system clock) RTC='20180218T112233Z' REQ.TvpKmc is given above	TvpKmc is recent
12	Set QE to Octet-String-to-Point(BASE16-DECODE(QEHex)) REQ.QEHex is given above	
	QEStr	x'0473E2C294EE44A17A5668ABE67C1F93CBDBCE38DEF4848584C279047A8DDCFFBAE8857C2CCC101A50E4ADB1ECDE9E14735B8CBFA88D18BD25F247DF0014298F48BB11CA8415320E7AF7172B0B20D5C00D57D04E33D07343EDE185299CF2CA1E10'
	QE.x	x'73E2C294EE44A17A5668ABE67C1F93CBDBCE38DEF4848584C279047A8DDCFFBAE8857C2CCC101A50E4ADB1ECDE9E1473'
	QE.y	x'5B8CBFA88D18BD25F247DF0014298F48BB11CA8415320E7AF7172B0B20D5C00D57D04E33D07343EDE185299CF2CA1E10'
13.1	Parse PUBKEY_MAN PUBKEY_MAN is given above	Parse OK
	PUBKEY_MAN.ID_SMMAN	'SMMAN.1:Prism:A:20180115T140000Z:105717ACA4A50852:8B04'
	PUBKEY_MAN.Qoctetstr	x'04ECDC178D70C8B495ED5F5A05A68393D710A6

		BB9DACDEDAD28EA5886E5AE683FF287A26FD23ABFBF20682F218DF76D724CD56C1E85A19F65C5DD7AAA590F342D95503CDE643AB7E0691B9725BABEEF15C7D26857A0416D96F61682FA83A14991A'
	PUBKEY_MAN.Expiry	'20210115T140000Z'
	PUBKEY_MAN.Issuer	'SMMAN.1:Prism:A:20180115T140000Z:105717ACA4A50852:8B04'
	PUBKEY_MAN.Signature	x'5F82E6B0FB0837F383D3E1E7D7061CC0A42DDA3530DB68E21F03F185271D85A46A9D4369FC1507B233C1CAFDA61D01020D6A649289CFBC05F919E3EFCBCDF8CFFFD756AB9B3DD63D66C3ED6D99BE3184124CDECA5A59FC7A14316A2DD0265AA4'
13.2	Obtain PUBKEY_MAN.Q (point) from Q_HEX (in PUBKEY_MAN)	
	PUBKEY_MAN.Q.x	x'ECDC178D70C8B495ED5F5A05A68393D710A6BB9DACDEDAD28EA5886E5AE683FF287A26FD23ABFBF20682F218DF76D724'
	PUBKEY_MAN.Q.y	x'CD56C1E85A19F65C5DD7AAA590F342D95503CDE643AB7E0691B9725BABEEF15C7D26857A0416D96F61682FA83A14991A'
13.3	Parse ID_SMMAN and verify fingerprint	Parse OK
	Actual fingerprint from ID_SMMAN	'105717ACA4A50852'
	Computed fingerprint (STS600-4-2 7.1):	
	Hash input (S)	'SMMAN.1:Prism:A:20180115T140000Z:04ECDC178D70C8B495ED5F5A05A68393D710A6BB9DACDEDAD28EA5886E5AE683FF287A26FD23ABFBF20682F218DF76D724CD56C1E85A19F65C5DD7AAA590F342D95503CDE643AB7E0691B9725BABEEF15C7D26857A0416D96F61682FA83A14991A:'
	SHA384 hash output	x'105717ACA4A508529DF381AE435D8CAD43315CECB4C81A3C945DBFCE01AC6851D0F763AD63149012336A0FF2C5F4F0A0'
	Computed fingerprint	'105717ACA4A50852'
	Fingerprint OK	
14.1	Parse PUBKEY_SM	already done above
14.2	Obtain PUBKEY_SM.Q (point) from Q_HEX (in PUBKEY_MAN)	
	PUBKEY_SM.Q.x	x'795CF0B4D74920C64A6879504A6DE9CA788076D946D2F70F8981C01137752C7C050DC6FA61C2CB3D77EFE4275826857C'
	PUBKEY_SM.Q.y	x'C286805608F43C2E9AC8752600D99FE92CCFB7E146742F0DC9C74CEF6568CBB75AB075D2DFED2EE5531554FA844B8DBC'
14.3	Parse ID_SM and verify fingerprint	Parse OK
	ID_SM.GNT	'20180120T090000Z'
	Actual fingerprint from ID_SM	'320C265FDC769D3E'
	Computed fingerprint (STS600-4-2 7.1):	
	Hash input (S)	'SMID.1:Prism:06000001:20180120T090000Z:04795CF0B4D74920C64A6879504A6DE9CA788076D946D2F70F8981C01137752C7C050DC6FA61C2CB3D77EFE4275826857CC286805608F43C2E9AC8752600D99FE92CCFB7E146742F0DC9C74CEF6568CBB75AB0

		75D2DFED2EE5531554FA844B8DBC:'
	SHA384 hash output	x'320C265FDC769D3E13D4AD85AF38DE1A33C4BD6CEDEDE61FCA9DCBD3FD77B17F50E185930818CEE921CB160CBBBBB359'
	Computed fingerprint	'320C265FDC769D3E'
	Fingerprint OK	
14.4	Is PUBKEY_SM issued by PUBKEY_MAN?	PUBKEY_SM.Issuer, ID_SM.GNT, PUBKEY_MAN.ID_SMMAN, PUBKEY_MAN.Expiry are given above
	Got Issuer PUBKEY for PUBKEY_SM	PUBKEY_SM generated before Issuer expiry
14.5	Is PUBKEY_SM expired? RTC, PUBKEY_SM.Expiry are given above	Key is not expired
15	Is PUBKEY_KMC expired? RTC, PUBKEY_SM.Expiry are given above	Key is not expired
16	Parse ID_KMC and verify fingerprint	Parse OK
	ID_KMC.GNT	'20180110T120000Z'
	Actual fingerprint from ID_KMC	'4712CFF444570C8A'
	Computed fingerprint (STS600-4-2 7.1):	
	Hash input (S)	'KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:044DED24DCA96783C3B240CEEBBB1D69EA36F96F15ACCB13D2EA68B698DDA34443A465E85531904F36F387F5C8908F7DFA4EF8CE0065F6EA5CEC23578EC1C96E4662F2B74184F91A552F9AFB96F99F3EEAFC8C1B5A800857E5B2AC3F0CB2197BD5:'
	SHA384 hash output	x'4712CFF444570C8AF67017644733D18E12E4932BB5597608AEBC36147D88DDF9BCADBCD571B5B27BBE4B7AD2FC0D333B'
	Computed fingerprint	'4712CFF444570C8A'
	Fingerprint OK	
17	Validate PUBKEY_MAN.Q PUBKEY_MAN.Q is given above	
		Q is not the point at Infinity
		Q.x and Q.y are in range
		Q is on the curve
		Q has the correct order for a public key (P=nQ is the point at Infinity)
18	Verify signature of PUBKEY_SM (STS600-4-2 7.2) PUBKEY_SM, PUBKEY_MAN.Q are given above	
	Input message M	DFCONCAT('|', rectype, Subject, Q_HEX, Expiry)
	M	'PK.ECDH.1|SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF|04795CF0B4D74920C64A6879504A6DE9CA788076D946D2F70F8981C01137752C7C050DC6FA61C2CB3D77EFE4275826857CC286805608F43C2E9AC8752600D99FE92CCFB7E146742F0DC9C74CEF6568CBB75AB075D2DFED2EE5531554FA844B8DBC|99991231T115959Z|'
		Signature OK
19	Validate PUBKEY_SM.Q	

	PUBKEY_SM.Q is given above	
		Q is not the point at Infinity
		Q.x and Q.y are in range
		Q is on the curve
		Q has the correct order for a public key (P=nQ is the point at Infinity)
20	Validate KMC.Q KMC.Q is given above	
		Q is not the point at Infinity
		Q.x and Q.y are in range
		Q is on the curve
		Q has the correct order for a public key (P=nQ is the point at Infinity)
21	Check that KMC has the correct value for its private key (KMC.Q = KMC.d . G) KMC.d, KMC.Q are given above	
		Private key is in range [1,n-1]
		KMC has correct private key
22	Validate ephemeral key QE QE is given above	
		Q is not the point at Infinity
		Q.x and Q.y are in range
		Q is on the curve
		Q has the correct order for a public key (P=nQ is the point at Infinity)
23	Set ZE = ECC-CDH(KMC.d, QE) KMC.d and QE are given above	
	ZE	x'B08EA35D0CDBD085C22D20C76F2EC65B69F4E562B9392CA5141994464C3C7EC9574477D06AC81F0619C85DAE87E6D359'
24	Set ZS = ECC-CDH(KMC.d, SM.Q) KMC.d and PUBKEY_SM.Q are given above	
	ZS	x'2BB3E105662B9241A3190EF60F79C72BC1EF11C1F9E67220375B951CE908DD47F564109CA163C59BA94A3813A79EFEA0'
25	Set Z = ZE || ZS	
	Z	x'B08EA35D0CDBD085C22D20C76F2EC65B69F4E562B9392CA5141994464C3C7EC9574477D06AC81F0619C85DAE87E6D3592BB3E105662B9241A3190EF60F79C72BC1EF11C1F9E67220375B951CE908DD47F564109CA163C59BA94A3813A79EFEA0'
26	Construct SharedInfo PUBKEY_SM.ID_SM, KMC.ID_KMC, REQ.TvpKmc are given above	LVCONCAT('STS.KAA.1', ID_SM , ID_KMC , TvpKmc)
	SharedInfo	x'04095354532E4B41412E313C534D49442E313A507269736D3A3036303030303030313A3230313830313230543039303030305A3A333230433236354644433736394433453A38454646494B4D4349442E313A7374732D4B6579416772656656D656E742D312E323A54455354313A32303138303131305431323030303030305A3A34373132434646343434353730433841 3A3443333110323031383031323554313530303030305A'

27	Set DKM Z, SharedInfo are given above	KDF-X963-SHA-384(Z, SharedInfo, 384)
	Hash input	'Z || Counter || SharedInfo', where Counter=x'00000001
	Hash input (S)	x'B08EA35D0CDBD085C22D20C76F2EC65B69F4E562B9392CA5141994464C3C7EC9574477D06AC81F0619C85DAE87E6D3592BB3E105662B9241A3190EF60F79C72BC1EF11C1F9E67220375B951CE908DD47F564109CA163C59BA94A3813A79EFEA00000000104095354532E4B41412E313C534D49442E313A507269736D3A3036303030303030313A3230313830313230543039303030305A3A333230433236354644433736394433453A38454646494B4D4349442E313A7374732D4B657941677265656D656E742D312E323A54455354313A3230313830313130543132303030305A3A3437313243464634343435373043384134A34433331103230313830313235543135303030305A'
	SHA384 hash output	x'82334CBC2FC7C893D4A86BE7AAA574F6C0B50F299B44186F99812E6BD366579CC811108E08E614856DE323F9399FE92C'
28	Set MacKey[192] || KEK[192]	DKM[384]
	MacKey	x'82334CBC2FC7C893D4A86BE7AAA574F6C0B50F299B44186F'
	KEK	x'99812E6BD366579CC811108E08E614856DE323F9399FE92C
29	Compute ExpMacTag_SM	
29.1	Construct MacData_SM PUBKEY_SM.ID_SM, KMC.ID_KMC, QEStr, TvpKmc, REQ.HWID, REQ.FWID are given above	LVCONCAT('U_2', ID_SM, ID_KMC, QEStr, TvpKmc, HWID, FWID)
	MacData_SM	x'0703555F323C534D49442E313A507269736D3A3036303030303030313A3230313830313230543039303030305A3A333230433236354644433736394433453A38454646494B4D4349442E313A7374732D4B657941677265656D656E742D312E323A54455354313A3230313830313130543132303030305A3A34373132434646343434353730433841333A34433331610473E2C294EE44A17A5668ABE67C1F93CBDBCE38DEF4848584C279047A8DDCFFBAE8857C2CCC101A50E4ADB1ECDE9E14735B8CBFA88D18BD25F247DF0014298F48BB11CA8415320E7AF7172B0B20D5C00D57D04E33D07343EDE185299CF2CA1E10103230313830313235543135303030305A0B507269736D2D56534D2D2D3108535453362D303031'
29.2	Compute ExpMacTag_SM	HMAC-SHA-384-192(MacKey, MacData_SM)
	ExpMacTag_SM	x'BE6CB4AC631E12EEB5D3F85496042A3274FEAB0477935778'
30	Check ExpMacTag_SM ExpMacTag_SM, REQ.MacTag_SMHex are given above	ExpMacTag_SM OK => VKLOADREQ is authentic!
31	Compute MacTag_KMC	
31.1	Construct MacData_KMC	LVCONCAT('V2', ID_KMC, ID_SM, TvpKmc, QEStr)

<table>
<tr><td></td><td>KMC.ID_KMC, PUBKEY_SM.ID_SM, TvpKmc, QEStr are given above</td><td></td></tr>
<tr><td></td><td>MacData_KMC</td><td>x'05025632494B4D4349442E313A7374732D4B657941677265656D656E742D312E323A54455354313A32303138303131305431323030305A3A343731324346463434343537304338413A344333313C534D49442E313A507269736D3A30363030303030313A32303138303132305430393030305A3A333230433236354644433736394433453A38454646103230313830313235543135303030305A610473E2C294EE44A17A5668ABE67C1F93CBDBCE38DEF4848584C279047A8DDCFFBAE8857C2CCC101A50E4ADB1ECDE9E14735B8CBFA88D18BD25F247DF0014298F48BB11CA8415320E7AF7172B0B20D5C00D57D04E33D07343EDE185299CF2CA1E10'</td></tr>
<tr><td>31.2</td><td>Compute MacTag_KMC</td><td>HMAC-SHA-384-192(MacKey, MacData_KMC)</td></tr>
<tr><td></td><td>MacTag_KMC</td><td>x'7E6DEC39AFE13B846C59B26EB059186BC521BCAD63718467'</td></tr>
<tr><td>31.3</td><td>Convert MacTag_KMC to hex</td><td></td></tr>
<tr><td></td><td>MacTag_KMCHex</td><td>'7E6DEC39AFE13B846C59B26EB059186BC521BCAD63718467'</td></tr>
<tr><td>32</td><td>Construct the Vending Key Load Response (STS600-4-2 7.4)</td><td></td></tr>
<tr><td></td><td>VKLOADRESP</td><td>BUILD-RECORD('VKLOAD.RESP.1', '|', 4, ID_KMC, ID_SM, TvpKmc, MacTag_KMCHex)</td></tr>
<tr><td></td><td>VKLOADRESP</td><td>'VKLOAD.RESP.1|KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:4712CFF444570C8A:4C31|SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF|20180125T150000Z|7E6DEC39AFE13B846C59B26EB059186BC521BCAD63718467|D743'</td></tr>
<tr><td>33</td><td>Securely store KEK, TvpKmc (as LastTvp for MANUFACTURER,MID)</td><td></td></tr>
</table>

12 KEY AGREEMENT, PHASE 3: KMC creates Key Load File

KMC Vending Key Load Response (Section 12)

(For demonstration purposes we use a simple counter for the Nonce, this meets the STS600-4-2 requirements although we recommend a random nonce for each WRAPPED-KEY)

STEP	Attribute/action	Value/result
1	Prerequisites	
	KEK	x'99812E6BD366579CC811108E08E614856DE323F9399FE92C'
2.1	Build a WRAPPED-KEY:	
	VendingKey	x'ABABABABABABABAB'
	Nonce	x'000000000000000000000001
	Attributes associated with VendingKey:	
	ACT	19930101T000000Z
	BDT	19930101T000000Z
	DKG	02
	KEN	255
	KRN	1
	KTC	2
	SGC	0000123456
	Attributes (STS600-4-2 7.5.1 card format), note ascending order of names	'ACT19930101T000000Z;BDT19930101T000000Z;DKG02;KEN255;KRN1;KTC2;SGC0000123456;'
	ProtectedKey	BASE16(AES-CCM(KEK, Nonce, Attributes, VendingKey)
	ProtectedKey	'D80D0BA61492E51E2AFE96FC69633DB5BE92932DEAECEA6C'
	WRAPPED-KEY	'KEY.1|000000000000000000000001|ACT19930101T000000Z;BDT19930101T000000Z;DKG02;KEN255;KRN1;KTC2;SGC0000123456;|D80D0BA61492E51E2AFE96FC69633DB5BE92932DEAECEA6C|4726'
	KCV	x'
2.2	Build a WRAPPED-KEY	
	VendingKey	x'ABABABABABABABAB94949494949494940123456 7'
	Nonce	x'000000000000000000000002
	Attributes associated with VendingKey:	
	ACT	20140101T000000Z
	BDT	20140101T000000Z
	CLM	5368D4A5
	CLU	0
	DKG	04
	EXP	20990101T000000Z
	IUT	20990101T000000Z
	KEN	255
	KRN	4
	KTC	2
	SBM	FFFF
	SGC	0000123457
	SGN	CTS 123457,4 VUDK BDT14 DKG04 AB.94.0-7

	ULM	1000000
	Attributes (STS600-4-2 7.5.1 card format), note ascending order of names:	'ACT20140101T000000Z;BDT20140101T000000Z;CLM5368D4A5;CLU0;DKG04;EXP20990101T000000Z;IUT20990101T000000Z;KEN255;KRN4;KTC2;SBMFFFF;SGC0000123457;SGNCTS 123457,4 VUDK BDT14 DKG04 AB.94.0-7;ULM1000000;'
	ProtectedKey	BASE16(AES-CCM(KEK, Nonce, Attributes, VendingKey)
	ProtectedKey	'ECC3BE7DD9F8D700BFE717EB9154C1BFD748BAB4BD2640DD89DD68B8E0BD1A74A8F72C9F'
	WRAPPED-KEY	'KEY.1|000000000000000000000002|ACT20140101T000000Z;BDT20140101T000000Z;CLM5368D4A5;CLU0;DKG04;EXP20990101T000000Z;IUT20990101T000000Z;KEN255;KRN4;KTC2;SBMFFFF;SGC0000123457;SGNCTS 123457,4 VUDK BDT14 DKG04 AB.94.0-7;ULM1000000;|ECC3BE7DD9F8D700BFE717EB9154C1BFD748BAB4BD2640DD89DD68B8E0BD1A74A8F72C9F|5AF9'
	KCV	x'
4	Outputs	IDKMC|IDSM|TVPkmc|MACTAGKMCHex|
	Key Load File (file-of-records format):	VKLOAD.RESP.1|KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:4712CFF444570C8A:4C31|SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF|20180125T150000Z|7E6DEC39AFE13B846C59B26EB059186BC521BCAD63718467|D743 KEY.1|000000000000000000000001|ACT19930101T000000Z;BDT19930101T000000Z;DKG02;KEN255;KRN1;KTC2;SGC0000123456;|D80D0BA61492E51E2AFE96FC69633DB5BE92932DEAECEA6C|4726 KEY.1|000000000000000000000002|ACT20140101T000000Z;BDT20140101T000000Z;CLM5368D4A5;CLU0;DKG04;EXP20990101T000000Z;IUT20990101T000000Z;KEN255;KRN4;KTC2;SBMFFFF;SGC0000123457;SGNCTS 123457,4 VUDK BDT14 DKG04 AB.94.0-7;ULM1000000;|ECC3BE7DD9F8D700BFE717EB9154C1BFD748BAB4BD2640DD89DD68B8E0BD1A74A8F72C9F|5AF9 #17123400EA6BF8B6B01806DF883CE740F8C11693

13 KEY AGREEMENT, PHASE 4: SM loads KLF

SM KEK confirmation and Vending Key Import (section 13)

STEP	Attribute/action	Value/result
1	Parse Key Load File	
	Check file structure	File structure OK
	Key Load File (file-of-records format): ***VKLOADRESP*** Wrapped key 1 Wrapped key 2 Sha1 Checksum of the entire file	 ***VKLOAD.RESP.1|KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:4712CFF444570C8A:4C31|SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF|20180125T150000Z|7E6DEC39AFE13B846C59B26EB059186BC521BCAD63718467|D743*** KEY.1|0000000000000000000000001|ACT19930101T000000Z;BDT19930101T000000Z;DKG02;KEN255;KRN1;KTC2;SGC0000123456;|D80D0BA61492E51E2AFE96FC69633DB5BE92932DEAECEA6C|4726 KEY.1|0000000000000000000000002|ACT20140101T000000Z;BDT20140101T000000Z;CLM5368D4A5;CLU0;DKG04;EXP20990101T000000Z;IUT20990101T000000Z;KEN255;KRN4;KTC2;SBMFFFF;SGC0000123457;SGNCTS 123457,4 VUDK BDT14 DKG04 AB.94.0-7;ULM1000000;|ECC3BE7DD9F8D700BFE717EB9154C1BFD748BAB4BD2640DD89DD68B8E0BD1A74A8F72C9F|5AF9 #17123400EA6BF8B6B01806DF883CE740F8C11693
	Parse KLF and check checksum Note: this SHA1 checksum check is not done by the HSM - it is done by the POS software since only the portion in bold italics is sent to the HSM!	Checksum OK
2	Check VKLOADRESP	
a	Retrieve SM state from storage	
	SM.ID_SM	'SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF'
	SM.TvpKmc	'20180125T150000Z'
	SM.KmcFingerprint	'4712CFF444570C8A'
	SM.ExpMacTagHex	'7E6DEC39AFE13B846C59B26EB059186BC521BCAD63718467'
b	Check age of key agreement session	
	RTC (These must be made real time at some stage)	'20180218T112233Z'
		Key is not expired
c	Parse VKLOADRESP	
	Input VKLOADRESP	'VKLOAD.RESP.1|KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:4712CFF444570C8A:

		4C31|SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF|20180125T150000Z|7E6DEC39AFE13B846C59B26EB059186BC521BCAD63718467|D743'
		Parse OK
	RESP.ID_KMC	'KMCID.1:sts-KeyAgreement-1.2:TEST1:20180110T120000Z:4712CFF444570C8A:4C31'
	RESP.ID_SM	'SMID.1:Prism:06000001:20180120T090000Z:320C265FDC769D3E:8EFF'
	RESP.TvpKmc	'20180125T150000Z'
	RESP.MacTag_KMCHex	'7E6DEC39AFE13B846C59B26EB059186BC521BCAD63718467'
c	Parse ID_KMC RESP.ID_KMC is given above	Parse OK
d	Check that VKLOADRESP is for this SM SM.ID_SM, RESP.ID_SM are given above	VKLOADRESP.ID_SM matches this SM
	Check that VKLOADRESP is for the current Key Agreement session SM.KmcFingerprint, ID_KMC.Fingerprint, SM.TvpKmc, RESP.TvpKmc are given above	VKLOADRESP matches SM key agreement session
	Check key confirmation SM.ExpMacTagHex, RESP.MacTag_KMCHex are given above	VKLOADRESP key confirmation OK -> KEK may now be used
3	Unwrap WRAPPED-KEY records:	
3.1	Unwrap WRAPPED-KEY, line 2	
	WRAPPED-KEY	'KEY.1|000000000000000000000001|ACT19930101T000000Z;BDT19930101T000000Z;DKG02;KEN255;KRN1;KTC2;SGC0000123456;|D80D0BA61492E51E2AFE96FC69633DB5BE92932DEAECEA6C|4726'
	Nonce	'000000000000000000000001'
	Attributes	'ACT19930101T000000Z;BDT19930101T000000Z;DKG02;KEN255;KRN1;KTC2;SGC0000123456;'
	ProtectedKey	'D80D0BA61492E51E2AFE96FC69633DB5BE92932DEAECEA6C'
		Nonce not previously seen in this KLF, OK
		Key unwrapped, OK
	Attributes	'ACT 19930101T000000Z BDT 19930101T000000Z DKG 02 KEN 255 KRN 1 KTC 2 SGC 0000123456 KCV C33F45'
	VendingKey	x'ABABABABABABABAB
3.2	Unwrap WRAPPED-KEY, line 3	
	WRAPPED-KEY	'KEY.1|000000000000000000000002|ACT20140101T000000Z;BDT20140101T000000Z;CLM5368D4A5;CLU0;DKG04;EXP20990101T000000Z;IUT20990101T000000Z;KEN255;KRN4;KTC2;SBMFFFF;SGC0000123457;SGNCTS 123457,4 VUDK BDT14 DKG04

		AB.94.0-7;ULM1000000;|ECC3BE7DD9F8D700BFE717EB9154C1BFD748BAB4BD2640DD89DD68B8E0BD1A74A8F72C9F|5AF9'
	Nonce	'000000000000000000000002'
	Attributes	'ACT20140101T000000Z;BDT20140101T000000Z;CLM5368D4A5;CLU0;DKG04;EXP20990101T000000Z;IUT20990101T000000Z;KEN255;KRN4;KTC2;SBMFFFF;SGC0000123457;SGNCTS 123457,4 VUDK BDT14 DKG04 AB.94.0-7;ULM1000000;'
	ProtectedKey	'ECC3BE7DD9F8D700BFE717EB9154C1BFD748BAB4BD2640DD89DD68B8E0BD1A74A8F72C9F'
		Nonce not previously seen in this KLF, OK
		Key unwrapped, OK
	Attributes	'ACT 20140101T000000Z BDT 20140101T000000Z CLM 5368D4A5 CLU 0 DKG 04 EXP 20990101T000000Z IUT 20990101T000000Z KEN 255 KRN 4 KTC 2 SBM FFFF SGC 0000123457 SGN {CTS 123457,4 VUDK BDT14 DKG04 AB.94.0-7} ULM 1000000 KCV 6C3DE4'
	VendingKey	x'ABABABABABABABAB949494949494949401234567