

Australian Standard[®]

**Electronic funds transfer—
Requirements for interfaces**

**Part 6.7: Key management—
Transaction keys—Derived unique key
per transaction (DUKPT)**



This Australian Standard® was prepared by Committee IT-005, Financial Transaction Systems. It was approved on behalf of the Council of Standards Australia on 30 November 2011.
This Standard was published on 23 December 2011.

The following are represented on Committee IT-005:

- Australian Association of Permanent Building Societies
- Australian Bankers Association
- Australian Industry Group
- Australian Payments Clearing Association
- EFTPOS Payments Australia

Additional Interests:

- Akyman Communications
 - Australia and New Zealand Banking Group
 - Coles Group
 - Eracom Technologies Australia
 - Inter-Tech Global Transaction Services
 - Mag-Tek
 - National Australia Bank
 - NCR Australia
 - Pacific Research
 - Quest Software
 - SecureNet
 - Sundial
 - Technical Security and Recovery Services
 - Witham Laboratories
-

This Standard was issued in draft form for comment as DR AS 2805.6.7.

Standards Australia wishes to acknowledge the participation of the expert individuals that contributed to the development of this Standard through their representation on the Committee and through the public comment period.

Keeping Standards up-to-date

Australian Standards® are living documents that reflect progress in science, technology and systems. To maintain their currency, all Standards are periodically reviewed, and new editions are published. Between editions, amendments may be issued.

Standards may also be withdrawn. It is important that readers assure themselves they are using a current Standard, which should include any amendments that may have been published since the Standard was published.

Detailed information about Australian Standards, drafts, amendments and new projects can be found by visiting www.standards.org.au

Standards Australia welcomes suggestions for improvements, and encourages readers to notify us immediately of any apparent inaccuracies or ambiguities. Contact us via email at mail@standards.org.au, or write to Standards Australia, GPO Box 476, Sydney, NSW 2001.

Australian Standard[®]

**Electronic funds transfer—
Requirements for interfaces**

**Part 6.7: Key management—
Transaction keys—Derived unique key
per transaction (DUKPT)**

First published as AS 2805.6.7—2011.

COPYRIGHT

© Standards Australia Limited

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher, unless otherwise permitted under the Copyright Act 1968.

Published by SAI Global Limited under licence from Standards Australia Limited, GPO Box 476, Sydney, NSW 2001, Australia

ISBN 978 1 74342 008 9

PREFACE

This Standard was prepared by the Standards Australia Committee IT-005, Financial Transaction Systems.

This Standard is Part 6.7 of the following series:

AS

- 2805 Electronic funds transfer—Requirements for interfaces
- 2805.1 Part 1: Communications
- 2805.2 Part 2: Message structures, format and content
- 2805.3.1 Part 3.1: PIN management and security—General
- 2805.3.2 Part 3.2: PIN management and security—Offline
- 2805.4.1 Part 4.1: Message authentication—Mechanism using a block cipher
- 2805.4.2 Part 4.2: Message authentication—Mechanisms using a hash-function
- 2805.5.1 Part 5.1: Ciphers—Data encipherment algorithm 1 (DEA 1)
- 2805.5.2 Part 5.2: Ciphers—Modes of operation for an n -bit block cipher
- 2805.5.3 Part 5.3: Ciphers—Data encipherment algorithm 2 (DEA 2)
- 2805.5.4 Part 5.4: Ciphers—Data encipherment algorithm 3 (DEA 3) and related techniques
- 2805.6.1.1 Part 6.1.1: Key management—Principles
- 2805.6.1.2 Part 6.1.2: Key management—Symmetric ciphers, their key management and life cycle
- 2805.6.1.4 Part 6.1.4: Key management—Asymmetric cryptosystems—Key management and life cycle
- 2805.6.2 Part 6.2: Key management—Transaction keys
- 2805.6.3 Part 6.3: Key management—Session keys—Node to node
- 2805.6.4 Part 6.4: Key management—Session keys—Terminal to acquirer
- 2805.6.5.1 Part 6.5.1: Key management—TCU initialization—Principles
- 2805.6.5.2 Part 6.5.2: Key management—TCU initialization—Symmetric
- 2805.6.5.3 Part 6.5.3: Key management—TCU initialization—Asymmetric
- 2805.6.6 Part 6.6: Key management—Session keys—Node to node with KEK replacement
- 2805.6.7 Part 6.7: Key management—Transaction keys—Derived unique key per transaction (DUKPT) (this Standard)
- 2805.9 Part 9: Privacy of communications
- 2805.10.1 Part 10.1: File transfer integrity validation
- 2805.10.2 Part 10.2: Secure file transfer (retail)
- 2805.11 Part 11: Card parameter table
- 2805.12.1 Part 12.1: Message content—Structure and format
- 2805.12.2 Part 12.2: Application and registration procedures for Institution Identification Codes (IIC)
- 2805.12.3 Part 12.3: Maintenance procedures for messages, data elements and code values
- 2805.13.1 Part 13.1: Secure hash functions—General
- 2805.13.2 Part 13.2: Secure hash functions—MD5
- 2805.13.3 Part 13.3: Secure hash functions—SHA-1
- 2805.14.1 Part 14.1: Secure cryptographic devices (retail)—Concepts, requirements and evaluation methods
- 2805.14.2 Part 14.2: Secure cryptographic devices (retail)—Security compliance checklists for devices used in financial transactions
- 2805.16 Part 16: Merchant category codes

In the AS 2805 series of Standards, definitions are specific to the Part in which they appear.

Acknowledgment is gratefully made to the American National Standards Institute for permission to reproduce certain extracts from the ANSI X9.24-1:2009 (Retail Financial Services Symmetric Key Management Part 1: Using Symmetric Techniques).

The terms ‘normative’ and ‘informative’ have been used in this Standard to define the application of the appendix to which they apply. A ‘normative’ appendix is an integral part of a Standard whereas an ‘informative’ appendix is for information and guidance only.

CONTENTS

	<i>Page</i>
1 SCOPE	5
2 APPLICATION.....	5
3 REFERENCED DOCUMENTS	5
4 DEFINITIONS	5
5 OVERVIEW	8
6 KEY MANAGEMENT SPECIFICATIONS	9
APPENDICES	
A DERIVED UNIQUE KEY PER TRANSACTION (DUKPT)	13
B KEY SET IDENTIFIERS	39
C VARIANTS OF THE CURRENT KEY.....	41

STANDARDS AUSTRALIA

Australian Standard

Electronic funds transfer—Requirements for interfaces

Part 6.7: Key management—Transaction keys—Derived unique key per transaction (DUKPT)

1 SCOPE

This Standard specifies—

- (a) a method of key management for keys used in the authentication, encipherment and decipherment of electronic messages relating to financial transactions using transaction keys;
- (b) key identification techniques; and
- (c) the data elements necessary for the transfer of security or key management information.

NOTE: Principles concerning key management and physical security are dealt with in AS 2805.6.1.1.

2 APPLICATION

This Standard may be adopted in all situations where a secure terminal to acquirer dialogue is desired in conjunction with minimally tamper resistant devices with tamper evidence characteristics as specified in AS 2805.14.1.

3 REFERENCED DOCUMENTS

The following documents are referred to in this Standard:

AS

2805	Electronic funds transfer—Requirements for interfaces
2805.3	Part 3: PIN management and security (all parts)
2805.4	Part 4: Message authentication
2805.5.4	Part 5.4: Ciphers—Data encipherment algorithm 3 (DEA 3) and related techniques
2805.6.1.1	Part 6.1.1: Key management—Principles
2805.9	Part 9: Privacy of communications
2805.14.1	Part 14.1: Secure cryptographic devices (retail)—Concepts, requirements and evaluation methods

4 DEFINITIONS

For the purpose of this Standard, the definitions below apply.

4.1 Acquirer

The institution, or its agent, that acquires the financial data relating to the transaction from the card acceptor and initiates that data into an interchange system.

4.2 Acquirer network

A network of one or more processing centres, which may represent one or more acquirers or card issuers or both.

4.3 Authentication

The act of determining that a message comes from a source authorized to originate messages of that type and that the message is as authorized.

4.4 Base derivation key (BDK)

A key that is typically common to a population of terminals and is used to derive the unique-per-terminal IK, which is injected into the TCU.

4.5 Cipher text

Enciphered information.

4.6 Data encipherment algorithm (DEA)

An encipherment algorithm designed to encipher and decipher blocks of data.

NOTE: A DEA is specified in AS 2805.5.4 (symmetric).

4.7 Data key (KD)

Generic reference to a key used to encipher/decipher message data.

NOTES:

- 1 The data key is used for messages from the terminal and the data response key is used for messages from the host.
- 2 The enciphered PIN block may be included as part of the plain text message data to be enciphered.

4.8 Decipherment

The transformation of cipher text into plain text.

NOTE: 'Decipherment' is sometimes referred to as 'decryption'.

4.9 Encipherment

The transformation of plain text into cipher text for the purpose of security, authentication or privacy.

NOTE: 'Encipherment' is sometimes referred to as 'encryption'.

4.10 Encipherment algorithm

A set of mathematically expressed rules for rendering information unintelligible by effecting a series of transformations to the normal representation of the information through the use of variable elements controlled by the application of a key.

4.11 Identification

The process whereby the identity of an individual is confirmed through the use of data or tokens previously known to be associated with that individual.

4.12 Initial key (IK)

The value that is initially loaded into the terminal, and that is used to derive, directly or indirectly, all transaction keys that will be used by the terminal. It is derived from the BDK and the unique terminal identifier and so is unique per terminal.

4.13 Key

A bit string quantity which is used for transformations between plain text and cipher text.

4.14 Key set identifier (KSI)

A value used to identify the BDK.

NOTE: The KSI is explained in Appendix B.

4.15 Key storage

A secure area used to store keys.

4.16 MAC key (KMAC)

The key used to calculate and/or verify the message authentication code associated with a message.

NOTE: The MAC key may be used for messages to and from the terminal. Alternatively, the MAC response key may be used for messages from the host.

4.17 Message authentication code (MAC)

A code appended to or included with a message for the purpose of verifying the origin of the message and for verifying that the message content has not been changed.

NOTE: The procedure for generating a MAC is specified in AS 2805.4.1 and AS 2805.4.2.

4.18 Modulo 2 addition

A mathematical operation equivalent to binary addition without carry.

NOTE: 'Modulo 2 addition' is represented by the symbol $\oplus$ and is sometimes referred to as an 'exclusive OR' operation.

4.19 Non reversible transformation

A method of generating new data (or keys) from old data (or keys) in such a way as to ensure that there is no feasible technique for determining the old data (or keys) given a knowledge of the process and the results.

4.20 One way function (OWF)

A cryptographic function that is not reversible.

NOTE: Even if the output and some of the input variables are known, it is not possible to reconstruct the entire input. Suitable one way functions are described in AS 2805.5.4.

4.21 Personal identification number (PIN)

A numeric or alphanumeric code or password made up of between 4 and 12 characters that the cardholder knows for the purpose of identification.

4.22 PIN block

A 64-bit value known as the formatted clear text PIN block and constructed in accordance with the requirements specified in AS 2805.3.1 and AS 2805.3.2.

4.23 PIN Encipherment Key (KPE)

A key used to encipher or decipher PIN blocks in a message.

4.24 Plain text

Unenciphered text.

4.25 Request message

A message initiating an action by the recipient and for which a response is expected.

4.26 Response message

A message sent in response to a request message.

4.27 Secret key

A key used with symmetric cryptographic techniques and useable only by a set of specified entities.

4.28 Security control module (SCM)

A hardware unit used to perform all cryptographic functions. Security requirements for security control modules can be found in AS 2805.14.1.

4.29 Security management information data (SMID)

The data element used to manage and control cryptographic operations.

4.30 Statistically unique

An acceptably low statistical probability of an item or code being duplicated by either chance or intent.

4.31 Terminal

Equipment that is capable of originating a transaction for input to a transaction interchange network.

4.32 Terminal cryptographic unit (TCU)

The part of a terminal that performs secure cryptographic functions.

4.33 Transaction

A collection of related messages designed to complete (as far as is possible) the intention of the initiator of the original message and normally concluded by a debit and/or credit to the cardholder's account. At a minimum, a transaction consists of a request and response message.

NOTE: Reversals carried out subsequently are considered to be separate transactions.

4.34 Transaction key

A key used to cryptographically process the transaction. If more than one key is used for different cryptographic functions, each may be a variant or a derivative of the transaction key.

5 OVERVIEW

5.1 General

This Clause provides an overview of the objectives and operation of the key management scheme described in this Standard.

5.2 Objectives of scheme

The objective of this scheme is to determine and generate (derive) keys that are unique for a given transaction. Derived unique key per transaction (DUKPT) allows the use of TCUs that do not rely completely on physical barriers and security procedures. This method is intended for use in applications where many transaction-originating TCUs, relying on DUKPT to preclude the determination of past keys, communicate with relatively few receiving SCMs, which rely upon physical barriers to preclude the determination of past keys. DUKPT is suitable for use with PIN Entry Devices (PED), where many such PIN Entry Devices operate with the same acquirer(s). This method is suitable for use between the TCU of the terminal and the SCM at the first receiving node that cryptographically processes the transaction.

DUKPT relies on the use of a BDK present only in the SCM of the first receiving node that cryptographically processes that transaction. The unique transaction keys used by the terminals' TCUs are derived from the base derivation key using only non-secret data transmitted as part of each transaction.

6 KEY MANAGEMENT SPECIFICATIONS

6.1 Methods of key management

This Clause specifies the methods that shall be used singularly or in combination for key management. Key management methods fall into two broad categories, each of which may be appropriate to specific types of TCUs, as follows:

- (a) Key management methods requiring compromise prevention controls:
These are not catered for in this Standard.
- (b) Key management methods requiring compromise detection controls:
DUKPT.

The conditions for PIN, MAC and data keys are detailed in Appendix C.

6.2 Method: DUKPT

With DUKPT, each transaction-originating TCU uses a unique key for each transaction, yet never contains any information that would allow the determination of any key previously used by this TCU, nor of any key that has been or will be used by any other transaction-originating TCU. The receiving SCM shall determine the current transaction key used by any transaction-originating TCU from the following:

- (a) Non-secret information contained in the transaction's key name.
- (b) A base derivation key which—
 - (i) shall reside in an SCM;
 - (ii) resides in one or more receiving (e.g., acquirer's) SCMs;
 - (iii) never resides in any originating (e.g., terminal's) TCU;
 - (iv) is used to generate the originating TCU's unique double length initial key using a unique key name;
 - (v) can be used to generate the unique double length initial keys for many originating TCUs; and
 - (vi) shall be a DEA 3 key.

This method shall operate at the receiving SCM, as shown in Figure 1. The key name shall utilize a key set identifier. The key name consists of three subfields. The left-most subfield is a key set identifier, which is used to select the base derivation key appropriate to the TCU originating the transaction. The second subfield is a TCU ID, and the concatenation of the key set identifier and the TCU ID is enciphered using the selected base derivation key. The result is the initial key, which had been loaded into the originating TCU (though erased from this TCU, perhaps years ago). The third subfield is a transaction counter. The originating TCU shall increase its transaction counter for each transaction. The originating TCU shall cease operation when its transaction counter overflows to zero. The receiving SCM should verify that the originator's TCU transaction counter in the key name has increased.

The initial key and the transaction counter are inputs to a non-reversible transformation process that produces the transaction key used for the current transaction. In the example implementation of this method, the transformation process requires no more than 10 DEA 3 cycles even though the transaction counter can have more than a million different values.

NOTES:

- 1 Appendix A describes an example of implementation.
- 2 The initially loaded key is a function of the base derivation key, the key set identifier, and the TCU ID. Therefore no two originating TCUs will be given the same initial key provided that no two originating TCUs with the same key set identifier have identical TCU IDs.

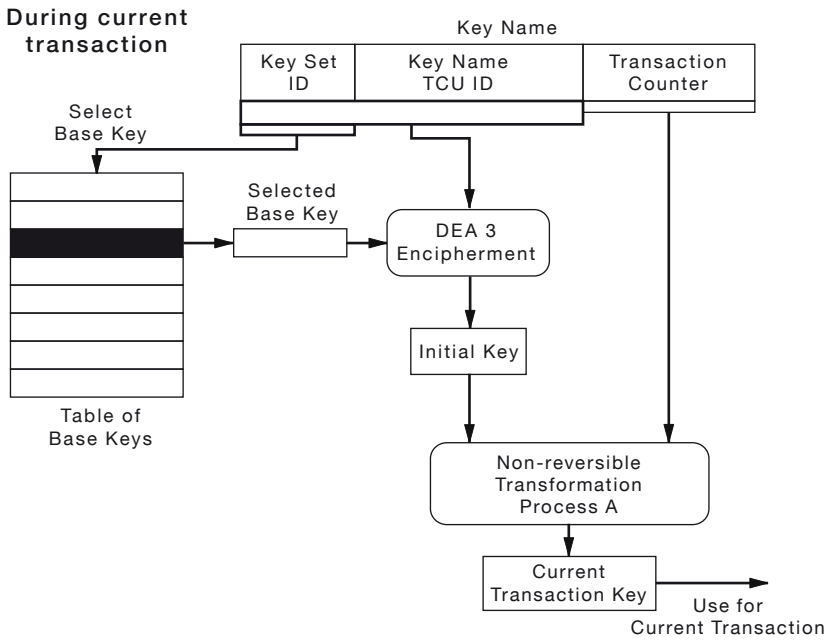


FIGURE 1 DUKPT AT RECEIVING SCM

The originating TCU shall generate and use a current transaction key such that the receiving SCM can determine that key using the process shown in Figure 2. The originating TCU shall also erase all record of the current transaction key immediately after completion of the current transaction.

Figure 2 shows how this method operates at the originating TCU. This TCU stores a number of future keys. At the beginning of a new transaction, the transaction counter (the right-most portion of the key name) is incremented, and then is used to select one of these future keys as the current transaction key. The selected key is erased from future key storage. Note that the key name is transmitted in the current transaction.

At the completion of the transaction, some number of future keys (sometimes none, sometimes one or more) are generated by non-reversibly transforming the current transaction key as a function of the transaction counter. These newly generated future keys are then stored into those locations in future key storage determined by the transaction counter. The current transaction key is then erased. Therefore, the TCU retains no information about any key used for any previous transaction.

In Figures 1 and 2 the non-reversible transformation processes ‘A’ in Figure 1 and ‘B’ in Figure 2 are different but related. Future keys are generated, stored and selected at the originating TCU in a manner such that the receiving SCM is able to determine the current transaction key.

NOTE: Appendix A provides more detailed information on the implementation of this technique.

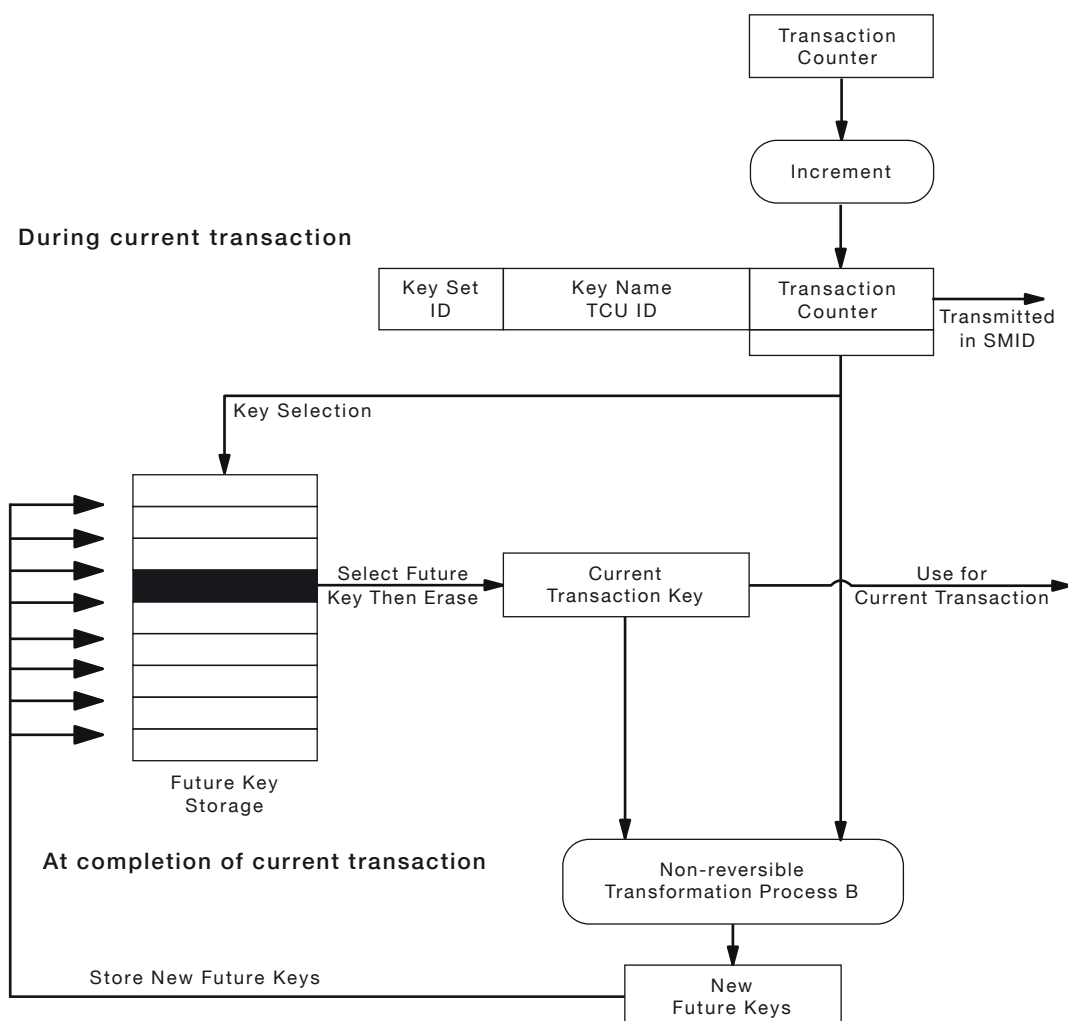


FIGURE 2 DUKPT AT ORIGINATING TCU

6.3 The security management information data (SMID) element

A SMID element may be used to convey security or key management information for use in the current transaction message or future transaction messages.

The SMID is typically used to identify the current transaction key. For example, the SMID may be used in a point-of-sale environment in which thousands of terminals interface with the same acquirer. In this case the SMID is transmitted from the terminal to the acquirer, and enables the acquirer to determine the key(s) used at the terminal for encipherment/authentication in the associated message.

For this method, if a SMID is used, the transaction key is identified as follows:

CONTROL	Optional
	Value is hex A5 or B5
KEY NAME	B5 . . . 16 mandatory
	The value used to derive the current transaction key. The key name shall incorporate a key set identification and TCU ID.

6.4 Additional key management requirements for DUKPT

Both the initial key and the first associated key name shall be loaded into the originating TCU. The originating TCU must ensure that the transaction counter is set to zero. The base derivation key used to derive this initial key shall be available to the receiving SCM before it can cryptographically process data from the originating TCU. It shall be ensured that no two originating TCUs initialized with the same KSI are initialized with the same TCUID.

6.5 Cryptographic keys synchronization

The DUKPT method is inherently self-synchronizing.

Special recovery is not required because synchronization cannot be lost.

However, other system, communication or operational failures may appear to impact cryptographic key synchronization.

6.6 Other considerations

The DUKPT method does not require an acquirer to maintain any database of key management related data, except for a relatively small number of base derivation keys that can be stored by the SCMs of this acquirer.

Because, in this method, a transaction identifies its own transaction key, it is susceptible to a fraud threat. An adversary compromises one transaction-originating TCU and then replicates the key identifier, in this case the key name (which forms the SMID transmitted by this TCU), and the key itself, in many originating TCUs. In this case this threat may be countered by two means:

- (a) The originating TCU automatically sets its transaction counter to zero when a new key is loaded. Since the compromised TCU had a non-zero transaction counter (the counter is incremented immediately after key loading), the fraudulently-loaded TCU cannot be made to simulate the compromised TCU.
- (b) An audit process is performed, perhaps offline using logged key names, to determine if two or more transaction-originating TCUs are associated with the same KSI and TCU ID. This audit process detects the use of a counterfeit TCU that has been loaded with data from a compromised TCU.

This method allows transaction-originating TCUs to be loaded with initial keys and initial key names prior to being distributed to their operational locations. If the TCUs are PIN Entry Devices that are used with POS terminals, this means that the PIN Entry Devices may be delivered to the merchant's non-secure facility with the keys already loaded. This prevents an adversary from replacing a legitimate TCU with a counterfeit unit that has all of the correct operational characteristics plus a secret data disclosing 'bug'. Since the design of the PED will ensure that the key is automatically erased if the TCU is opened, it also prevents the placing of a secret data disclosing 'bug' within a legitimate TCU. The counterfeit TCU could not contain a valid key and therefore transactions from that TCU would be rejected.

This method allows a TCU that has been loaded with a key to be subsequently associated with a transaction-originating terminal without the need for any coordination with the acquirers. If the TCUs are POS PIN Entry Devices, the merchant can install any PIN Entry Device at any time with any terminal and can replace a failed PIN Entry Device at any time with a spare unit, all without the need for manual coordination with the acquirers.

This method can be used with a 'sub-key' methodology that enables a set of TCUs, for example, a merchant's POS PIN Entry Devices, to change from one acquirer to another without the need to transfer to this second acquirer the base derivation key, which the first acquirer might use for many other merchants as well.

APPENDIX A

DERIVED UNIQUE KEY PER TRANSACTION (DUKPT)

(Informative)

A1 GENERAL

This Appendix describes the various PIN Entry Device storage areas and then specifies the PIN Entry Device functions that are used for the triple DEA method of derived unique key per transaction of processing PINs. Either the methodology as described below or its functional equivalent is performed to ensure that the key name and enciphered PIN block are generated correctly.

A2 STORAGE AREAS

A2.1 General

The PIN Entry Device maintains certain storage areas only during the PIN processing operation. Other storage areas are permanently maintained.

A2.2 PIN Processing

The contents of the following storage area relating to PIN processing are maintained only during a given PIN encipherment operation:

Account number register (12 decimal digits) Holds the 12 right-most digits, excluding the check digit, of the primary account number received from the terminal in the 'Request PIN Entry' command.

A2.3 Key management

The following storage areas relating to key management are maintained from the time of the 'Load Initial Key' command for the life of the PIN Entry Device:

- (a) *Initial Key Serial Number Register (59 bits)* Holds the left-most 59 bits of the Key Serial Number initially injected into the PIN Entry Device along with the initial PIN Encipherment Key during the 'Load Initial Key' command. The contents of this register remain fixed for the service-life of the PIN Entry Device or until another 'Load Initial Key' command.

- (b) *Encipherment Counter (21 bits)* A counter of the number of PIN encipherment operations that have occurred since the PIN Entry Device was first initialized. Certain counter values are skipped (as explained below), so that about 1 million PIN encipherment operations are possible.

NOTE: The concatenation (left to right) of the Initial Key Serial Number Register and the Encipherment Counter form the 80-bit (20 hexadecimal digits) Key Serial Number Register.

- (c) *Future Key Registers (21 registers of 34 hexadecimal digits each)* A set of 21 registers, numbered #1 to #21, used to store future PIN encipherment keys. Each register includes a 2 hexadecimal digit longitudinal redundancy check (LRC) or a 2 hexadecimal digit cyclical redundancy check (CRC).

The following storage areas relating to key management are required on a temporary basis and may be used for other purposes by other PIN processing routines:

- (i) *Current Key Pointer (approximately 4 hexadecimal digits)* Contains the address of the Future Key Register whose contents are being used in the current cryptographic operation.