Effective:
1 January 2025
Version 016

AUSTRALIAN PAYMENTS NETWORK LIMITED

ABN 12 055 136 519

A Company limited by Guarantee

Code Set

for

ISSUERS AND ACQUIRERS COMMUNITY FRAMEWORK

Volume 4
Device Requirements and Cryptographic Management

Commenced 1 July 2015

Australian Payments Network Limited
Telephone: (02) 9216 4888

Code Set for

ISSUERS AND ACQUIRERS COMMUNITY FRAMEWORK

Volume 4
Device Requirements and Cryptographic Management

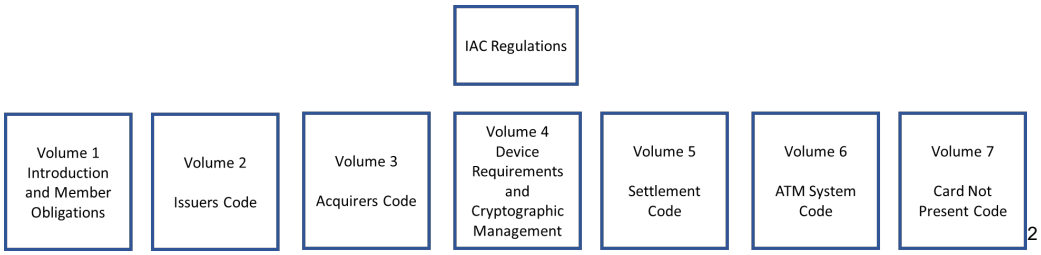
INDEX

PART 1 INTRODUCTION, INTERPRETATION AND DEFINITIONS

1.1 Purpose of this volume[1]

The IAC has been established to develop, implement and operate effective standards, policies and procedures to promote the efficiency, security and integrity of Australian Card Payments. These include minimum security standards, interoperability standards and value added services that support how payment cards are used throughout Australia.

These standards and requirements are contained within the IAC Code Set which is structured as follows:



[2]

Volume 4 is intended to be read in conjunction with Volumes 1, 2 & 3.[3]

It is an IAC requirement that all Devices, Solutions and Non-Standard Technologies hold a current AusPayNet approval prior to and during use within the IAC.[4]

This volume is structured in four parts. Part 1 provides introductory material and details the definitions that are used throughout the IAC Manual. Part 2 is no longer used. Part 3 addresses the process of approval for Devices, Solutions and Non-Standard Technologies. Cryptographic standards such as key length and approved algorithms are detailed in Part 4 including Terminal Key Management requirements.[5]

1.2 Interpretation[6]

Interpretations are located in a separate document entitled ‘Interpretation & Definitions'.

1.3 Definitions

Definitions are located in a separate document entitled ‘Interpretation & Definitions’.

Next page is Part 2

[1] Amended effective 1/1/19, version 008 r&p 002.18
[2] Amended effective 1/7/19, version 009 r&p 001.19
[3] Amended effective 16/12/21, version 013 r&p 001.21
[4] Last amended effective 16/12/21, version 013 r&p 001.21
[5] Last amended effective 16/12/21, version 013 r&p 001.21
[6] Amended effective 1/1/23, version 014 r&p 002.22

PART 2 DEVICE SECURITY STANDARDS [DELETED][7]

[Deleted]

Next page is Part 3

[7] Deleted effective 16/12/21, version 013 r&p 001.21

PART 3 DEVICE APPROVALS [8]

This Part 3 contains the IAC's requirements of the Company in relation to the approval of Devices, Solutions and Non-Standard Technologies for use in Interchange. Part 1.1 of this Volume 4 states the purpose of the IAC. In the context of Approved Devices that purpose includes balancing the interest of maintaining the security and integrity of Australian Card Payments with the interest of promoting innovation and competition.

3.1 Device Approval Process[9]

(a) The Company is responsible for:

(i) establishing the Device Approval Process;

(ii) reviewing and determining applications from Device Approval Applicants for approval of Non-Standard Technologies including determining any conditions to be attached to an approval and issuing Letters of Approval;[10]

(iii) revocation of any device approval, as contemplated in the Device Approval Process.[11]

(iv) amending the Device Approval Process; and

(v) publishing the Approved Devices List on the AusPayNet website.

(b) Each of the responsibilities of the Company specified in (a) may be exercised by the Company with the approval of the Chief Executive Officer, without the need to obtain approval of the IAF or any other person.

3.2 Approved Devices[12]

(a) The Device Approval Process sets out the process for approval of Devices for use in the IAC.

(b) Subject to the Device Approval Process, a Device is approved for use in the IAC if the Device is:

(i) listed as approved on the website of an Approved Standards Entity and complies with an Accepted Standard; or

(ii) listed in the AusPayNet-Approved Devices List published on the Company's website; or

[8] Last amended effective 16/12/21, version 013 r&p 001.21
[9] Last amended effective 16/12/21, version 013 r&p 001.21
[10] Amended effective 1/1/25, version 016 r&p 001.24
[11] Amended effective 1/1/25, version 016 r&p 001.24
[12] Inserted effective 1/1/25, version 016 r&p 001.24

(iii) approved for the use in a pilot under a Pilot Letter of Approval issued by the Company.

3.3 Period of permitted use of Approved Devices[13]

(a) The Device Approval Process determines the period of permitted use in the IAC of Approved Devices.

(b) Subject to the Device Approval Process, the period of permitted use in the IAC:

(i) for Devices listed as approved on the website of an Approved Standards Entity is:

(A) the Approval Period which expires on the expiry date for the Approved Device; and

(B) the Sunset Period which expires on the sunset date published in the schedule of sunset dates on the Company's website;

(ii) for Devices listed in the AusPayNet-Approved Devices List is:

(C) the Approval Period which expires on the expiry date for the Approved Device; and

(D) the Sunset Period which expires on the Device's sunset date published in the AusPayNet-Approved Devices List;

(iii) for Devices approved for use in a pilot the Approval Period which expires on the date stated in the Pilot Letter of Approval.

(c) During a Sunset Period only Devices purchased during the Approved Period can be used.

(d) The period of permitted use in the IAC may be revoked by the Company as provided in the Device Approval Process and if revoked will be published in the revocation section of the AusPayNet-Approved Devices List.

3.4 Approval of Devices [Deleted][14]

3.5 Approved Evaluation Facilities [Deleted][15]

3.6 Evaluation Costs [Deleted][16]

[13] Inserted effective 1/1/25, version 016 r&p 001.24
[14] Deleted effective 16/12/21, version 013 r&p 001.21
[15] Deleted effective 16/12/21, version 013 r&p 001.21
[16] Deleted effective 1/1/19, version 008 r&p 002.18

3.7 Agreements [Deleted][17]

3.8 Evaluation Facility Accreditation Process [Deleted][18]

Next page is Part 4

[17] Deleted effective 1/1/19, version 008 r&p 002.18

[18] Deleted effective 1/1/19, version 008 r&p 002.18

PART 4 CRYPTOGRAPHIC STANDARDS AND KEY MANAGEMENT

4.1 Cryptographic Key Management – General

Unless specifically detailed elsewhere, the following key management practices must apply. All cryptographic key management practices must conform to AS 2805.6.1.

4.2 Transport Keys

4.2.1 *Approved Encryption Algorithms for Transport Keys*

DEA2 and DEA3 are the only approved algorithms for the protection of keys in transport.

4.2.2 *Minimum Key Length for Transport Keys*

(a) DEA2 keys of less than 2048 bits are to be treated as single use keys and their use is deprecated.

(b) DEA2 key lengths of less than 1024-bits are unsuitable for general use. Preferred DEA2 key lengths are equal to or greater than 2048 bits in length and should be used in all new implementations where hardware constraints do not exist.

(c) Triple DES (DEA3) may use either 128-bit or 192-bit key sizes.

4.2.3 *Key Life Cycle Practices for Transport Keys*

(a) DEA3 Key Transport Keys are single use keys only.

(b) Symmetric Key Transport Keys must be freshly generated to protect keys in transport and then securely destroyed after use.

(c) At the time of publication, DEA2 keys of size equal to or in excess of 2048 bits are deemed acceptable for a key change interval (life time) of two (2) years.

4.3 Domain Master Keys (DMK)

These keys are used within a financial institution to protect keys stored internal to the organisation.

4.3.1 *Minimum Key Length for Domain Master Keys*

Domain Master Keys must be DEA3 keys with a minimum length of 128-bits (112 effective).

4.4 IAC Interchange Cryptographic Keys[19]

4.4.1 *Introduction*

Interchange keys are used to protect financial Transactions initiated at Acquirer Terminals while in transit to the Issuer institution. Interchange keys may be either:

(a) PIN encrypting keys – used to protect the customer PIN from the point of origin to the point of authorisation. PIN encrypting keys are a specific instance of session keys;

(b) Message authentication keys – used to ensure message integrity. Message authentication keys are a specific instance of session keys;

(c) Data Protection Keys – used to provide confidentiality of messages. Data protection keys are a specific instance of session keys;

(d) Session keys – used to secure, validate and protect the financial message. Session keys can be further qualified into those used in the Terminal to Acquirer environment (Terminal session keys) or on node to node links (interchange session keys);

(e) Key Encrypting Keys (KEK)– used to protect other keys (e.g., session keys) during exchange; or

(f) Transport Keys – used to protect keys (e.g., KEKs) during transport to the partner institution.

4.4.1 *Cryptographic Algorithms*

(a) DEA3 and DEA2 are the only approved algorithms for the protection of interchange information (full details of these algorithms may be found in the Australian standards AS 2805.5.4 and AS 2805.5.3 respectively).

(b) DEA3 keys are 128 bits in length (effectively 112 bits) and are generally referred to as triple DES or 3DES keys (the corresponding encryption algorithm is specified in AS 2805.5.4). Triple DES may also be acceptably implemented using a key length of 192 bits (effectively 168 bits).

(c) DEA3 with a key length of 128 bits and DEA2 with key lengths equal to, or greater than 2048 bits are the minimum acceptable requirements for the effective protection of interchange information at the time of the issuance of this document.

(d) In accordance with AS 2805.3.1, DEA3 must be used for PIN encipherment. Acquirers who do not comply with this requirement are responsible for any Issuer loss (direct or indirect) arising from the compromise of PIN data due to a breach of this requirement.[20]

[19] Amended effective 1/1/20, version 010 r&p 002.19
[20] Amended effective 16/12/21, version 013 r&p 001.21

4.5 IAC Interchange Links[21]

4.5.1 *IAC Interchange Security Requirements[22]*

For all IAC Interchange Links, Issuers and Acquirers must ensure that: [23]

(a) security for Transactions processed over that IAC Interchange Link complies with: AS 2805.6 series;[24]

(b) security for Transactions from Terminal to Acquirer and from Acquirer to Issuer complies with: AS 2805.6 series;

(c) PIN security and encryption complies with AS 2805. 3.1 and clause 4.8 of this IAC Code Set Volume 4;[25]

(d) Key management practices comply with AS 2805.6.1;

(e) Message Authentication must apply to all IAC Interchange Links;[26]

(f) The Message Authentication Code (MAC) must be calculated using, as a minimum, a DEA 3 (128-bit) key, Triple-DES and an algorithm conforming to AS 2805.4.1; and

(g) all interchange PIN and MAC cryptographic functions must be performed within an SCM that is an Approved Device.[27]

4.5.2 *Key Management Practices – IAC Interchange Links[28]*

Clause 4.5.2 is Confidential

[21] Amended effective 1/1/20, version 010 r&p 002.19
[22] Amended effective 1/1/20, version 010 r&p 002.19
[23] Amended effective 1/1/20, version 010 r&p 002.19
[24] Amended effective 1/1/20, version 010 r&p 002.19
[25] Amended effective 29/4/16, version 003 r&p 001.16
[26] Amended effective 1/1/20, version 010 r&p 002.19
[27] Amended effective 16/12/21, version 013 r&p 001.21
[28] Amended effective 1/1/20, version 010 r&p 002.19

4.6 KEK Establishment

4.6.1 *Introduction*

(a) The security of Interchange is critically dependent on the secure installation of the Interchange Key Encrypting Keys. It is critically important that safe, sound and secure practices be adopted for the generation, handling, transport, storage and installation of interchange Key Encrypting Keys.

(b) The initial establishment of Key Encrypting Keys must employ one of the methods identified in this clause namely:

(i) AS 2805.6.6 method;

(ii) Native RSA key method;

(iii) KTK method;

(iv) KEK Component method.

(c) For those members employing AusPayNet standard Security Control Modules where RSA functionality exists, the Native RSA initialisation method is preferred.

4.6.2 *AS 2805.6.6 method*

(a) This Interchange key initialisation process employs an RSA key pair generated internally by the Security Control Module (SCM).

(b) With this method each SCM has a set of pre-generated RSA key pairs.

(c) The key exchange procedure is the following:

(i) partners exchange (via a secure channel[31]) their public RSA keys (IPK) and the associated verification codes;

(ii) each partner authenticates and installs the partner's IPK;

(iii) Key management proceeds in accordance with the requirements of AS 2805.6.6.

[31] In the absence of a secure email channel, authenticity of public keys should be achieved by some other means, for example by verifying the corresponding PVC-s through a different communication channel, such as telephone or facsimile

(d) Advantages

This method is the only mechanism providing for full automation of subsequent key changes and for that reason is preferred.

(e) Disadvantages

This method may require changes to the application if it is to be supported.

4.6.3 *Native RSA key method*

(a) This Interchange key initialisation process employs a RSA key pair generated internally by the Security Control Module (SCM).

(b) With this method each SCM has a set of pre-generated RSA key pairs.

(c) When generated on request, the Interchange Key Encrypting Key (KEKs) is signed by the native private key[32] and encrypted by the partner's public key. In this signed and encrypted format, the Interchange KEKs will be sent to the partner where it will be translated into the form required by the application (that is by encryption under the KM). For the receiving partner it will become KEK Receive.

(d) The key exchange procedure is the following:

(i) Partners exchange (via a secure channel[33]) their public RSA keys. This is a prerequisite to generate KEKs. The format of the data for the exchange of the public key uses three lines of text:

(A) the public key modulus;

(B) the public key exponent; and

(C) the public key verification code (PVC).

Note that the ASCII hex presentation of data applies.

(e) The PVC will be mutually confirmed over the telephone by the key exchange representatives:

(i) Each partner generates their KEK Send, that is cryptographically protected under RSA;

[32] Actually the hash of the key is signed.

[33] In the absence of a secure email channel, authenticity of public keys should be achieved by some other means, for example by verifying the corresponding PVC-s through a different communication channel, such as telephone or facsimile.

(ii) Each partner submits the protected KEK Send to the Interchange partner (typically by secure email). The format of the data for the exchange of the KEK uses three lines of text:

(A) the signed hash;

(B) the encrypted KEK; and

(C) the key verification code (KVC).

Note that the ASCII hex presentation of data applies.

(f) The KVC will be mutually confirmed over the telephone by the key exchange representatives.

(i) the received KEK becomes KEK Receive. KEK Receive is translated from encryption/signing under RSA(s) to encryption under KM for local key database storage;

(ii) both KEK Send and KEK Receive are stored in the required location in the key database; ensuring that the corresponding KEK KVC matches on both sides;

(iii) the interchange is started using the new Interchange KEK keys.

(g) The corresponding SCM functions are: C500 GETPUBLIC, C600 NODEKEKSEND, C610 NODEKEKREC.

(h) Advantages

(i) This method does not require any specific update/integration on the application part. i.e., the use of RSA is completely transparent to the application and therefore all Interchange parties can exchange keys through this method without any proprietary changes to their native application (as long as they have the required functions in their SCM).

(ii) There is significant current experience with this method more so than with the other two random KEK methods - this method has proved to be very efficient and reliable in practice.

(i) Disadvantages

(i) The main operational disadvantage is the dependency upon a particular (“dedicated”) security device. In a generic case there is no guarantee that the used RSA key pair, from a particular SCM device, has not changed since the last key exchange, e.g., if the device was reset or a new device installed. Therefore the interchange key (KEK) change process requires exchange of RSA keys every time. For this reason this method is currently implemented as an off-line process and as such it is not recommended for automation.

4.6.4 *KTK Method*

(a) This method relies on a transport 3DES key that is provided to the SCMs of both Interchange partners and used to encrypt the Interchange KEKs. For key loading, KTK will typically be presented in multiple XOR key components and each partner will contribute to its construction supplying at least one component.

(b) In the AusPayNet SCM specification SCMs, the functions used are D501 KEKGEN-6.3 and D502 KEKREC-6.3.

(c) When generated on request, the Interchange key (KEK Send) is encrypted under the KTK and submitted to the partner where it needs to be translated into the form required by the application (encryption under the KM). For the receiving partner it will become KEK Receive.

(d) The key exchange procedure is the following:

(i) each interchange partner generates at least one KTK component and submits it through a secure channel to the corresponding Interchange partner for loading into an SCM;

(ii) KTK is loaded by each partner;

(iii) the KVCs are verified;

(iv) each partner generates their KEK Send, that is cryptographically protected under KTK;

(v) each partner submits the protected (encrypted) KEK Send to the partner (typically by secure email);

(vi) the received KEK becomes KEK Receive. KEK Receive is translated from encryption under KTK to encryption under KM for local key database storage;

(vii) both KEK Send and KEK Receive are stored in the required location in the key database; ensuring that the corresponding KVC matches on both sides;

(viii) the interchange is re-started using the new Interchange keys.

(e) Advantages

For parties that cannot support RSA keys either functionally or by security policy, this is a simple reliable ‘traditional’ approach. Its impact to the application design is the same as for the RSA native method, i.e., either method may be used transparently to the application as long as the SCM interface utility supports the corresponding SCM calls.

(f) Disadvantages

The clear KTK components must be securely exchanged between the partners and also loaded into the SCMs through a 'secure key entry process'. They also must be securely stored e.g., in a safe. All these operational support requirements increase the operational cost of this method and security risks (of staff collusion, negligence, etc.).

4.6.5 *KEK Component Method*

(a) This method is a 'traditional' method of the interchange key initialisation and as such is supported by older Security Control Module designs. It is still maintained by many interchange partners and in particular by many smaller organizations.

(b) This method does not involve use of initial keys such as RSA or KTK but is based on direct manual storage of 3DES interchange keys in the SCM devices, therefore the interchange keys (KEKs) in this method are generated externally and are loaded into the device in components. The key material requires a secure key loading procedure and also secure storage of the key components.

(c) This method is included for 'backward compatibility' and for a fall-back situation.

(d) The key exchange procedure is the following:

(i) the partners generate interchange keys in at least two XOR components and exchange paper components using a secure channel;

(ii) the keys are loaded into the SCM device under dual control - the corresponding KVCs are noted for verification; the keys may also be encrypted under the KM for storage in the key data base;

(iii) the partners confirm the KVCs;

(iv) the paper components are stored in the secure storage (e.g., safes under dual control);

(v) afterwards, the KEKs are ready for use.

(e) Advantages

This method is still in wide spread use across the industry. For this reason and because of its manual handling nature, it is a good fallback solution.

(f) Disadvantages

The extensive use of manual procedures renders subsequent key changes, as are required under IAC Rules more difficult than some of the other methods.

4.7 IAC Interchange Lines[34]

IAC Interchange Lines must be subject to whole-of-message encryption, excluding communications headers, using at a minimum, triple-DES and a DEA 3 (128-bit)-bit key in accordance with AS 2805.5.4.

4.7.1 *IAC Interchange Line Cryptographic Management[35]*

(a) Subject to clause 4.6, the use of transport level data encryption (e.g., IPSec) is permitted subject to the following conditions:

(i) data encryption must use either triple DES with either a 112-bit or 168-bit key length, exclusive of parity bits, or AES;

(ii) the data stream must be fully encrypted with the exception of communication headers;

(iii) where IPSec is used, the system must be configured to use Encapsulating Security Payload, and authentication must be HMAC-SHA-1;

(iv) either certificates or encrypted pre-shared secrets must be used (plain text shared secrets not acceptable);

(v) tunnel termination points must be within the IA Participant's or their trusted agent's facilities;

(vi) the facility must be supported by documented device management procedures with identified roles and responsibilities and subject to internal audit as prescribed by the IA Participant's security policy;

(vii) ownership and control of end-points must reside with the terminating IA Participant;

(viii) split tunnelling is not to be used; and

(ix) the minimum Diffie-Hellman MODP group size is 1536-bits;

(x) Internet Key Exchange, if used, must be configured to only use main mode. Specifically, aggressive mode must NOT be used.

[34] Amended effective 1/1/20, version 010 r&p 002.19
[35] Amended effective 1/1/21, version 012 r&p 002.20

(b) Where encrypted shared-secrets are used, key management, including the process of key (secret) entry must comply with the requirements of AS 2805.6.1, especially the requirement that no one person must have the capability to access or ascertain any plain text secret or private key.

4.7.2 *Key Management Practices for IAC Interchange Lines* [36]

Clause 4.7.2 is Confidential

4.8 Terminal Key Management

4.8.1 *Terminal key management requirements*

For all Terminal to Acquirer Links, Acquirers must ensure that:

(a) Security for Transactions from Terminal to Acquirer complies with: AS 2805.6 series;

(b) PIN security and encryption complies with AS 2805.3.1 and 5.4;

(c) Key management practices comply with AS 2805.6.1;

(d) Message Authentication must apply to all Acquirer Links for all financial and key management messages;[39]

(e) the Message Authentication Code (MAC) must be calculated using a DEA 3 (128-bit key), or AES; and[40]

(f) an algorithm conforming to ISO 9797-1 or ISO/IEC 19772;[41]

[36] Amended effective 1/1/20, version 010 r&p 002.19
[39] Amended effective 1/1/19, version 008 r&p 002.18
[40] Amended effective 1/1/25, version 016 r&p 001.24
[41] Inserted effective 1/1/25, version 016 r&p 001.24

(g) all PIN cryptographic functions must be performed within an Approved Device;[42]

(h) an Approved device must be used for one or both of the following cryptographic operations:[43]

(i) MAC generation and verification functions; or

(ii) Encryption and decryption functionality used for privacy of communications;

(i) where MAC cryptographic functions are not performed in an Approved Device, the system components generating or verifying MACs and associated keys shall be considered part of the host system and assessed against the requirements from IAC Code Set Volume 3, clause 3.5;[44]

(j) Message Authentication Codes shall be used to protect all payment-related messages passing through all communication links between a terminal and the host system driving the terminals; and[45]

(k) for EFTPOS Terminals privacy of communication complies with AS 2805.9 or any other privacy of communication standard approved by the Management Committee.

4.8.2 *Key Management Practices*

Clause 4.8.2 is Confidential

[42] Last amended effective 1/1/25, version 016 r&p 001.24
[43] Inserted effective 1/1/25, version 016 r&p 001.24
[44] Inserted effective 1/1/25, version 016 r&p 001.24
[45] Inserted effective 1/1/25, version 016 r&p 001.24

4.8.3 *Key Rolling Process for Session Keys*

Session key roll over should occur without operator intervention and in a manner compliant with AS 2805.6.2, AS 2805.6.4 or other AusPayNet approved, Terminal key management protocol.

Next page is Annexure A

[49] Inserted effective 20/8/18, version 007 r&p 001.18
[50] Inserted effective 1/1/25, version 016 r&p 001.24

ANNEXURE A. MINIMUM EVALUATION CRITERIA FOR IP ENABLED TERMINALS [DELETED][51]

[Deleted]

Next page is Annexure B

[51] Deleted effective16/12/21, version 013 r&p 001.21

ANNEXURE B. PCI PLUS REQUIREMENTS [DELETED][52]

[Deleted]

Next page is Annexure C

[52] Deleted effective16/12/21, version 013 r&p 001.21

ANNEXURE C. DEVICE EVALUATION FAQ [DELETED][53]

[Deleted]

Next page is Annexure D

[53] Deleted effective16/12/21, version 013 r&p 001.21

ANNEXURE D. DEVICE APPROVAL PROCESS [DELETED][54]

[Deleted]

Next page is Annexure E

[54] Deleted effective16/12/21, version 013 r&p 001.21

ANNEXURE E. IAC LABORATORY ACCREDITATION CHECKLIST [DELETED][55]

[Deleted]

The next page is Annexure F

[55] Deleted effective16/12/21, version 013 r&p 001.21.

ANNEXURE F. INTRODUCTION TO DEVICE SUPPORT AND SCM FUNCTIONALITY

[Informative]

F.1 Introduction

This annexure illustrates how the functionality provided by the AusPayNet SCM may be used to provide device driving support for ATMs and POS Terminals including remote initialisation.

It is based on the use of the approved triple-DES SCM specification referred to as AusPayNet SCM specification which is at revision V5.0 at the time of writing (January 2015) This annexure illustrates a method of implementing both ATM and POS device support using AusPayNet specification Security Control Modules. Only a limited subset of the possible key management schemes and associated SCM functions are described, in particular, transaction based key management schemes are not addressed.

Description of transactions and messages is confined to cryptographic items such as keys, PIN blocks, and MACs, and excludes financial and other items.

F.2 References and Related Documentation

1. SCM Spec Specification for a Security Control Module Function Set, AusPayNet Technical Security Sub-Committee, Version 5.0, June 25th, 2013.[56]

2. AS 2805.3-2000 Electronic funds transfer - Requirements for interfaces - PIN management and security.

3. AS 2805.4.1/Amdt 1/2006 Electronic funds transfer - Requirements for interfaces - Message authentication - Mechanisms using a block cipher.

4. AS 2805.5.1-1992 Electronic Funds Transfer - Requirements for Interfaces, Part 5.1: Ciphers - Data encipherment algorithm 1 (DEA 1).

5. AS 2805.5.3-2004 Electronic funds transfer - Requirements for interfaces - Ciphers - Data encipherment algorithm 2 (DEA 2).

6. AS 2805.5.4-2000 Electronic Funds Transfer - Requirements for Interfaces, Part 5.4: Ciphers - Data encipherment algorithm 3 (DEA 3) and related techniques.

7. AS 2805.6.2-2002 Electronic funds transfer - Requirements for interfaces - Key management - Transaction keys.

8. AS 2805.6.4-2001 Electronic funds transfer - Requirements for interfaces - Key management - Session keys - Terminal to acquirer.

[56] Amended effective 20/8/18, version 007 r&p 001.18

9. AS 2805.6.5.3-2004 Electronic funds transfer - Requirements for interfaces - Key management - TCU initialization - Asymmetric.

10. NCR NDC+ Programmer's Reference Manual.

F.3 Overview

Section F.4 describes the key specifiers which the AusPayNet SCMs use to manage keys with different lengths and attributes. It also describes the key variants that are used.

Section F.5 shows how AusPayNet SCM functions can be used to perform 3DES ATM key management with double-length keys, MACing, and remote initialisation. *The AusPayNet SCM functions currently only provide support for NCR's NDC+ 3DES ATMs. Details of other ATM manufacturer's 3DES functionality are covered.*

Section F.6 shows how AusPayNet SCM functions can be used to perform 3DES POS key management (double-length session keys) with remote initialisation.

The scheme described in section F.6 is AS 2805.6.4 key management (session keys) and AS 2805.6.5.3 remote initialisation. The AusPayNet SCM also provides 3DES functions to support AS 2805.6.2 key management (transaction keys). This is not covered in this document.

For both ATM and EFTPOS Terminals, there are associated new or upgraded remote initialisation standards and associated SCM functions, which interface with the 3DES session key management functions. The IAC Manual should be consulted to determine the appropriate key lengths to be used when implementing any remote key initialisation scheme.[57]

For ATM devices, section F.5.6 describes how double-length master keys can be loaded manually instead of by remote initialisation.

For EFTPOS Terminals, section F.6.5 describes how it is possible to combine 3DES session key management with remote initialisation using the existing 512-bit RSA keys.[58][59]

F.4 Key Specifiers and Variants

The AusPayNet SCM specification introduces a new data structure: the key specifier. A key specifier allows various attributes to be associated with a key:

- key length: single, double, triple, etc.;
- keyblock encipherment algorithm: DEA, AES, etc.;

[57] Amended effective 20/8/18, version 007 r&p 001.18

[58] It is a challenge for POS Terminals with 8-bit hardware to perform signing and ciphering with 1024-bit keys. It is not unknown for a Terminal to take 11-12 minutes to perform this calculation with 512-bit keys after being sent the sponsor's public key (see F.6.4).

[59] Amended effective 20/8/18, version 007 r&p 001.18

- keyblock encipherment mode: ECB, CBC, etc. ;
- storage mode: host or SCM.

These attributes are encoded as different hexadecimal values of a one-byte key specifier format code. Thus format code 21, for example, specifies a key with the following attributes:

- key length: double (128-bit);
- keyblock encipherment algorithm: DEA;
- keyblock encipherment mode: CBC;
- storage mode: host (because this format includes an index of the KM under which the key is enciphered for storage on the host.

Many AusPayNet SCM functions allow more than one key specifier format to be used in the request or the response.

The following key specifier formats are applicable to the host-stored keys used for ATM and EFTPOS Terminal key management:[60]

Format 21 DEA CBC Enciphered key - 128-bit with KM index			
Length	**Attrib**	**Content**	**Description**
1	H	21	Format Code
1	X	i	KM index (Range 00-FF)
16	X	eKMi(K)	Enciphered key

Format 23 DEA ECB Enciphered key - 128-bit with KM index			
Length	**Attrib**	**Content**	**Description**
1	H	23	Format Code
1	X	i	KM index (Range 00-FF)
16	X	eKMi(K)	Enciphered key

Format 31 DEA CBC Enciphered key - 128-bit			
Length	**Attrib**	**Content**	**Description**
1	H	31	Format Code
16	X	eKEK(K)	Enciphered key

[60] Amended effective 20/8/18, version 007 r&p 001.18

Format 41 Cleartext key - DEA 2			
Length	**Attrib**	**Content**	**Description**
1	H	41	Format Code
1	X	n	Number (n) of 8-byte blocks in modulus
16*n	X	PK	Clear text DEA 2 public key

Format 42 Enciphered key - DEA 2 with KM index			
Length	**Attrib**	**Content**	**Description**
1	H	42	Format Code
1	X	n	Number (n) of 8-byte blocks in modulus
1	X	i	KM index (Range 00-FF)
16*n	X	eKMi(PK) or eKMi(SK)	DEA CBC Enciphered DEA 2 key. Either the public key or the private key.

Each SCM function implicitly requires keyblocks in a predetermined format.
In an SCM Spec function, the key specifier is preceded by a length prefix, which adds one or more bytes to each of the above formats. The value of the length prefix does not include its own length. It is not necessary to store or transmit the length prefix, as its value is implied by the format code. The lengths of each of the above key specifiers are as follows:

Length	**Format code**	**Key specifier**
	21	DEA CBC Enciphered key - 128-bit with KM index
18	23	DEA ECB Enciphered key - 128-bit with KM index
17	31	DEA CBC Enciphered key - 128-bit
16n + 2	41	Cleartext DEA 2 public key - n 8-byte blocks
16n + 3	42	DEA CBC Enciphered DEA 2 key - n 8-byte blocks

The following figures reflect the different way of representing key variants in the AMB and SCM Spec specifications. SCM Spec function specifications represent the repeated byte of each hexadecimal variant constant, as shown below:

AMB variant	SCM Spec variant	variant constant for ECB-enciphered keys	variant constant for CBC-enciphered keys (SCM Spec)
V1	V24	24242424242424242424242424242424	24C024C024C024C024C024C024C024C0
V2	V28	28282828282828282828282828282828	28C028C028C028C028C028C028C028C0
V3	V22	22222222222222222222222222222222	22C022C022C022C022C022C022C022C0
V4	V48	48484848484848484848484848484848	48C048C048C048C048C048C048C048C0
V5	V42	42424242424242424242424242424242	42C042C042C042C042C042C042C042C0
V6	V44	44444444444444444444444444444444	44C044C044C044C044C044C044C044C0
V7	V82	82828282828282828282828282828282	82C082C082C082C082C082C082C082C0
V8	V84	84848484848484848484848484848484	84C084C084C084C084C084C084C084C0
N/A	VA0	A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0	A0C0A0C0A0C0A0C0A0C0A0C0A0C0A0C0
V10	VAA	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	AAC0AAC0AAC0AAC0AAC0AAC0AAC0AAC0
N/A	VAC	ACACACACACACACACACACACACACACACAC	ACC0ACC0ACC0ACC0ACC0ACC0ACC0ACC0

In subsequent figures, a box such as 21 in front of a key indicates the key specifier format.

F.5 ATM Terminal - 3DES

AusPayNet have defined an ATM 3DES solution that matches to the NCR ATM NDC+ 3DES specifications. Accordingly the following sections are based on this solution.

For remote initialisation, three RSA key pairs are used. The modulus of each key pair is 2048 bits in size:

- The manufacturer's key (SK-NCR, PK-NCR);
- The host's key (SK-HSM, PK-HSM);
- The Encrypting PIN pad's key (SK-EPP, PK-EPP).

Signatures are created by signing a hash of the target key or data, allowing all of the above keys to be the same size (unlike RSA keys for POS - see F.6).

NCR nomenclature for RSA key usage is as follows:

- (key) * SKsignature of key (or data) with secret key;
- [key] PK encryption of key (or data) with public key.

F.5.1 Exchange of Public Keys between Manufacturer and Host

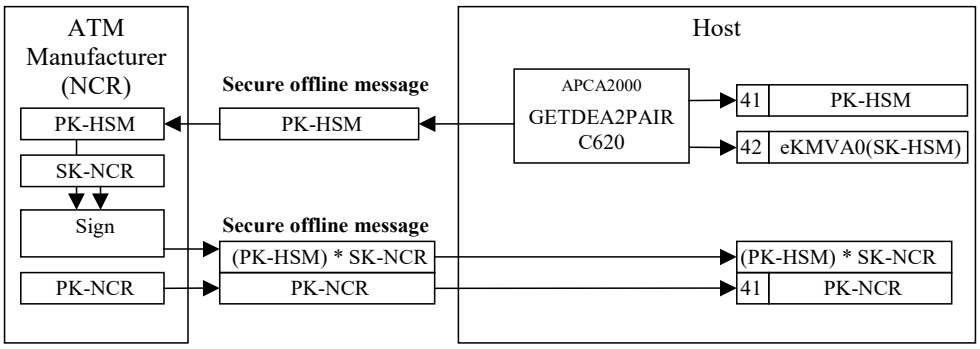


Figure 1 Exchange of RSA Public Keys between ATM Manufacturer and Host

This is a one-off offline procedure, which precedes installation of any of the manufacturer's Encrypting PIN Pads on the host's network. The keys exchanged will be used in common for all ATMs on the network (unless either party needs to replace their RSA keys in the future).

1. The host uses SCM function C620 to generate a general-purpose RSA key pair. This function is called with the size of the modulus set to 32 8-byte blocks and the public key exponent set to 65537.

2. The host sends the host's public key to the manufacturer in a secure offline message (encrypted with PGP, for example).

3. The host stores the host's public key for sending to ATMs (see F.5.3).

4. The host stores the host's encrypted secret key for signing ATM master keys (see F.5.4).

5. The manufacturer signs the host's public key with the manufacturer's secret key, and returns the signature in a secure offline message (encrypted with PGP, for example), along with the manufacturer's public key.

6. The host appends the fixed exponent 65537 to the manufacturer's public key and stores it for checking the signature of EPP public keys (see F.5.3).

7. The host stores the signed host's public key for sending to ATMs (see F.5.3).

F.5.2 Authentication by Host of ATM's EPP Serial Number

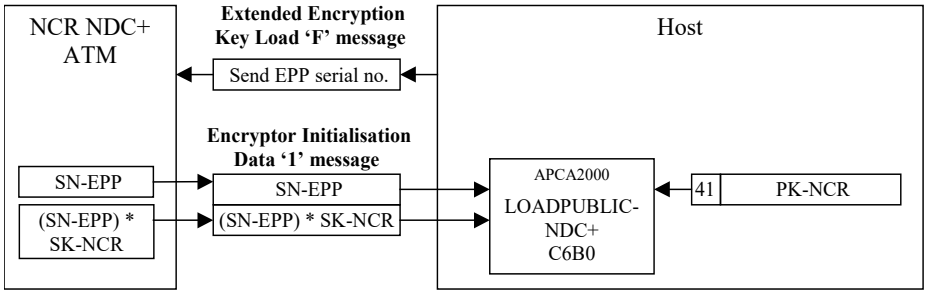


Figure 2 Authentication by Host of ATM's EPP Serial Number

1. The host requests the serial number of the ATM's EPP using an Extended Encryption Key Load Message (Message Class 3, Message Sub-class 4) with Modifier 'F' - 'Send EPP serial number and signature'.

2. The ATM returns the EPP's serial number and its signature, which were loaded into the EPP during manufacture. They are sent in an Encryptor Initialisation Data message (Message Class 2, Message Sub-class 3) with Information Identifier '1' - 'EPP serial number and signature'.

3. There is no function in the AusPayNet SCM specifically designed to verify the signature on an EPP serial number. The signature can be verified, however, by making it look like a public key.

4. The host pads the EPP serial number and appends the fixed exponent 65537 to produce a format 41 key for sending to the SCM.

5. The host decodes the EPP's serial number signature from base-94 for sending to the SCM.

6. The host uses SCM function C6B0 to verify the EPP's serial number, using the manufacturer's public key provided by the manufacturer (see F.5.1).

7. If function C6B0 indicates that the EPP's serial number signature is invalid, the host displays a console message[61].

F.5.3 Exchange of RSA Public Keys between ATM and Host

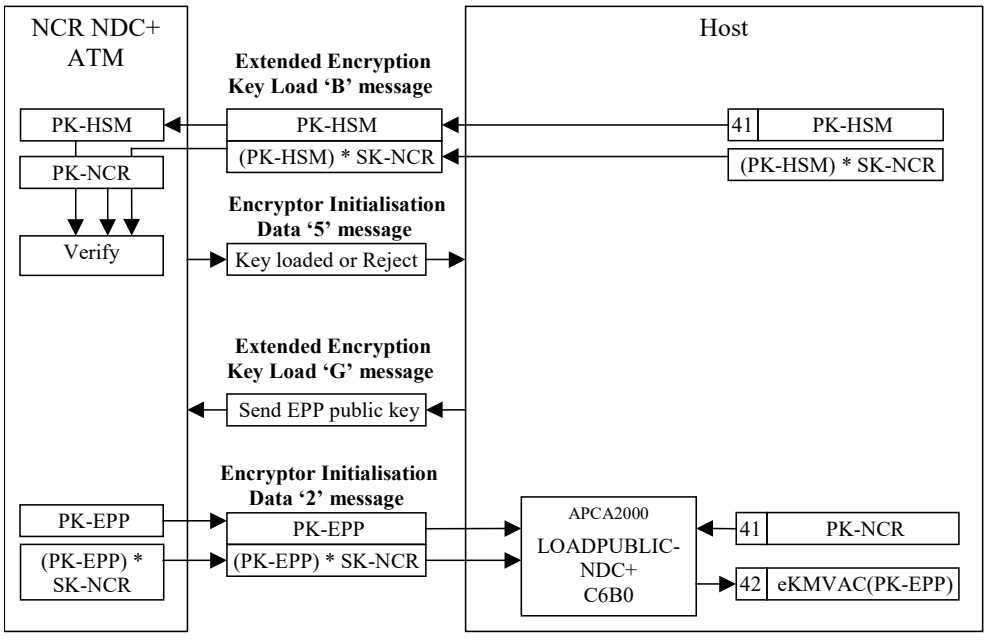


Figure 3 Authentication by Host of ATM's EPP Serial Number

[61] ATM sent invalid EPP serial number. Master key load will be unsuccessful

1. The host sends the host's public key to the ATM, along with the signature provided by the manufacturer (see F.5.1). They are sent in an Extended Encryption Key Load message (Message Class 3, Message Sub-class 4) with Modifier 'B' - 'Load HSM public key and signature'. For the message, the host removes the exponent from the host's public key, and encodes the host's public key modulus and the signature to base-94.

2. The ATM's EPP verifies the signature of the host's public key, using the manufacturer's public key which was loaded into the EPP during manufacture.

3. The ATM sends an Encryptor Initialisation Data message (Message Class 2, Message Sub-class 3) with Information Identifier '5' - 'Key Loaded'.

4. If the host does not receive this 'Key Loaded' message, it displays a console message[62] and does not proceed with the key exchange.

5. The host requests the public key of the ATM's EPP using an Extended Encryption Key Load Message (Message Class 3, Message Sub-class 4) with Modifier 'G' - 'Send EPP public key and signature.

6. The ATM returns the EPP's public key and its signature, which were loaded into the EPP during manufacture. They are sent in an Encryptor Initialisation Data message (Message Class 2, Message Sub-class 3) with Information Identifier '2' - 'EPP public key and signature'.

7. The host decodes the EPP's public key modulus from base-94 and appends the fixed exponent 65537 to produce a format 41 key for sending to the SCM.

8. The host decodes the EPP's public key signature from base-94 for sending to the SCM.

9. The host uses SCM function C6B0 to verify the EPP's public key, using the manufacturer's public key provided by the manufacturer (see F.5.1)

10. If function C6B0 indicates that the EPP's public key signature is valid, it encrypts the EPP's public key under a variant of the domain master key, and the host stores it for encrypting an ATM master key (see F.5.4).

11. If function C6B0 indicates that the EPP's public key signature is invalid, the host displays a console message[63] and does not store an encrypted EPP's public key.

[62] The key exchange failed and no keys were loaded. Master key load will be unsuccessful.
[63] Host's public key loaded on ATM but ATM's public key not loaded on host. Master key load will be unsuccessful.

F.5.4 Generation of double-length ATM Master Key

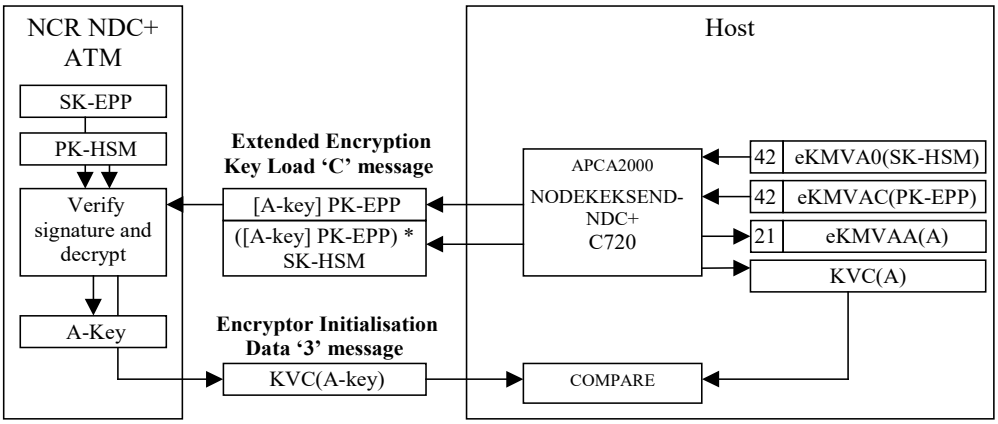


Figure 4 Generation of double-length ATM Master Key

F.5.5 Remote Initialisation

1. The host uses SCM function C720 to generate a random double-length master key. The SCM function pads the master key to 256 bytes and encrypts it with the EPP's public key received earlier (see F.5.3) and signs it with the HSM's secret key generated earlier (see F.5.1).

2. The hosts sends the encrypted master key and signature to the ATM in an Extended Encryption Key Load message (Message Class 3, Message Sub-class 4) with Modifier 'C' - 'Load initial master key (A-key) with RSA key'. For the message, the host encodes the encrypted public key and the signature to base-94.

3. The host saves the master key, encrypted under a variant of the domain master key, for encrypting session keys (see F.5.7)

4. The ATM's EPP verifies the signature using the HSM's public key received earlier (see F.5.3).

5. The ATM's EPP decrypts the master key using the EPP's secret key which was loaded into the EPP during manufacture.

6. The ATM's EPP stores the A-key for decrypting session keys (see F.5.7).

7. The ATM sends the KVC (aka KVV) of the master key to the host in an Encryptor Initialisation Data message (Message Class 2, Message Sub-class 3) with Information Identifier '3' - 'New KVV for key just loaded'.

8. The host compares the KVC with the KVC returned by SCM function C720. If they do not match, the host displays a console message[64].

[64] The master key has been loaded incorrectly on the ATM. Session key loads will be unsuccessful.

F.5.6 Manual Load

As an alternative to remote initialisation, items F.5.1 - F.5.5 can be replaced by a manual load of the double-length ATM Master Key in two double-length components:

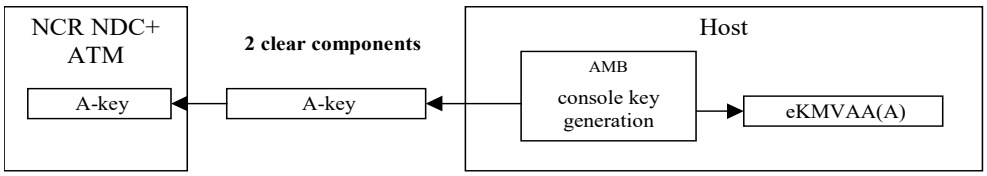


Figure 5 Generation of double-length master key for manual loading

1. The host uses a custom key generation command on the SCM console to generate a random double-length ATM master key.

2. The host stores the ATM master key, encrypted under a variant of the domain master key, for encrypting session keys (see F.5.7).

3. The host prints two clear double-length components of the ATM master key in sealed key mailers.

4. The clear components of the ATM master key are loaded into the ATM as the A-key.

F.5.7 Generation of double-length session keys

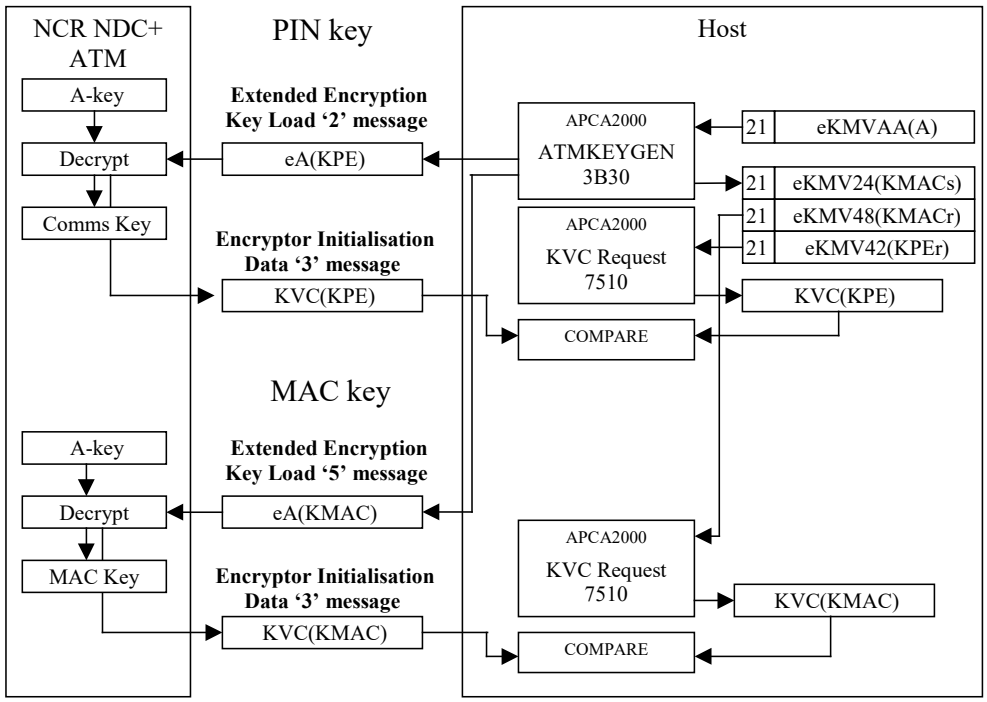


Figure 6 Generation of double-length session keys

F.5.8 PIN and MAC keys

1. The host uses SCM function 3B30 to generate random double-length PIN encryption and MAC keys. This function is called with the key length set to 2 (double) and the cipher mode set to 0 (ECB). It encrypts the session keys with the master key generated earlier (see F.5.4).

2. The host stores the PIN encryption key, encrypted under a variant of the domain master key, for decrypting PIN blocks (see F.5.11).

3. The host stores the MAC key, encrypted under variants of the domain master key, for generating and verifying MACs (see F.5.12 and F.5.13).

4. The host uses SCM function 7510 to calculate the KVCs of the PIN encryption key and the MAC key.

F.5.9 PIN key

1. The host sends the encrypted PIN encryption key to the ATM in an Extended Encryption Key Load message (Message Class 3, Message Sub-class 4) with Modifier '2' - 'Decipher new communications key with current master key'.

2. The ATM's EPP decrypts the communications key and stores it for encrypting PIN blocks see (F.5.11).

3. The ATM sends the KVC of the communications key to the host in an Encryptor Initialisation Data message (Message Class 2, Message Sub-class 3) with Information Identifier '3' - 'New KVV for key just loaded'.

4. The host compares the KVC of the communications key with the KVC returned by SCM function 7510. If they do not match, the host displays a console message[65].

F.5.10 MAC key

1. The host sends the encrypted MAC key to the ATM in an Extended Encryption Key Load message (Message Class 3, Message Sub-class 4) with Modifier '5' - 'Decipher new MAC key with current master key'.

2. The ATM's EPP decrypts the MAC key and stores it for generating and verifying MACs (see - and F.5.13).

3. The ATM sends the KVC of the MAC key to the host in an Encryptor Initialisation Data message (Message Class 2, Message Sub-class 3) with Information Identifier '3' - 'New KVV for key just loaded'.

4. The host compares the KVC of the MAC key with the KVC returned by SCM function 7510. If they do not match, the host displays a console message[66].

[65] The communications key has been loaded incorrectly on the ATM. PIN decryption will be unsuccessful.
[66] The MAC key has been loaded incorrectly on the ATM. MAC verification will be unsuccessful.

F.5.11 PIN translation with double-length session key

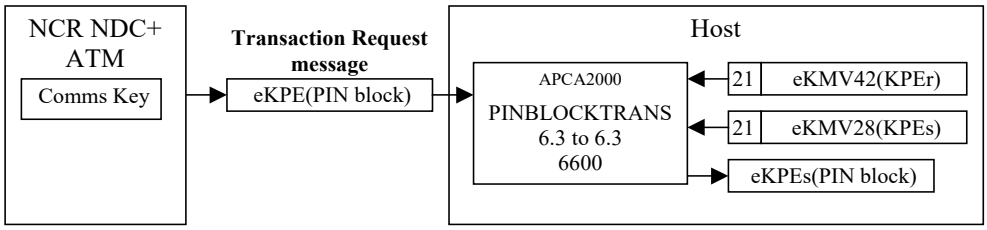


Figure 7 PIN translation with double-length session key

1. The ATM is configured to encrypt the PIN block with the ATM Comms key. Prior to encryption, the ATM's EPP formats the PIN in an AS 2805.3.1 format 0 PIN block (same as ISO format 0).[67]

2. The ATM sends the encrypted PIN block in a Transaction Request message (Message Class 1, Message Sub-class 1).

3. The host uses SCM function 6600 to translate the PIN block from encryption under the PIN encryption receive key to encryption under the PIN encryption send key. The PIN encryption receive key is the same as the ATM's Communications key (see F.5.7). The PIN encryption send key is the host's Switch Working Key[68].

4. For an 'on us' transaction, the host uses the translated PIN block to verify the PIN. For a 'not on us' transaction, the host performs a second PIN translation to encrypt it under the issuer's PIN encryption key.

F.5.12 MAC generation with double-length session key

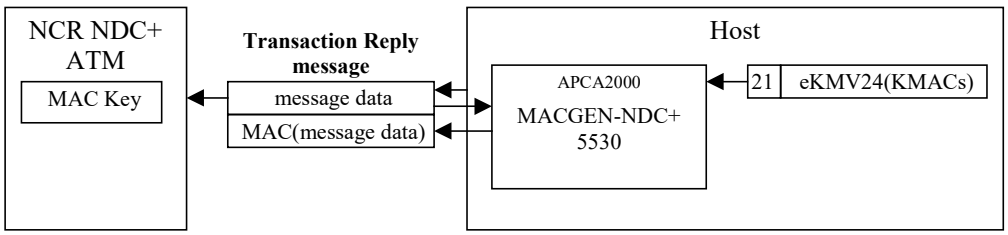


Figure 8 MAC generation with double-length session key

1. The host uses SCM function 5530 to generate the MAC. The MAC send key is the same as the ATM's MAC key (see F.5.7). The MAC algorithm used by SCM function 5530 will be standardised as MAC algorithm 3 in the amendment to AS 2805.4.1 which is under preparation by the IT-5-4 committee. The MAC is calculated over the entire message.

2. The host sends the MAC in a Transaction Reply message (Message Class 4).

[67] Amended effective 29/4/16, version 003 r&p 001.16
[68] Assuming the host uses a SWK to encrypt all PIN blocks during internal processing on the switch.

3. The ATM is configured to verify the MAC in the message data with the ATM MAC key.

F.5.13 MAC verification with double-length session key

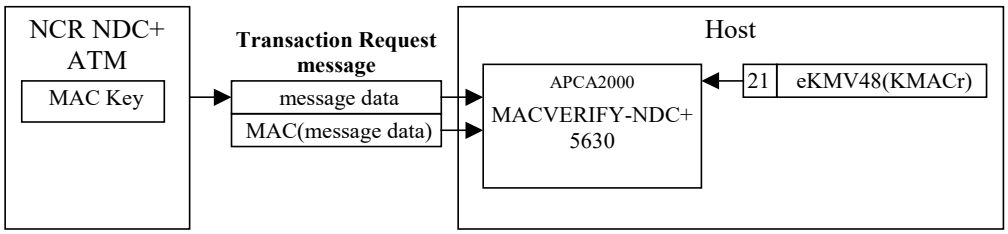


Figure 9 MAC verification with double-length session key

1. The ATM is configured to MAC the message data with the ATM MAC key.

2. The ATM sends the MAC in a Transaction Request message (Message Class 1, Message Sub-class 1).

3. The host uses SCM function 5630 to verify the MAC. The MAC receive key is the same as the ATM's MAC key (see F.5.7). The MAC algorithm used by SCM function 5630 is MAC algorithm 2 from AS 2805.4.1 (functionally equivalent to MAC algorithm 3 in ISO 9797-1). The MAC is calculated over the entire message.

F.6 EFTPOS Terminals - 3DES[69]

Key management is accomplished by the exchange of messages between Terminal and host system(s), and the execution of complementary cryptographic functions by the Terminal and host application software. The following diagrams and descriptions are indicative of the messages and functions needed to support remote initialisation and session key management. Only those message fields relevant to key management are shown.

For remote initialisation, three RSA key pairs are used. The modulus of each key pair is nominally 1024 bits in size, but the actual sizes are constrained to prevent reblocking for operations involving more than one key pair:

1. The manufacturer's key (SKman, PKman) is 1024 bits, stored on the host as 16 8-byte blocks (1024 = 16 × 8 × 8).

2. The Terminal's key (SKtcu, PKtcu) is 960 bits, so that its modulus or exponent can be signed by SKman, which is one block bigger (960 = 15 × 8 × 8).

3. The sponsor's key (SKsp, PKsp) is 896 bits, so that data (*KI, etc) enciphered with this key can be signed by SKtcu, which is one block bigger (896 = 14 × 8 × 8).

[69] Amended effective 20/8/18, version 007 r&p 001.18

F.6.1 Key Loading of a Terminal by the Manufacturer

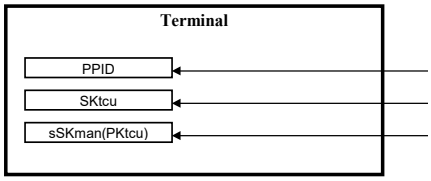


Figure 10 Key Loading of a Terminal by the Manufacturer

The following items are loaded into the Terminal by manufacturer in a secure area before the Terminal is installed in the field:

1. PPID: a unique PIN pad identifier consisting of 16 decimal digits. The PPID includes a manufacturer code, year and month of manufacturer, and a unique PIN pad serial number.

2. SKtcu: the secret key of the TCU. The modulus of this key contains 960 significant bits.

3. sSKman(PKtcu): the public key of the TCU, signed with the secret key of the manufacturer.

The TCU key pair is statistically unique for each Terminal manufactured.

F.6.2 Key Loading and Generation by the Sponsor

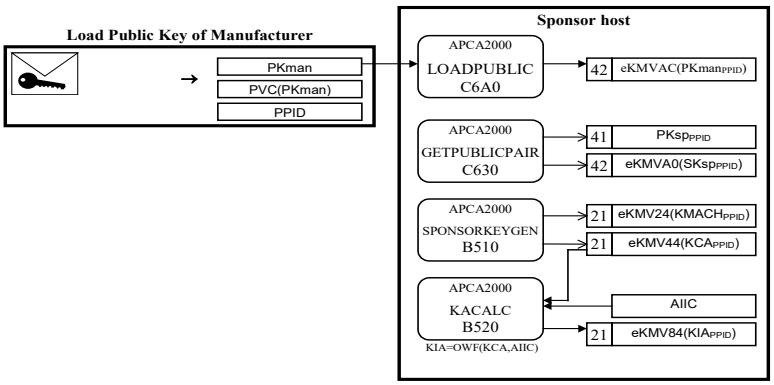


Figure 11 Key Loading and Generation by the Sponsor

1. The PPID of the Terminal and the manufacturer's public key are communicated to the sponsor in a secure manner. The sponsor loads the manufacturer's public key on the host system. The manufacturer can use the same public key for all Terminals for this sponsor, or for batches of Terminals for this sponsor, but it must not be disclosed to any other party. The modulus of this key contains 1024 significant bits.

2. The sponsor generates a public and secret key pair. The same key pair may be used for all Terminals or for batches of Terminals. The modulus of these keys contains 896 significant bits.

3. The sponsor generates a random cross acquirer key (KCA) and MAC housekeeping key (KMACH).

4. The sponsor uses the KCA to derive the sponsor's acquirer initialisation key (KIA) using the sponsor's Acquiring Institution Identification Code (AIIC).

These are all off-line procedures performed by the sponsor before the Terminal is installed.

F.6.3 Key Transmission to an Acquirer

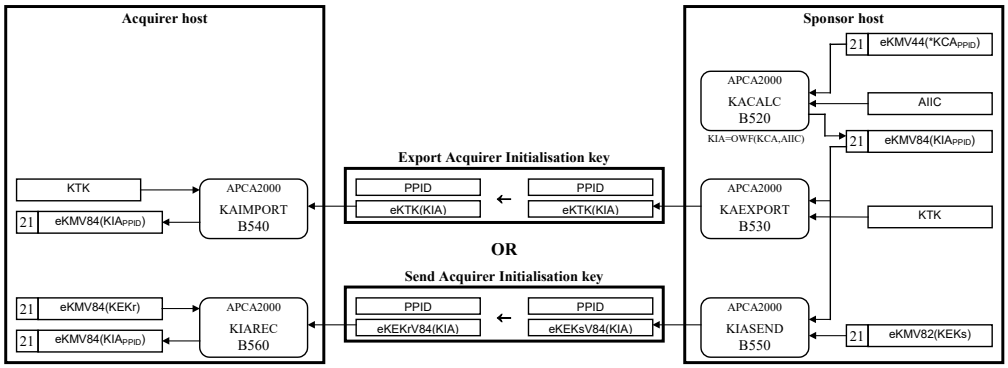


Figure 12 Key Transmission to an Acquirer

For multi-acquirer Terminals, the sponsor conveys the acquirer initialisation key (KIA) for each Terminal to each acquirer. The KIA is encrypted for transmission using either a key transport key (KTK) or a Key encrypting Key (KEK). The KTK or KEK will have been previously loaded into the SCM of sponsor and acquirer. These are off-line procedures performed by the sponsor and acquirer(s) before the Terminal is installed.

F.6.4 Remote Initialisation of a Terminal by the Sponsor

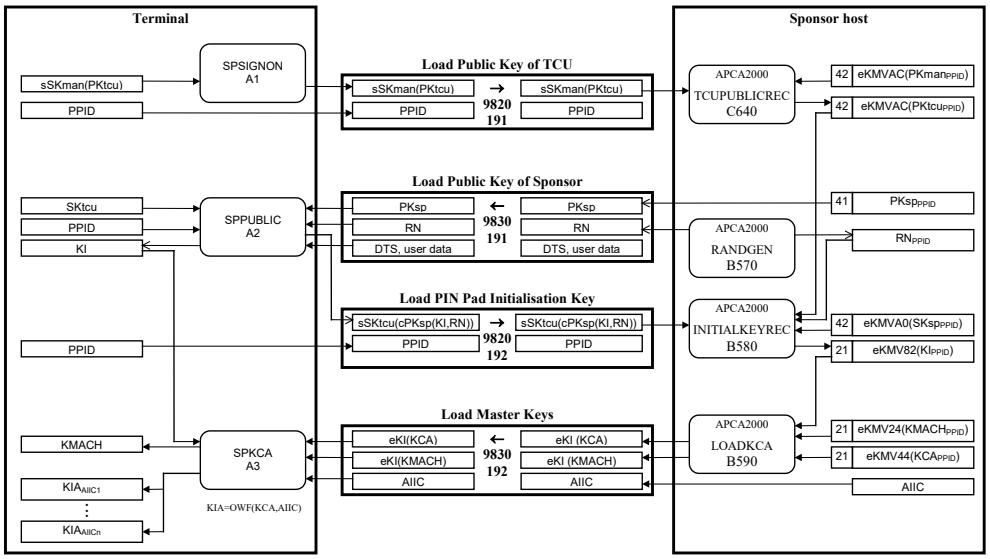


Figure 13 Remote Initialisation of a Terminal by the Sponsor

1. The Terminal sends the public key of the TCU, signed by the secret key of the manufacturer. The sponsor unsigns the TCU public key with the public key of the manufacturer.

2. The sponsor sends the public key of the sponsor, along with a random number (RN), a date time stamp (DTS), and user data.

3. The Terminal generates a random Terminal initialisation key (KI), and enciphers it with the public key of the sponsor, along with the random number RN, the PPID, the DTS, and the user data. The cipher text is signed with the secret key of the TCU and sent to the sponsor.

4. The sponsor unsigns and deciphers the message, checks the RN, PPID, DTS, and user data, and saves the KI.

5. The sponsor sends the cross acquirer key (KCA) and MAC housekeeping key (KMACH) to the Terminal, encrypted under the KI.

6. The Terminal decrypts the KCA and KMACH with KI, which is then erased. The Terminal uses KCA to derive the acquirer initialisation key (KIA) for each acquirer in its acquirer table. The KCA is then erased.

Remote initialisation is performed when a Terminal is first installed in the field. It is initiated by a password-protected command entered on the Terminal. It will be necessary to repeat the remote initialisation if the Terminal cannot log on to an acquirer using either KEK1 or KEK2, implying that the values of KEK2 have become out of step between Terminal and acquirer. This is expected to be happen infrequently - no more than once per year.

F.6.5 Remote Initialisation of a Terminal by an Acquirer

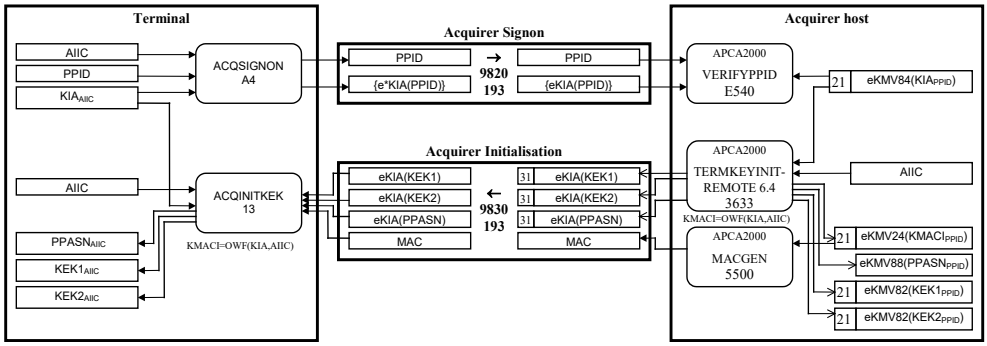


Figure 14 Remote Initialisation of a Terminal by an Acquirer

The Terminal performs this procedure for each acquirer in its acquirer table (the sponsor being the first acquirer).

1. The Terminal encrypts the PPID with the acquirer's KIA and sends the high-order 32 bits to the acquirer.

2. The acquirer verifies that the encrypted PPID is correct, thereby confirming that the Terminal is using a genuine KIA.

3. The sponsor generates random initial values for KEK1, KEK2, and the PIN pad acquirer security number (PPASN). These are encrypted under KIA and sent to the Terminal. The sponsor derives an initial MAC key (KMACI) from the KIA and the AIIC and uses it to generate a MAC for the message containing the encrypted keys.

4. The Terminal also derives KMACI and uses it to verify the MAC on the message.

5. The Terminal decrypts KEK1, KEK2, and PPASN and stores them in its key storage memory for the acquirer. The KIA for this acquirer is then erased.

Note that functions E540 and 3633 can both be supplied with the encrypted KIA eKMV84(KIA) in format 23 (ECB-encrypted) as well as format 21 (CBC-encrypted). A format 23 KIA can be constructed from the e*KMV8(*KIA) produced for 1DES POS Terminals. This would allow support of a hybrid POS Terminal which performed remote initialisation with 512-bit RSA keys but performed 3DES session key management. It is may be AusPayNet's intention, however, to discontinue support for a format 23 KIA when 3DES migration is complete.

F.6.6 Logon by a Terminal to an Acquirer

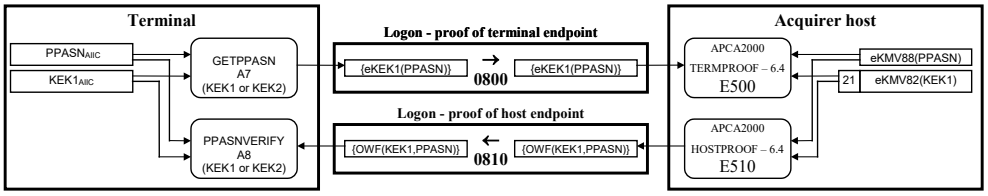


Figure 15 Logon by a Terminal to an Acquirer

1. The Terminal sends the acquirer a cryptographic function of KEK1 and PPASN which the acquirer verifies to prove that the Terminal is genuine.

2. The acquirer sends the Terminal a cryptographic function of KEK1 and PPASN which the Terminal verifies to prove that the acquirer is genuine.

This is just the cryptographic part of Terminal logon - other functions are performed by Terminal and acquirer at the same time. Proof of endpoint is normally performed with KEK1, as indicated by a flag in the messages. If proof of endpoint is unsuccessful with KEK1, suggesting that transformation of KEK1 has become out of step between Terminal and acquirer, proof of endpoint is attempted with KEK2.

A session key change, as described below, is performed immediately after a successful proof of endpoint.

F.6.7 Session Key Change by an Acquirer

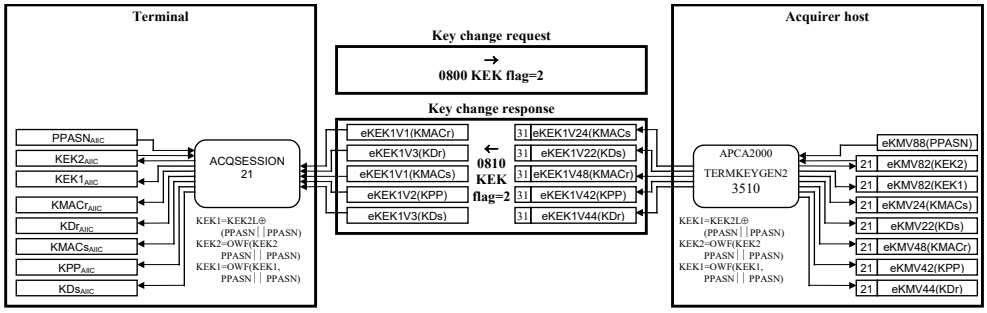


Figure 16 Session Key Change by an Acquirer

1. The acquirer responds to a key change request by generating a set of random double-length session keys, encrypting them under variants of KEK1, and sending them to the Terminal. The KEK1 is transformed by a one way function before it is used.

2. The Terminal transforms KEK1, and decrypts the session keys.

Note that the format 31 session keys generated by functions 3500 and 3510 are CBC-enciphered and that the variants of KEK1 or KEK2 are the ones shown in section F.4 (with C0 in alternate bytes).

Session key change is normally performed with KEK1, as indicated by a flag in the messages. If the key verification codes are incorrect, suggesting that transformation of KEK1 has become out of step between Terminal and acquirer, a session key change is attempted with KEK2. This causes both acquirer and Terminal to derive a new KEK1 from KEK2 and transform KEK2 with a one way function. A session key change with KEK2 is also requested after doing a KEK2 proof of endpoint during Terminal logon.

Although the key change request originates from the Terminal, each acquirer host can effectively control the frequency of session key changes by setting a “key change required” flag in a previous message to the Terminal, such as a financial transaction response.

F.6.8 Financial Transaction from a Terminal to an Acquirer

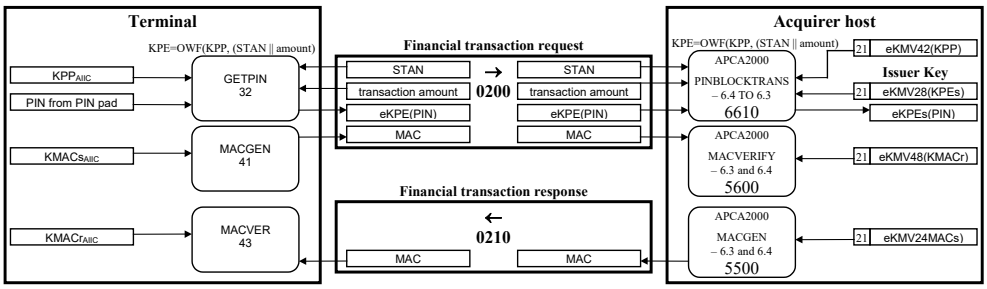


Figure 17 Financial Transaction from a Terminal to an Acquirer

1. The Terminal encrypts the PIN (entered by the customer on the PIN pad) using a PIN encryption key KPE which is derived from the PIN protection key (KPP) combined with the STAN and amount of the transaction. A MAC is generated on the financial transaction request message using the MAC session key (KMACs).

2. The acquirer verifies the MAC using the MAC session key (KMACr).

3. If the acquirer is the card issuer for the transaction (or is standing in for the card issuer), the customer's PIN is verified using the issuer's PIN verification key. Otherwise the transaction is switched to the card issuer for PIN verification - this is the case illustrated above, where the PIN block is translated to encryption under the KPEs for the issuer. The KPE used to decrypt the incoming PIN block for verification or translation is derived from the KPP, STAN, and amount, as on the Terminal.

4. The acquirer generates a MAC on the financial transaction response message using the MAC session key (KMACs).

5. The Terminal verifies the MAC on the financial transaction response message using the MAC session key (KMACr).

F.7 Glossary

3DES Triple DES encipherment, performed by three 56-bit DES operations. Same as DEA 3 if 112-bit keys are used (as they are in SCM Spec).

AES Advanced Encryption Standard - a new encryption algorithm which is the US standard to replace DES.

AMB Australian Major Banks - an industry standard set of SCM functions.

AusPayNet Australian Payments Network Limited - the industry body which regulates EFT interchange.

AusPayNet TSSC The AusPayNet Technical Security Sub-Committee - a committee of security experts from the Australian EFT industry.[70]

CBC Cipher Block Chaining - a mode of operation of DEA 1 or DEA 3 in which each 64-bit block of enciphered data is dependent on the previous block.

DEA 1 Data Encipherment Algorithm with 56-bit keys, same as DES.

DEA 3 Data Encipherment Algorithm with 112-bit keys, performed by three 56-bit DEA 1 operations.

DES Data Encryption Standard algorithm with 56-bit keys.

Double-length Key A 128-bit cryptographic key of which 112-bits are used for encipherment, 16 bits for parity checking.

[70] Amended effective 20/8/18, version 007 r&p 001.18

ECB Electronic Code Book - a mode of operation of DEA 1 and DEA 3 in which each 64-bit block of data is enciphered independently.

EPP Encrypting PIN Pad - the component of an ATM which captures PINs and performs cryptographic functions.

Host The processing system which drives ATM and EFTPOS Terminals. It runs EFT application software and sends function requests to an SCM.[71]

Interchange The exchange of EFT messages between acquirers of EFT transactions and card issuers.

Inversion In the context of proof-of-endpoint, inversion of a random number, shown by the symbol "~", means a ones complement operation, equivalent to exclusive OR with the hexadecimal constant FFFFFFFFFFFFFFFF.

KEK Key Encipherment Key - a cryptographic key used to encipher another cryptographic key.

Key Management The secure exchange and storage of cryptographic keys.

Keyblock A data structure used to store enciphered cryptographic keys.

KM A Master Key, stored in an SCM, which is used to encipher cryptographic keys stored on the host.

KM index The ordinal number of a particular master key (KM), in an SCM which can hold more than one master key.

KVC Key Verification Code. A value, derived from a cryptographic key, which is used to verify that the key is correct. Same as KVC.

KVV Key Verification Value. A value, derived from a cryptographic key, which is used to verify that the key is correct. Same as KVV.

SCM Security Control Module - a physically secure server which performs cryptographic functions.

SCM Spec The SCM specification published by AusPayNet TSSC to support 3DES.[72]

SWK Switch Working Key, key used to encrypt all PIN blocks during internal processing on an EFT switch.

Session key A cryptographic key used for a session of limited duration before being replaced, under dynamic key management.

Single-length Key A 64-bit cryptographic key of which 56-bits are used for encipherment, 8 bits for parity checking.

[71] Amended effective 20/8/18, version 007 r&p 001.18
[72] Amended effective 20/8/18, version 007 r&p 001.18

Variant A constant which is used to modify a KEK or KM before it is used to encipher another key, to enforce key separation. Different types of key are enciphered with different variants, so that they can only be used in the appropriate SCM functions.

The next page is Annexure G

ANNEXURE G. DEVICE APPROVAL PROCESS [DELETED][73]

[Deleted]

END

[73] Deleted effective 16/12/21, version 013 r&p 001.21