



Federal Office
for Information Security

Amendment to BSI TR-03151 Secure Element API (SE API)

02.12.2019



Federal Office for Information Security
Post Box 20 03 63
D-53133 Bonn

E-Mail: registrierkassen@bsi.bund.de
Internet: <https://www.bsi.bund.de>
© Federal Office for Information Security 2019

Table of Contents

1	Introduction.....	5
1.1	Terminology.....	5
2	Corrections and clarifications.....	6
	References.....	14

Figures

Tables

1 Introduction

The Technical Guideline BSI TR-03151 [TR-03151] specifies the Secure Element API (SE API).

The SE API is a digital interface that wraps functionality of a Secure Element and allows access to security functionalities by an application in a standardized way regardless of the specific type of Secure Element in use.

The Technical Guideline BSI TR-03151 [TR-03151] focuses on the Secure Element functionality, the creation and structure of log messages, their export and the specification of the integration interfaces to the application.

This document amends version 1.0.1 of BSI TR-03151 [TR-03151] and contains clarifications.

1.1 Terminology

This document lists clarifications. To facilitate the detection of those clarifications color cues are used, if deemed necessary (for example to highlight an addition in a table). **Yellow** marked text indicates that the marked words or characters are new. Additions are usually only highlighted, if they are made in between segments, to ease detection and otherwise maintain readability. **Red** marked words or characters indicate that the marked words or characters are wrong. **Green** marked words or characters indicate that the marked words or characters are correct.

2 Corrections and clarifications

This section gives clarifications and corrections of BSI TR-03151 [TR-03151] .

Chapter 2.1, page 10

Add sentence after table 2:

“Even if individual entries in the table are not categorized as mandatory, it is possible that the usage of this structure is further refined in this document or by the respective requirements given by the referencing Technical Guideline of this document.”

Chapter 2.1, page 10, chapter 2.2, page 11, chapter 2.4, page 14, chapter 4.4.1.2, page 28, chapter 4.5.5.2, page 28, chapter 5.1.3, page 60, chapter 9, page 77


Add clarification:

“Serial numbers SHALL NOT be base64 encoded. The octet string of a serial number SHALL be the representation of hex values such as AAFF99.”

Chapter 2.3.1, page 12

Replace

“A transaction log MUST be identified by the following object identifier (id-SE-API-transaction-log):

- bsi-de (0.4.0.127.0.7 ) applications (3) sE-API (7) sE-API-dataformats(1) 1”

with

“A transaction log MUST be identified by the following object identifier (id-SE-API-transaction-log):

- bsi-de (0.4.0.127.0.7) applications (3) sE-API (7) sE-API-dataformats(1) 1”.

Chapter 2.3.1, page 12


Add sentence at the end of the first bullet point in the second enumeration:

“The individual elements in an indefinite length element SHOULD be TLV encoded so that it is not necessary to search for end-of-contents octets in the process data itself.”

Chapter 2.3.2, page 13

Replace

“A system log MUST be identified by the following object identifier (id-SE-API-system-log):

- bsi-de (0.4.0.127.0.7 ) applications (3) sE-API (7) sE-API-dataformats(1) 2”

with

“A system log MUST be identified by the following object identifier (id-SE-API-system-log):

- bsi-de (0.4.0.127.0.7) applications (3) sE-API (7) sE-API-dataformats(1) 2”.

Chapter 2.3.3, page 13

Replace

“An audit log MUST be identified by the following object identifier (id-SE-API-SE-audit-log):

- bsi-de (0.4.0.127.0.7 **D**) applications (3) sE-API (7) sE-API-dataformats(1) 3”

with

“An audit log MUST be identified by the following object identifier (id-SE-API-SE-audit-log):

- bsi-de (0.4.0.127.0.7) applications (3) sE-API (7) sE-API-dataformats(1) 3”.

Chapter 2.4, page 14

Add clarifications at end of chapter:

“The origin of the certificate used to verify a signature MUST be verified using the certificate chain provided.”

Chapter 2.4, page 14, last sentence

Replace

“After the affiliation and the correctness of the certificate belonging to the log messages has been verified, the log message signature verification SHALL be performed analogue to the signature creation using the public key of the certificate for the verification of **S** log message type.”

with

“After the affiliation and the correctness of the certificate belonging to the log messages has been verified, the log message signature verification SHALL be performed analogue to the signature creation using the public key of the certificate for the verification of **the given** log message type.”

Chapter 2.5.4, text 2, page 16

Delete “operationType” from “From Secure Element:”.

Chapter 2.5.4, page 16

Add second note to Text 2:

The parameter “operationType” inside “certifiedData” is set by SE API itself.

Chapter 2.5.4, text 2, page 16

Replace

“clientId**D**”

with

“clientId”.

Chapter 4, page 19

Add new bullet point to “Utility functions”

- “GetTimeSyncVariant”

Chapter 4.1.1.3, page 20

Delete “initialize” from enumeration.

Add sentence after enumeration:

“The exception `ErrorCertificateExpired` may be implemented by the SE API function `initialize`.”

Chapter 4.1.1.4, page 21

Add new item to list

- “GetTimeSyncVariant”

Chapter 4.3.2.3, table 14, page 25

Add missing exception “`ErrorInvalidTime`”, which is thrown if an invalid time value was provided:

Exception	Meaning
<code>ErrorUpdateTimeFailed</code>	The execution of the Secure Element functionality to set the time failed.
<code>ErrorInvalidTime</code>	The time value is invalid (eg. year out of bounds)
<code>ErrorRetrieveLogMessageFailed</code>	The execution of the Secure Element functionality to retrieve log message parts has failed.
<code>ErrorStorageFailure</code>	Storing of data of the log message has failed.
<code>ErrorSeApiNotInitialized</code>	The function <code>updateTime</code> is invoked although the SE API has not been initialized.
<code>ErrorCertificateExpired</code>	The certificate corresponding to key signing the log message expired. The exception <code>ErrorCertificateExpired</code> SHALL be raised after the data of the log message has been stored.
<code>ErrorSecureElementDisabled</code>	The Secure Element has been disabled.
<code>ErrorUserNotAuthorized</code>	The user who has invoked the function <code>updateTime</code> is not authorized to execute this function.
<code>ErrorUserNotAuthenticated</code>	The user who has invoked the function <code>updateTime</code> has not the status <code>authenticated</code> .

Table 14: Exceptions for `updateTime` function

Chapter 4.3.3, page 26

Add second sentence:

“With this command only the Secure Element is deactivated. The execution of the export function is still possible.”

Chapter 4.3.3.4, page 27

Extend enumeration with point 8:

“8. Functions in which the Secure Element is not involved can still be executed (e.g. Export).”

Chapter 4.4.2.1, page 30, table 19

Extend meaning in Table 19. The parameter “processType” SHALL be used in case of a “signed update” but SHALL NOT be used for an “unsigned update”.

Name	Type (OMG IDL)	Required?	Meaning
clientId	string	REQUIRED	Represents the ID of the application that has invoked the function.
transactionNumber	unsigned long	REQUIRED	This parameter is used to unambiguously identify the current transaction.
processData	octet []	REQUIRED	This parameter represents all the necessary information about the process since the initial state of the process or its last update.
processType	string<100>	OPTIONAL	This parameter is used to identify the type of the transaction as defined by the application. Note that this parameter shall be used in case of a “signed update” but shall not be used for an “unsigned update”.

Table 19: Input parameters for updateTransaction function

Chapter 4.4.2.3, page 31, Table 21, first column, third row

Replace

“ErrorLogMessageRetrievalFailed”

with

“ErrorRetrieveLogMessageFailed”.

Chapter 4.4.2.4, page 31, bullet 1 of the list

Replace

“ErrorUpdateExternalTransactionFailed”

with

“ErrorUpdateTransactionFailed”.

Chapter 4.4.2.4, page 31, within fourth bullet point of enumeration

Replace

“1. Next, the function SHALL retrieve the parts of the log message determined by the Secure Element. If the execution of this function fails, the exception ErrorLogMessageRetrievalFailed SHALL be raised.”

with

“1. Next, the function SHALL retrieve the parts of the log message determined by the Secure Element. If the execution of this function fails, the exception ErrorRetrieveLogMessageFailed SHALL be raised.”

Chapter 4.4.3.3, page 33, table 24

Add missing exception “ErrorNoTransaction”, which is thrown if no transaction is known to be open under the provided transaction number.

Exception	Meaning
ErrorFinishTransactionFailed	The execution of the Secure Element functionality to finish a transaction failed.
ErrorRetrieveLogMessageFailed	The execution of the Secure Element functionality to retrieve the parts of the log message has failed.
ErrorNoTransaction	No transaction is known to be open under the provided transaction number.
ErrorStorageFailure	Storing of the log message failed.
ErrorSeApiNotInitialized	The function finishTransaction is invoked although the SE API has not been initialized.
ErrorTimeNotSet	The function finishTransaction is invoked although the time managed by the Secure Element has no defined value.
ErrorCertificateExpired	The certificate corresponding to key signing the log message expired. The exception ErrorCertificateExpired SHALL be raised after the data of the log message has been stored.
ErrorSecureElementDisabled	The Secure Element has been disabled.

Table 24: Exceptions for finishTransaction function

Chapter 4.4.3.4, page 33

Extend enumeration additional second point:

“The following description specifies the behavior of the finishTransaction function in detail:

1. The function SHALL invoke the functionality of the Secure Element to finish a transaction and pass on the clientId, the transactionNumber of process to finish (, the processType) and the processData. If the execution of the function fails, the exception ErrorFinishTransactionFailed SHALL be raised.

2. The Secure Element SHALL check whether the transactionNumber belongs to an open transaction. If this is not the case, the function SHALL return the error ErrorNoTransaction and exit.
3. Next, the function SHALL retrieve the parts of the log message determined by the Secure Element. If the execution of this function fails, the exception ErrorRetrieveLogMessageFailed SHALL be raised.
4. The process data, created since the start or the last signed update (cf. Chapter 4.3.2.4) of the transaction and the data of the retrieved log message parts SHALL be stored. If the data has not been stored successfully, the function SHALL raise the exception ErrorStorageFailure.
5. The function SHALL return the time of the log message creation by logTime, the signature counter by signatureCounter and MAY return the signature value by signatureValue. Additionally, the SE API SHALL return the return value EXECUTION_OK to indicate that the execution of the function finishTransaction has been successful.”

Chapter 4.5.1.4, page 36

Extend enumeration with point 6:

6. If clientId has been provided, only transactions should be exported, in which the clientId was involved. The SE-API MAY export the complete transaction, or just the transaction log message files, which clientId is identical to the provided one.

Chapter 4.6

Add section 4.6.7:

4.6.7 GetTimeSyncVariant

The function getTimeSyncVariant can be used to obtain information how updates of the current date/time are performed. In this context, the following variants for updating the current date/time can be supported:

1. The underlying Secure Element supports time synchronization.
2. The current time needs to be provided in ASN.1 UTCTime (see [ITU2015]).
3. The current time needs to be provided in ASN.1 GeneralizedTime (see [ITU2015]).
4. The current time needs to be provided in Unix Time (see [IEEECS2018]).

4.6.7.1 GetTimeSyncVariant – Input parameters

None.

4.6.7.2 GetTimeSyncVariant – Output parameters

Name	Type (OMG IDL)	Required?	Meaning
supportedSyncVariant	enum SyncVariants { noInput, utcTime, generalizedTime, unixTime }	REQUIRED	Represents the supported variant for updating the current date/time.

Output parameters for getTimeSyncVariant function

4.6.7.2 GetTimeSyncVariant – Exceptions

Exception	Meaning
ErrorGetTimeSyncVariantFailed	The identification of the supported variant for updating the current date/time failed.
ErrorSeApiNotInitialized	The function getTimeSyncVariant is invoked although the SE API has not been initialized.
ErrorSecureElementDisabled	The Secure Element has been disabled.

Exceptions for getTimeSyncVariant function

4.6.7.4 GetTimeSyncVariant – Detailed description

The function getTimeSyncVariant SHALL identify the supported variants to update the current date/time. The following description specifies the behavior of the function getTimeSyncVariant in detail:

1. If the identification of the supported update variant fails, the function SHALL raise the exception ErrorGetTimeSyncVariantFailed and exit the function.
2. If the identification of the supported update variant has been successful, the function SHALL return the information regarding the supported update variants over the output parameter supportedSyncVariant. Additionally, the function SHALL return the return value EXECUTION_OK to indicate that the execution of the function has been successful.

Chapter 4.6.6, page 47

Replace

“The function deleteStoredData deletes all **data that is** stored in the storage.”

with

“The function deleteStoredData deletes all **log message files that are** stored in the storage.”

Chapter 4.7.1, page 48

Add sentence:

“The Secure Element may create (signed) log messages for authentication functions, even if it has not been initialized.”

6. Chapter 5.1.1, page 55, second sentence,

Replace

“**coma**”

with

“**comma**”.

Chapter 5.1.1, page 55

Add clarifications at end of chapter:

“All information in Text 4 SHOULD be inside the first line of the file and encapsulated in double quotes. As the values (\$1, \$2, \$3) are encapsulated in double quotes, the values may contain commas. The file “info.csv” SHALL NOT provide any additional information.”

Chapter 5.1.2.2, page 58, last section,

Replace

“**ANS**.1”

with

“**ASN**.1”.

Chapter 6, page 61

Add three more sentences:

“The data type SHOULD not be tagged. For example, in a system log message of an initialization, there should be no “0x13” tag for printable strings after “0x81LL” inside of “description”. If a time-stamp is logged, the format of “logTime” SHOULD be used.”

Chapter 6.4, page 61, table

Add missing role “unknown”, which is used for an unknown UserID.

Data field	Tag	Data type	Mandatory?	Description
userId	0x81	PrintableString	m	MUST contain the ID of the user who or application that has invoked the authentication function.
role	0x82	ENUMERATED{ admin, timeAdmin }	m	MUST represent the role of the user/application. Must contain the value <ul style="list-style-type: none"> – “unknown” for an unknown UserID or – “admin” for the Admin role or – “timeAdmin” for the role TimeAdmin (see chapter 4.2).
authenticationResult	0x83	BOOLEAN	m	MUST contain the result of the authentication procedure. The value “TRUE” SHALL indicate that the authentication has been successful. The value “FALSE” SHALL indicate that the authentication has failed.

Chapter 10, Appendix E, page 79-80

“ecdsa-plain-SHA224” and “ecdsa-plain-SHA3-224” are used in Appendix E.

Add sentence:

“Please note that depending on the context, some algorithms listed here SHALL NOT be used.”

References

TR-03151: BSI, Technical Guideline BSI TR-03151 Secure Element API (SE API), Version 1.0.1, 20.12.2018, 2018

TR-03116: BSI, Technische Richtlinie TR-03116 "Kryptographische Vorgaben für Projekte der Bundesregierung - Teil 5: Anwendungen der Secure Element API", 01. 02.2019,