Singapore Standard SS 518 : 2014

Specification for contactless e-purse application

AMENDMENT NO. 1

January 2019

1. **Page 9, Clause 0 Introduction**

Replace the entire paragraph 2 with the following:

This standard describes 2 versions of the contactless e-purse application – CEPAS 2.0 and CEPAS 3.0.

2. **Page 10, Clause 3 Definitions / Abbreviated terms**

Add the following definitions / abbreviated terms to the table:

Definitions / Abbreviated terms	Description
AES	Advanced Encryption Standard
AES-CBC	Advanced Encryption Standard CBC operation
AES-ECB	Advanced Encryption Standard ECB operation
Cert	Certificate
DCAN	Digitised CAN
ECC	Elliptic-Curve Cryptography
ECDSA	Elliptic-Curve Digital Signature Algorithm
HSM	Hardware Security Module
KCV	Key Check Value
SHA-256	Secure Hash Algorithm 2 using 32-bit words
TA	Transaction Acquirer
Txn	Transaction

3. **Page 14, 4.4.4 Terminal Reference Parameter (TRP)**

Replace “In CEPAS 1.0, the TRP is a 3-byte value while in CEPAS 2.0, the TRP is a 4-byte value.” with “In CEPAS 2.0 and CEPAS 3.0, the TRP is a 4-byte value.”.

4. **Page 15, Section Three – Detailed description of CEPAS 1.0**

Delete the entire Section Three, “Detailed description of CEPAS 1.0” and renumber Section Four, “Detailed description of CEPAS 2.0”, including all clauses, tables and figures accordingly (e.g. “9.1” will be replaced with ‘8.1”).

5. **New section**

Add the following new section after Section Three, “Detailed description of CEPAS 2.0”.

Section Four – Detailed description of CEPAS 3.0

9 CEPAS 3.0 – CEPAS tokenization

9.1 CEPAS 3.0 overview

9.1.1 The design of CEPAS tokenization takes into consideration the following requirements and features:

– The design is based on CEPAS 2.0 card (SS518: 2014).
– CEPAS token is a flat file that aim to be easily portable to other card media/form factor. If it is used in other type of card media, the prevailing card media mutual authentication method should be used to ensure card authenticity.
– There are no changes to the base CEPAS card operating system. Existing Transit EF should be converted to be Token EF. For newly issued card, Token EF will be created during personalisation phase.
– Transaction computations are computed by the terminal device with key information retrieve from card.
– DCAN complies with ISO/IEC 7812-1
– The token data that is stored in the Token file is signed by Token Signature Key, an ECDSA private key. A public key on the terminal device is used to verify the token signature.

CEPAS Token Transactions signed by device using Transaction Cert Key. The signing computation is performed by the terminal device (not the card). The token certificate key is unique to each CEPAS token.

9.2 CEPAS token commands

The following commands will still be used in CEPAS 3.0 without the atomic update:

– Read Purse (8.4)
– Read Binary (8.8)
– Get Challenge (8.9)
– Update Command without atomic update chain, CLA = 90 (8.5)

9.3 CEPAS token EF file design

Data object	Length	Security
DCAN	10 bytes	None
Token Signature Key Version	1 byte	None
Token Format Version	1 byte	None
Token Start Date	2 bytes	None
Token Expiry Date	2 bytes	None
Authentication Mode	1 byte	None
Form Factor Indicator	1 byte	None
Acquirer Encrypting Key Version	1 byte	None
Transaction Cert Key Version	1 byte	None
Diversified Transaction Cert Key	24 bytes	AES-CBC (Diversified. Acquirer Encryption Key)
Diversified Transaction Cert Key KCV	3 bytes	None
Issuer Encrypting Key Version	1 byte	None

Diversified Transaction Cert Key	24 bytes	AES-CBC (Diversified Issuer Encryption Key)
Token Signature	48 bytes	None
Acquirer Data	32 bytes	None

The Token Signature is produced by using ECDSA on the token data above which is all the data above the Token Signature using the ECC Private Key as shown in Figure 7.

9.4 CEPAS token EF file initialisation

Initialising the Token File or conversion of Transit File to Token File involves connection to the Tokenization Server in order to generate the DCAN, CEPAS token signature, encrypted keys and the token data. This token data will be validated via the signature before initialising it to the card.

Tokenization server high level processing:

a) Securely retrieve the Diversified Transaction Cert Key and the corresponding key version from HSM or from Card Manager SAM. Transaction Cert Key management, diversification is under discretion of Card Manager and beyond the scope of this document.

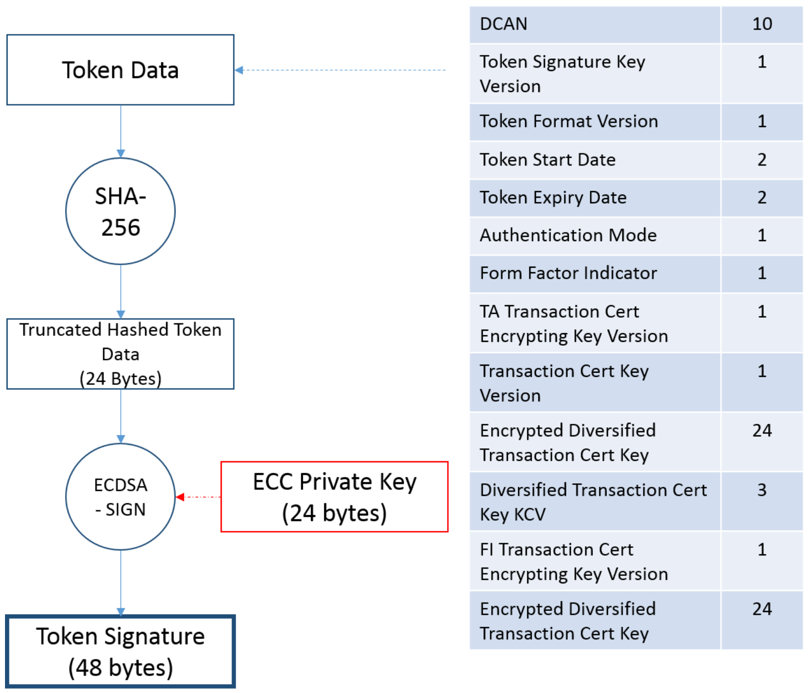
b) Retrieve Acquirer AutoLoad Key version and compute the Diversified Acquirer AutoLoad Key.

c) Encrypt the Diversified Transaction Cert Key using Diversified Acquirer AutoLoad Key.

d) Retrieve Diversified Issuer AutoLoad Key and the key version from HSM or from Card Manager SAM.

e) Encrypt the Diversified Transaction Cert Key using Diversified Issuer AutoLoad Key.

f) After which the Tokenization Server will structure the CEPAS token data as described in Token EF the EF File (Items 1-14) and generate CEPAS Token signature.



DCAN	10
Token Signature Key Version	1
Token Format Version	1
Token Start Date	2
Token Expiry Date	2
Authentication Mode	1
Form Factor Indicator	1
TA Transaction Cert Encrypting Key Version	1
Transaction Cert Key Version	1
Encrypted Diversified Transaction Cert Key	24
Diversified Transaction Cert Key KCV	3
FI Transaction Cert Encrypting Key Version	1
Encrypted Diversified Transaction Cert Key	24

Figure 7 – Token signature creation

9.5 Verification and update of token EF file

Below are the steps in the verification and updating of the Token EF File

Prerequisite:
Device/reader must have the CEPAS Acquirer/Issuer AutoLoad Key to authenticate CEPAS card.
Device/reader must to have the ECDSA Public Key to verify CEPAS token signature.

Typical flow:

a) Perform CEPAS Card authentication with Read Purse with Authentication command using either Diversified Acquirer AutoLoad Key or Diversified Issuer AutoLoad key.

b) Verify the CEPAS token signature using ECDSA public key.

c) Obtain Diversified Transaction Cert Key, either using Diversified Acquirer AutoLoad Key or Diversified Issuer AutoLoad Key.

d) Device generate Transaction Certification using Diversified Transaction Cert Key.

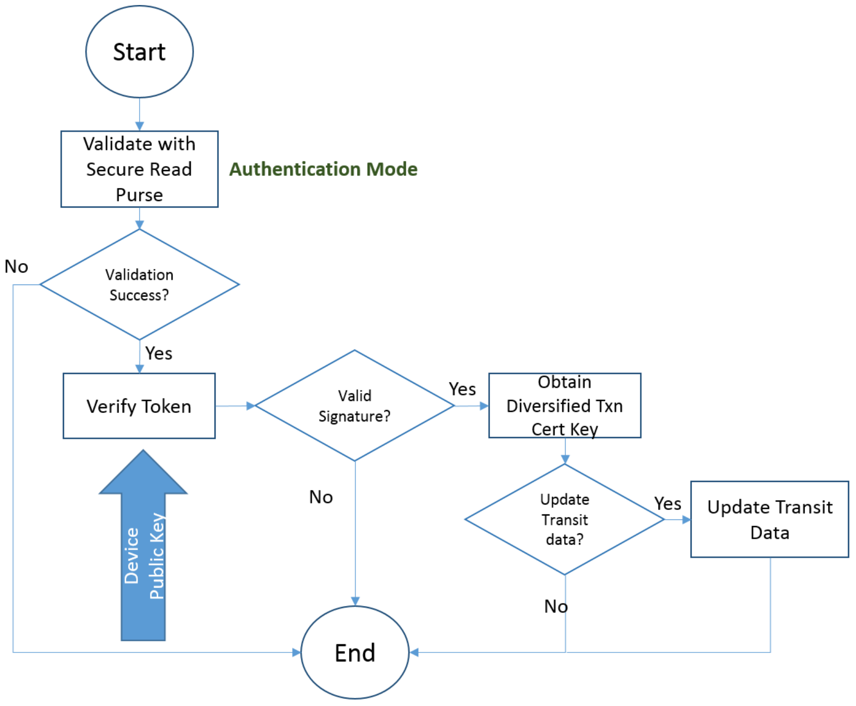


Figure 8 – Flow on the verfication and update of Token EF File (CEPAS 3.0)

9.6 Transaction certificate computation

A Transaction Signed Certificate (Figure 9) is computed by performing AES on the Transaction Header and hashed DCAN ID using a Signing Session Key. The Transaction Header to be encrypted is always 16-byte block, hence a Transaction Signed Certificate is always 16 bytes.

Hashed DCAN is the most significant 8 bytes of DCAN hash using SHA-256.

Transaction Header to be signed (encrypted) by the Signing Session Key:

Transaction Header			**Hashed DCAN**
Transaction Type	**Transaction Amount**	**Transaction DateTime**	
1 byte	3 bytes	4 bytes	8 bytes

The Signing Session Key is computed by performing AES-CBC (with IV) on the following 16-byte record using the selected Diversified Signing Key on the card.

Device random	**Card random**
8 bytes	8 bytes

Device Random and Card Random generated during the prior GET CHALLENGE

Command for Card Authentication.

The composition of the IV as stated below:

TRP	Filler = zero
4 bytes	12 bytes

Device random	Card random
8 bytes	8 bytes

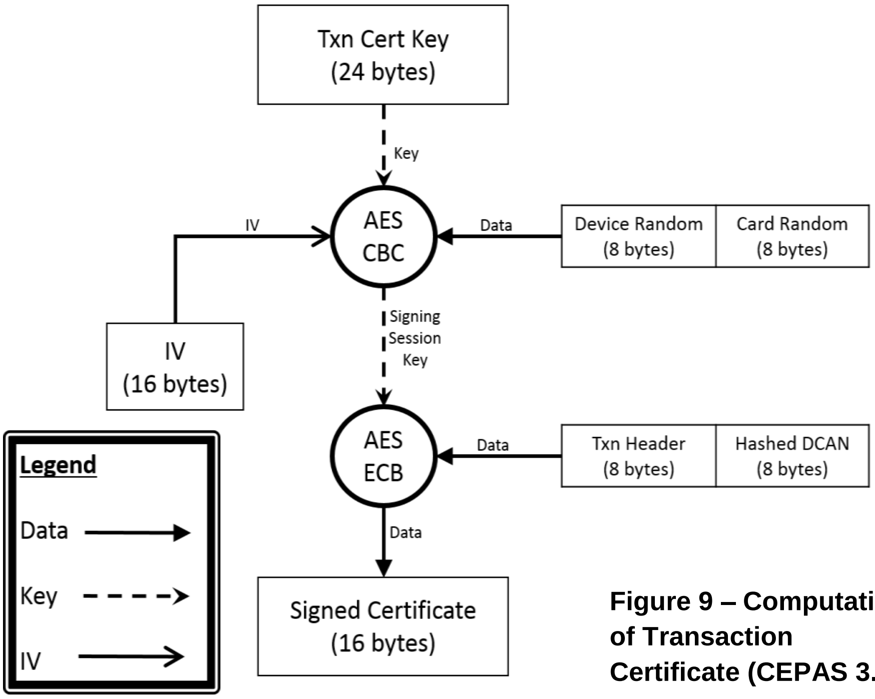


Figure 9 – Computation of Transaction Certificate (CEPAS 3.0)

6. Page 42, "Annex B"

(a) *Delete* "B.1", "B.2", "B.3", "B.4" and "B.5".

(b) *In "B.6", replace* "As described in 9.5," with "As described in 8.7,"

(c) *In "B.7" and "B.8", replace* "As described in 9.2," with "As described in 8.2,"

(d) *In "B.9" and "B.10", replace* "As described in 9.3," with "As described in 8.3,"

(e) *In "B.11", replace* "As described in 9.4," with "As described in 8.4,"

(f) *Replace* "B.6", "B.7", "B.8", "B.9", "B.10" and "B.11". numbering with "B.1", "B.2", "B.3", "B.4", "B.5" and "B.6" respectively.

(g) *Add* a new clause as follows:

B.7 CEPAS 3.0 – Token data signature creation

As described in 9.4, a sample set of data for the signature creation of the CEPAS Token EF File

```
Curve:                  secp192r1

Private Key:            09 55 69 0B 98 15 8F A0 29 EA E9 F5 33 0D 8F 2E
                        9C 2C 81 67 BF EB 75 E7
Public Key:             1C A6 F9 7F 66 0C CE FB 93 76 96 37 93 14 2E 9A
                        09 8E D4 11 C4 1F 24 FD 5C B5 EB E4 F0 3D CE 3B
                        2B 4E C2 15 24 34 DF 8F B0 0C 52 7E 2E 5F C4 CF

Token Data:             97 02 02 13 00 19 76 53 69 5F 01 01 1D 1C 27 CD
                        01 01 55 01 C3 28 49 74 12 CA 97 A6 14 76 41 4C
                        F7 95 B9 CB 8A F6 8B 72 F5 C2 C5 BC CC 07 4E 56
                        58 BE 61 90 B9 DB CB 4E 7C A9 AE 24 A0 85 6E 8F
                        9B 0F 95 2D BF 66 09 F8 C4 B4 D6 6D 46 16 27 D0
                        93 E6 3E 2D A2 21 64 0B 12 3B 18 27 6D D1 A5 90
                        EC FA 0E 13 BA 08 89 FE EC CB 25 64 C7 2D AC 17
                        D1 F4 B6 9A FB BD 98 53 FF FF FF FF FF FF FF FF
                        FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
                        FF FF FF FF FF FF FF FF

SHA-256 hash:           E5 AE AD C7 08 59 30 D4 22 8F 64 4E 21 FA 68 71
                        B3 53 4F 58 1C 13 CD 23 6D 22 17 2F EA 43 FF DA

Truncated SHA-256 hash:
                        E5 AE AD C7 08 59 30 D4 22 8F 64 4E 21 FA 68 71
                        B3 53 4F 58 1C 13 CD 23
< Output >
Signature:              21 0A D6 74 6F 7A 36 6E E4 2D C3 D1 B4 A5 40 F3
                        38 C0 27 1D 69 0F 11 89 7E 64 EC DD 63 AC 5B F4
                        D9 97 0C DC 91 53 D5 28 85 55 AF E8 21 10 3B BA
```