Elektrobit

EB zentur HSM Firmware v1.16

Product description

Document date: 2023-06-30

Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany
Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333
Email: info.automotive@elektrobit.com

Technical support

https://www.elektrobit.com/support

Legal disclaimer

Confidential information.

Table of Contents

1. Important notice

This product description is provided by Elektrobit Automotive GmbH, hereinafter referred to as Elektrobit. It is legally binding if it is declared and agreed in a quotation. The content of this document is subject to change according to the Elektrobit product strategy.

This document provides a general description of EB zentur HSM Firmware. Its content is bound to the content of the corresponding release.

All images are only examples to show how EB zentur HSM Firmware may look like. The actual implementation of EB zentur HSM Firmware can differ.

2. Safe and correct use of EB zentur products

2.1. Intended usage of EB zentur products

EB zentur products are intended to be used in automotive projects based on AUTOSAR. For more information on the AUTOSAR consortium, see www.autosar.org.

2.2. Possible misuse of EB zentur products

WARNING **Possible misuse and liability**

- You may use the software only in accordance with the intended usage and as permitted in the applicable license terms and agreements. Elektrobit Automotive GmbH assumes no liability and cannot be held responsible for any use of software not in compliance with the applicable license terms and agreements.
- If you use the product in applications that are not defined by the AUTOSAR consortium, the product and its technology may not conform to the requirements of your application. Elektrobit Automotive GmbH is not liable for such misuse.
- Use of this product without taking appropriate risk-reduction measures throughout the entire development phase can result in unexpected behavior. Elektrobit Automotive GmbH is not liable for this misuse. To find out about risk-reduction measures, see the **EB tresos maintenance and support annex**. You have received this document together with your product quote.

2.3. Target group and required knowledge

- Basic software engineers
- Application developers
- Programming skills and experience in programming AUTOSAR-compliant ECUs

3. Introduction

EB zentur HSM Firmware is part of the EB zentur product line. The EB zentur product line offers microcontroller-specific software to enable the hardware security features and abstract to the higher level basic software.

EB zentur HSM Firmware is a performance- and resource-optimized solution for hardware security modules to access cryptographic hardware accelerators or provide software implementation for selected algorithms. It also enables other security-critical features like secure boot and firmware update. EB zentur HSM Firmware can be integrated into various operating systems.

The communication between host and HSM is encapsulated via either EB zentur HSM CrySHE or EB zentur HSM Crypto Driver Hardware. With these modules, you can easily integrate EB zentur HSM Firmware into the ACG8 environment.

It is possible to integrate EB zentur HSM Firmware into the EB tresos AutoCore Generic 8 Crypto and Security Stack.

4. Product details

4.1. Standard packages overview

EB zentur HSM Firmware is available in the standard packages Basic and Evita with the following features:

- **Basic**
  - SHE, SHE+
  - Host secure boot
  - Symmetric encryption and decryption
  - MAC generation
  - Hash generation
  - Key management
  - Up to 20 symmetric keys supported
- **Evita**
  - Basic feature package as described above
  - Certificate management
  - HSM Firmware update
  - Secure boot
  - Life cycle management
  - Asymmetric cryptography support
  - Signature verification incl.
    - Signature generation
    - Asymmetric key load
  - Up to 50 symmetric keys supported

4.2. Extension packages overview

EB zentur HSM Firmware is available in an extension package with features as follows:

- **Evita OEM extension - VW**

- Evita feature package + VKMS 2.1

  For a detailed feature list, see Appendix A, “Supported features by hardware platform“.

  For details on VKMS 2.1, see Appendix B, “Supported features compliant with the VKMS 2.1 specification“.

4.3. Supported features

EB zentur HSM Firmware supports a variety of features following the SHE, SHE+ and EVITA standards and more. In addition, EB zentur HSM Firmware supports OEM-specific extensions. The availability of a feature can vary depending on the release and hardware capabilities. EB zentur HSM Firmware utilizes cryptographic hardware resources when available on the chipset.

4.3.1. Cryptographic features

Table 4.1, “Cryptographic services” lists all cryptographic services supported by EB zentur HSM Firmware.

Cryptographic service	Description
RNG initialization	Initialize the seed and RNG and derive a key for the PRNG
Generate random number	Generates a vector of 128 random bits
Symmetric encryption	Encrypt a given plain text with a given key and return a ciphered text
Symmetric decryption	Decrypt a given ciphered text with a given key and return a plain text
Symmetric block encryption, ECB and CBC	Encrypt a given plain text (which can have a multiple length of block width 128bit) with a given key and return a ciphered text
Symmetric block decryption, ECB and CBC	Decrypt a given ciphered text with a given key and return a plain text (which can have a multiple length of block width 128bit)
Key generation	Generate a NIST ECC P-256 key pair
MAC generation	Generate a MAC of a given message with the help of a given key
MAC verification	Verify a MAC of a given message with the help of a given key against a provided MAC
Secure key update	Update/load an internal/protected/ciphered key
Plain key update	A given key is loaded (without encryption and verification of the key), so the key is handed over in plain text. The plain key can only be loaded into the RAM KEY slot.

Cryptographic service	Description
Export key	Export the RAM KEY into a format protected by SECRET KEY. The key can be imported again via a secure key update.
Export public key	Export the EC public key into a raw format. The key can only be exported. The key cannot be modified after its creation because we currently support only one key-slot for the ECC NIST P-256 key pair (generated via hardware accelerator).
Hashing	SHA-2 256 bits
Digital signatures	Generate or verify digital signatures using asymmetric cipher suite

Table 4.1. Cryptographic services

4.3.2. Cryptographic algorithms

Depending on the support of the target device and hardware capabilities, cryptographic algorithms may be configured to use the available hardware accelerators. Additional cryptographic algorithms may be integrated into the EB zentur HSM Firmware as software-based algorithms. The following cryptographic algorithms are supported:

- With use of hardware accelerators:
  - Cipher-based message authentication code (CMAC) generation and verification, AES-128 based
  - Symmetric encryption and decryption of data, supporting electronic cipher book mode (ECB) and cipher block chaining (CBC) using AES-128
  - Pseudo random number generation (PRNG) of 128 bits, including generation of a random seed through a true random number generator (TRNG) and random seed extension
  - HASH generation SHA2-256
  - Digital signature generation and verification using ECDSA NIST curve P-256 (secp256r1)
- With use of software:
  - Digital signature generation using
    - RSASSA-PKCS1-v1_5
    - Ed25519ph curve
  - Digital signature verification using
    - Ed25519ph curve
    - RSASSA-PSS
    - RSASSA-PKCS1-v1_5
  - MAC generation

- SipHash-2-4 and SipHash-4-8 families

4.3.3. Key management

- Key management of symmetric 128-bit AES-based keys (see Section 4.4.2, “Key types and supported keys”)
  - 1 secret key slot
  - 1 master ECU key slot
  - 1 boot MAC key slot
  - 1 boot MAC slot
  - 1 RAM key slot
  - 20 NvM key slots (or up to 50 NvM key slots with TC3xx)
  - Ciphered key container load according to SHE specification v1.1
- Key management of asymmetric keys with a maximum storage space of 800 bytes (1200 bytes for RSA)
  - ECDSA and EdDSA private keys
  - ECC public keys
  - RSA private and public keys
- Additional software features for strong key management
  - Loading a plain key into RAM key slot
  - Loading ciphered keys (in: M1-M3, out: M4-M5) into RAM key slot and non-volatile key slots
  - Exporting RAM key as ciphered key container
  - SHE command CMD_DEBUG with the restriction that it can only be used to delete all keys stored in the flash memory

4.3.4. Certificate management

The main purpose of the certificate management is to provide a storage for certificates and a chain of trust in order to verify all public keys used for signature verification. Certificates are verified prior to their storage to the EB zentur HSM Firmware certificate store. During an update procedure, the certificate is verified and, if the validation is successful, the previous certificate is overwritten.

EB zentur HSM Firmware supports X.509v3 ITU-T X.690 Distinguished Encoding Rules (DER) certificates and OTC-CVC Profile Version 1.0 format certificates. The following algorithms are supported:

- ECDSA NIST P-256
- RSASSA-PKCS1-v1_5 and RSASSA-PSS with the key length of 2048 and 3072 bits

EB zentur HSM Firmware has the capacity to store certificates, based on either ECC or RSA cryptography, with a maximum of either:

- 9 ECC-based certificates
- 4 RSA-based certificates

Additionally, EB zentur HSM Firmware supports certificates with a maximum depth of 3 levels.

4.3.5. Secure boot

The main purpose of secure boot is to ensure the integrity of the application software that runs on the host. The architecture is based on the SHE specification adapted to the context of EB tresos AutoCore and EB zentur HSM Firmware on a Classic AUTOSAR system on chip (SoC).

The validation of the EB zentur HSM Firmware image itself is not within the scope of the secure boot functionality. To preserve the chain of trust, validation needs to happen before the EB zentur HSM Firmware is initialized. The verification of EB zentur HSM Firmware may be provided by an HSM bootloader, a SoC HSM Boot ROM, the one-time programmable (OTP) protection of the EB zentur HSM Firmware flash, or by other mechanisms supported by the hardware.

4.3.6. Firmware update

Firmware update is designed to enable the update of EB zentur HSM Firmware after it is in use on the field. The updatability concept varies depending on the hardware platform and derivative.

During firmware update, the new EB zentur HSM firmware is verified by signature verification. EB zentur HSM Firmware supports also a rollback feature if the system with the new firmware is detected to have unpredicted issues. EB zentur HSM can be stopped during concurrent PFlash operations, e.g to enable a firmware update on the host side if the same PFlash band is used by host and HSM.

4.3.7. Life cycle management

To prevent unauthorized key update, the EB zentur HSM Firmware's life cycle management provides private key and root certificate locking.

4.4. Applicable standards

EB zentur HSM Firmware is compatible with the SHE - Secure Hardware Extension Functional Specification version 1.1 (rev 439).

4.4.1. SHE+ requirements

In addition to the SHE requirements, EB zentur HSM Firmware implements the following requirements commonly known as SHE+:

- Additional key slots (SHE+ extended keys)

  EB zentur HSM Firmware extends the 10 standard generic key slots specified by SHE v1.1 to a total of 20 generic key slots.

- Key property flag: Verify-only

  This flag applies to any regular SHE key or any of the extended SHE+ keys. If this flag is set, then SHE restricts usage of the key to the verification of MACs only.

4.4.2. Key types and supported keys

SHE conceptually distinguishes the following key types:

ROM key
: This is a one-time programmable key that is stored in the read-only memory area. SHE specifies one single ROM key, the SHE SECRET_KEY, which is unique to the chip.

NVM keys
: These are non-volatile memory (NVM) keys. The stored NVM keys persist through a power cycle.

RAM key
: This is a volatile key held in secure RAM area. SHE specified one single RAM key slot, the SHE RAM_KEY.

EB zentur HSM Firmware supports the SHE standard keys specified in Table 4.2, “Supported SHE keys”:

Key name	Key type	SHE / SHE+
SECRET_KEY	ROM key	SHE
MASTER_KEY	NVM key	SHE
BOOT_MAC_KEY used by secure boot	NVM key	SHE
BOOT_MAC used by secure boot	NVM key (MAC calculated)	SHE

Key name	Key type	SHE / SHE+
KEY_1 - KEY_10	NVM key	SHE
RAM_KEY	RAM key	SHE
KEY_11 - KEY_20	NVM key	SHE+

Table 4.2. Supported SHE keys

4.5. Limitations and deviations

This section lists the limitations and deviations for EB zentur HSM Firmware.

- Input buffers, containing the messages to be processed, must be 16-byte aligned. Output buffers, receiving the encrypted/decrypted messages, must be 4-byte aligned.

  Length variable(s) for input/output buffer, like signLengthPtr in eb_hsm_sign(), must be 16-byte aligned and must have a size of at least 16 bytes.
- Number of supported asymmetric keys:
  - 1 ECDSA private key
  - 1 EdDSA private key
  - 9 ECC public keys
  - 1 RSA private key
  - 4 RSA public keys
- RSA key length of 1024 bits is not supported.
- For MAC verification with the key BOOT_MAC_KEY, the MAC length must be 128 bits.
- For ECC key generation, only NIST P-256 curve is supported.
- Only one ECC key pair (generated using the hardware accelerator) can be stored.
- Key generation of an asymmetric key pair for a certain key slot (only 1 available currently) can be executed only once to avoid overwriting. If triggered again, the error COMM_KEY_UPDATE_ERROR is thrown.

The following deviation to the SHE specification 1.1. exists:

- BOOT_MAC is write-protected from loading outside the HSM

  A CMAC of the host bootloader used in secure boot is stored in BOOT_MAC, and allowing the writing may compromise the secure boot process. This violates the SHE specification chapter 4.2.3, which states that BOOT_MAC can be written with the knowledge of the MASTER_ECU_KEY or BOOT_MAC_KEY and is protected by the common lock mechanisms described in chapters 4.1.1, 4.1.2, 4.1.3, and 4.1.4.

5. System and tool requirements

To use EB tresos Studio, your system must provide the following.

5.1. Software

Item	Requirement
Operating system	Microsoft Windows 10 LTSC, 64-bit Linux Ubuntu 16.04 LTS, 64-bit (CLI only)
ECU configuration tool	EB tresos Studio for ACG8
PDF reader for the user documentation	Adobe Acrobat Reader
Web browser to contact support and get the latest product news	e.g. Mozilla Firefox

Table 5.1. Software requirements

5.2. Hardware

Item	Requirement
Processor	Dual Core (minimal) Quad Core (recommended)
RAM	2 GB (minimal) 8 GB (recommended)
Connectors	USB port, if dongled licenses are used
Network	Network connection, if network licenses are used

Table 5.2. Hardware requirements

5.3. Supported target platforms

The EB zentur HSM Firmware has built-in support for multiple hardware platforms and derivatives as shown in the following table.

Hardware platform	Derivative	Package	Default compiler version
Infineon TriCore 2xx	TC23x TC27x TC29x	Basic	TASKING_TriCore-v6.2r2p4
Infineon TriCore 3xx	TC37x TC38x TC39x	Evita Evita OEM extension - VW	TASKING_TriCore-v6.2r2p4

Table 5.3. Supported hardware variants and derivatives

The EB zentur HSM Firmware is provided for one specific microcontroller derivative and compiler version as well as the options as stated in the release notes.

The EB zentur HSM Firmware is tested and qualified for this specific compiler version and these options on this specific microcontroller derivative. Other combinations of compilers or compiler versions and options, or hardware derivatives might be functional. The use of other combinations requires additional qualification packages as described in *PD_EB_tresos_Solutions_for_Platforms*. Elektrobit provides product support for qualified products only.

6. Open-source software

You can distinguish the software that is delivered with the EB zentur product into the following two categories:

- Software that is executed on the electronic control unit (ECU).
- Software that is used for the development infrastructure (configuration, generation, building) and thus executed on the development platform.

6.1. Open-source software in software executed on the ECU

EB tresos AutoCore software that is executed on the electronic control unit (ECU) does not include any open-source software.

This statement excludes software that does not originate from Elektrobit, for example, components of the AUTOSAR microcontroller abstraction layer which are manufactured by microcontroller vendors.

7. Compatibility with other Elektrobit products

EB zentur HSM Firmware is delivered by default as preconfigured binary.

EB zentur CRYSHE for EB zentur HSM Firmware is compatible with EB tresos AutoCore Generic 8 products and is delivered as a source code.

EB zentur Crypto Driver Hardware for EB zentur HSM Firmware is compatible with EB tresos AutoCore Generic 8 and is delivered as a source code.

8. Deliverables

EB zentur HSM Firmware will include the following content:

- The EB zentur HSM Firmware in binary format with flash scripts
- The EB zentur CrySHE or EB zentur Crypto Driver Hardware for EB zentur HSM Firmware
- User documentation including user manual and release notes
- A quality statement for the EB zentur HSM Firmware release

8.1. User documentation

User documentation is available in the English language and supports users with the following:

- To find instructions in the everyday use of the product
- To understand concepts of the product

8.2. Maintenance

The maintenance of an EB zentur product depends on the purchased license type. See the license agreement for details or contact our customer support.

8.3. Licenses

See the license agreement for details or contact our customer support.

9. Glossary

The following table lists abbreviations that are used in this document:

Abbreviation	Description
ACG	AutoCore Generic
AES	Advanced Encryption Standard (standardised encryption algorithm)
API	Application Programming Interface
BMHD	Boot Mode HeaDer
CBC	Cipher Block Chaining
CMAC	Cipher based Message Authentication Code
CrySHE	Cryptographic library module for SHE
DER	Distinguished Encoding Rules
DBGCTRL	system control specific DeBuG ConTRol bridge register
ECB	Electronic Code Book
EC	Elliptic Curve
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EdDSA	Edwards-curve Digital Signature Algorithm
EEPROM	Electrically Erasable Programmable Read Only Memory
FCON	Flash CONfiguration register
FEE	Flash EEPROM Emulation
FWU	FirmWare Update
HSM	Hardware Security Module
HSM2HT	HSM to Host
HT2HSM	Host to HSM
HW	Hardware
ITU-T	International Telecommunication Union-Telecommunication
ITU-T X.690	ITU-T X.690 is a standard defining the Abstract Syntax Notation One encoding rules
KDF	Key Derivation Function
MAC	Message Authentication Code
MCAL	MicroController Abstraction Layer

Abbreviation	Description
NIST	National Institute of Standards and Technology
NIST P-256	Defines a NIST P-256 curve over prime fields
NVM	Non Volatile Memory
OTP	One Time Programmable
PFlash	Program Flash
PKC	Public Key Cryptography
PKCS	Public-Key Cryptography Standards
PRNG	Pseudo Random Number Generator
RSA	Rivest–Shamir–Adleman public-key cryptosystem
RSA-PSS	RSA Probabilistic Signature Scheme
RSA-PKCS1	Signature Scheme standardized in version 1.5 of PKCS#1
SHA	Secure Hash Algorithm
SHE / SHE+	Secure Hardware Extension as specified by the *Herstellerinitiative Software (HIS)*, an association of German automobile manufacturers
SSW	Start-up Software
TRNG	True Random Number Generator
UCB	User Configuration Block
UID	Unique IDentification
X.509	X.509 is a standard defining the format of public key certificates

Table 9.1. Abbreviations

Appendix A. Supported features by hardware platform

Note: Variations may be possible due to different hardware capabilities.

Feature	Hardware					
	TC23x	**TC27x**	**TC29x**	**TC37x**	**TC38x**	**TC39x**
EB zentur HSM Firmware package	**Basic**			**Evita**		
EB zentur OEM extension				**VW**		
SHE 1.1 compatible *Deviation: BOOT_MAC is write-protected and prevented update outside of the HSM.*	x	x	x	x	x	x
SHE+ compatible	x	x	x	x	x	x
Debug access management *Restriction: Deletes only all the stored keys in NvM storage.*	x	x	x	x	x	x
HSM firmware update				x	x	x
HSM secure boot *CMAC (AES-128 based)*				x	x	x
Host secure boot ► CMAC (AES-128 based) ► Host boot loader ► Max 5 host application regions	x	x	x	x	x	x
Life cycle management Root certificate locking mechanism				x	x	x
Concurrent program flash operations				x	x	x
Cryptographic support						
Symmetric encryption						
AES-128 (ECB, CBC)	HW			HW		
MAC generation						

Feature	Hardware					
	TC23x	**TC27x**	**TC29x**	**TC37x**	**TC38x**	**TC39x**
EB zentur HSM Firmware package	**Basic**			**Evita**		
EB zentur OEM extension				**VW**		
CMAC (AES-128 based)	HW			HW		
SipHash-2-4 and SipHash-4-8, 128-bit key, 64-bit MAC				SW		
Signature verification						
RSASSA-PSS (2048, 3072) RSASSA-PKCS1-v1_5 (2048, 3072)				SW		
ECDSA (ECC NIST P-256)				HW		
EdDSA (ECC Ed25519ph)				SW		
Signature generation						
ECDSA (NIST P-256)				HW		
EdDSA (Ed25519ph)				SW		
RSASSA-PKCS1-v1_5 (2048, 3072)				SW		
Hash generation						
SHA-256				HW		
Key management						
Key generation						
ECC NIST P-256				x	x	x
Symmetric key load						
Ciphered format	x	x	x	x	x	x
Plain format (RAM key slot)	x	x	x	x	x	x
Asymmetric key load						
RSA-2048, RSA-3072				x	x	x
ECC Ed25519ph				x	x	x
ECC NIST P-256				x	x	x
Key export						
ECC NIST P-256 public key				x	x	x
RAM key (only if loaded plain)	x	x	x	x	x	x
EB zentur OEM extension						

Feature	Hardware					
	TC23x	**TC27x**	**TC29x**	**TC37x**	**TC38x**	**TC39x**
EB zentur HSM Firmware package	**Basic**			**Evita**		
EB zentur OEM extension				**VW**		
OEM extension VW ► VW Security Requirements ► VKMS 2.1				x	x	x
Key slots						
Secret key (hardware key, not writable nor readable)	1	1	1	1	1	1
Master ECU key slot	1	1	1	1	1	1
Boot MAC key slot	1	1	1	1	1	1
RAM key slot	1	1		1	1	1
NvM flash key slots symmetric keys (AES, CMAC)	20	20		50	50	50
ECC Ed25519ph private key slot	0	0		1	1	1
ECC NIST P-256 private key slot	0	0		1	1	1
ECC public key/certificate slots	0	0		9	9	9
RSA private key slots	0	0		1	1	1
RSA public key/certificate slots	0	0		4	4	4
Certificate management (via Proxy/CDH)						
► Chain: root, intermediate, signing ► Formats: X.509v3 ITU-T X.690 Distinguished Encoding Rules (DER), OTC-CVC Profile Version 1.0 ► Algorithms: ECDSA, RSA-PSS, RSASSA-PKCS1-v1_5 ► Key lengths: ECC NIST P-256, RSA-2048, RSA-3072				x	x	x

Appendix B. Supported features compliant with the VKMS 2.1 specification

Currently supported derivatives: Infineon Aurix TC37X/TC38X

VKMS: versions 2.1 and 2.2

Key storage and supported keys:

- 50 symmetric NvM AES-128 keys
- 1 RAM key
- Master ECU key
- Boot MAC key
- Boot MAC
- 1 ECDSA private key
- 1 EdDSA private key
- 9 ECC public keys
- 1 RSA private key
- 4 RSA public keys
- VKMS keys: for the default VKMS keys, see Appendix C, “Default VKMS keys“

Key management:

- Loading plain key into RAM key slot
- Loading ciphered keys into RAM key slot and NvM key slots
- Exporting RAM key as ciphered key container
- Supported VKMS version is only 2.1
- Supported DLC format is only version 2

Supported cryptographic algorithms via Crypto stack:

- AES-128 encryption / decryption: modes ECB, CBC, GCM
- SipHash24 (sw lib)
- CMAC generation / verification: AES-128 based
- HMAC-SHA256
- PRNG of 128 bits, including generation of a random seed through a TRNG and random seed extension

- HASH generation SHA2-256
- ECC key pair generation using ECDSA NIST curve P-256 (secp256r1)
- Exporting ECC public key in a raw format
- Digital signature verification using
  - ECDSA NIST curve P-256 (secp256r1)
  - EdDSA Ed25519, software algorithm
  - RSASSA-PSS, software algorithm
  - RSASSA-PKCS1-v1_5, software algorithm
- Digital signature generation using
  - ECDSA NIST curve P-256 (secp256r1)
  - EdDSA Ed25519, software algorithm
  - RSASSA-PKCS1-v1_5, software algorithm

Other features:

- Concurrent PFlash operations
- Life cycle management to block repeated loading of root certificate
- Certificate management: x509v3 and CVC
- Secure boot + extensions to verify host software applications
- HSM firmware update
- Cancel command

Appendix C. Default VKMS keys

```
/*
 * !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 * Note: The following VKMS_KEY_ID_* macros define default VKMS key IDs.
 * !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 */

#define VKMS_KEY_ID_PSS128      (0x0015U)
/* Symmetric 128. Used for PSS 21dec with DLC version 2 */

#define VKMS_KEY_ID_AES128_01   (0x1010U)
/* Symmetric 128 (genus 0x11, 0x13, 0x15 or 0x17) */

#define VKMS_KEY_ID_AES128_02   (0x1011U)
/* Symmetric 128 (genus 0x11, 0x13, 0x15 or 0x17) */

#define VKMS_KEY_ID_AES128_03   (0x1012U)
/* Symmetric 128 (genus 0x11, 0x13, 0x15 or 0x17) */

#define VKMS_KEY_ID_AES128_04   (0x1013U)
/* Symmetric 128 (genus 0x11, 0x13, 0x15 or 0x17) */

#define VKMS_KEY_ID_AES128_05   (0x1014U)
/* Symmetric 128 (genus 0x11, 0x13, 0x15 or 0x17) */

#define VKMS_KEY_ID_AES128_06   (0x1015U)
/* Symmetric 128 (genus 0x11, 0x13, 0x15 or 0x17) */

#define VKMS_KEY_ID_AES128_07   (0x1016U)
/* Symmetric 128 (genus 0x11, 0x13, 0x15 or 0x17) */

#define VKMS_KEY_ID_AES128_08   (0x1017U)
/* Symmetric 128 (genus 0x11, 0x13, 0x15 or 0x17) */

#define VKMS_KEY_ID_AES128_09   (0x1018U)
/* Symmetric 128 (genus 0x11, 0x13, 0x15 or 0x17) */

#define VKMS_KEY_ID_AES128_10   (0x1019U)
/* Symmetric 128 (genus 0x11, 0x13, 0x15 or 0x17) */

#define VKMS_KEY_ID_AES256_01   (0x101AU)
/* Symmetric 256 (genus 0x12 or 0x14) */

#define VKMS_KEY_ID_SYMVAR_01   (0x101BU)
```

```
/* Symmetric variable (genus 0x1F): max. size 32 bytes */

#define VKMS_KEY_ID_ASYMVAR_01  (0x101CU)
/* Asymmetric variable (genus 0x21 or 0x22): max. size 32 bytes */

#define VKMS_KEY_ID_CERT_01     (0x101DU)
/* Certificate (genus 0x01): max. size 800 bytes */

#define VKMS_KEY_ID_DATA_01     (0x101EU)
/* Data (genus 0x02): max. size 64 bytes */

/* Changes to the default VKMS key list may result in additional fees and
 * changes for delivery timeline and always need to be discussed separately.
 */
```