One example of a method for generating the Unpredictable Number is performing an exclusive-OR operation on all the previous ARQCs, TCs, AACs, and AARs.[4] (See Annex B for details on this data element.)

2.5 Card Reading

If the terminal does not have a combined IC and magnetic stripe reader, when the magnetic stripe of the card is read and the service code begins with a '2' or a '6' indicating that an IC is present, the terminal shall prompt for the card to be inserted into the IC reader such as by displaying the 'Use Chip Reader' message.

If the terminal has a combined IC and magnetic stripe reader, when the magnetic stripe of the card is read and the service code begins with a '2' or a '6' indicating that an IC is present, the terminal shall process the transaction using the IC.

2.5.1 IC Reader

The IFD should have a pictogram near the card slot indicating how to insert the card into the IC reader.

As soon as the card is inserted into the reader, the message 'Please Wait' should be displayed to reassure the cardholder or attendant that the transaction is being processed so that the card is not removed prematurely.

When the card is inserted into the IFD, the card should be accessible to the cardholder at all times during the transaction. When the card is not accessible at all times or when the terminal has a 'tight grip' to hold the card, there should be a mechanism, for example, a button, to recall or release the card in case of terminal malfunction, even if there is a power failure. For an unattended terminal with card capture capability, where captured cards remain in the secure housing of the terminal (such as for an ATM), the card release function is not required.

When the card is inserted into the IFD, the cardholder or attendant should not be able to accidentally dislodge the card from the reader.

If the card is removed from the terminal prior to completion of the transaction, the terminal should abort the transaction and should ensure that neither the card nor the terminal is damaged. The message 'Processing Error' should be displayed. (For additional requirements on abnormal termination of transaction processing, see the *Integrated Circuit Card Specification for Payment Systems*.)

[4] This exclusive-OR operation is performed at each GENERATE AC response on the current application cryptogram and the previous exclusive-OR result, which is stored in the terminal.

2.5.2 Exception Handling

When an attended terminal attempts and fails to read the ICC but the magnetic stripe of the card is successfully read, the terminal shall set the POS Entry Mode Code in the transaction message(s) to 'Magnetic stripe read, last transaction was an unsuccessful IC read' if the service code on the magnetic stripe indicates that an IC is present.[5]

[5] This does not imply that the terminal shall support this ISO 8583:1987 data element. An issuer or an acquirer may define an equivalent data element. The specific code will be set by individual payment systems.

3. Physical Characteristics

Physical characteristics vary depending on the intended usage of the terminal, the environment at the point of transaction (including its security), and the terminal configuration.

3.1 Key Pad

A terminal should have a key pad for the entry of transaction-related data and its functional operation. The key pad shall support one or more types of keys:

- Numeric: '0' - '9'
- Alphabetic and special: For example, 'A' - 'Z', '*', '#',
- Command: 'Cancel', 'Enter', 'Clear'
- Function: Application-dependent keys, such as a selection key, 'F1', 'F2', 'Backspace', 'Escape'

A key pad may consist of a single key, such as a function key that could be a button on a vending machine to indicate selection of an application or to indicate that a receipt is to be printed.

A touch screen is considered to be a key pad (see section 4 for security requirements).

3.1.1 Command Keys

Command keys are used to control the flow of data entry by the cardholder or attendant. The description of the command keys is as follows:

Enter	Confirms an action
Cancel	Either cancels the whole transaction or, if no 'Clear' key is present, cancels the operation in progress
Clear	Erases all the numeric or alphabetic characters previously entered

The following colours, if used, shall be reserved for the command keys, either for the lettering or for the keys themselves:

Enter	Green
Cancel	Red
Clear	Yellow

When the command keys are horizontally arranged, the 'Cancel' and 'Enter' keys should be located on the bottom row of the key pad, and 'Cancel' should be the furthest key left and 'Enter' should be the furthest key right. When the command keys are vertically arranged, 'Cancel' should be the uppermost key and 'Enter' the lowest key.

3.1.2 PIN Pad

The terminal should be designed and constructed to facilitate the addition of a PIN pad, if not already present, such as having a serial port.

If the terminal supports PIN entry, a separate key pad may be present for PIN entry or the same key pad may be used for both PIN entry and entry of other transaction-related data. The PIN pad shall comprise the numeric and 'Enter' and 'Cancel' command keys. If necessary, the command key for 'Clear' may also be present.

The numeric layout of the PIN pad shall comply with ISO 9564 as shown in Figure I-4, except for cardholder-controlled terminals such as personal computers (PCs), where the keyboard may contain a numeric key pad in a different format for PIN entry. An example of the placement of the 'Cancel' and 'Enter' keys on the bottom row is shown in Figure I-4

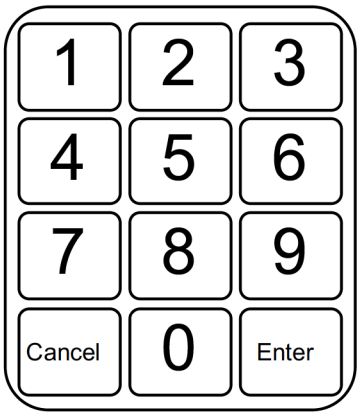

Figure I-4 - PIN Pad Layout

The key for '5' should have a tactile identifier (for example, a notch or raised dot) to indicate to those whose sight is impaired that this is the central key from which all others may be deduced.

3.2 Display

A display is used to help the cardholder or attendant monitor transaction flow and data entry, validate transaction-related data, and select options.

An attended terminal shall have a display for the attendant and may have an additional display for the cardholder, such as when a PIN pad is present. In order that different information may be displayed and different languages used for the attendant and cardholder, it is recommended that an attended terminal has two separate displays.

An unattended terminal should have a cardholder display.

At a minimum, the message display shall be capable of displaying at least 32 alphanumeric characters (two lines of 16 positions each). The two lines of 16 characters should be simultaneously displayed. To facilitate the display of different languages used in different geographical areas, the terminal should support a graphic display.

A terminal capable of supporting several applications should have a display that can provide cardholder application selection by allowing the 16-character Application Preferred Name(s) or Application Label(s) stored in the ICC to be displayed.

3.3 Memory Protection

Software as well as data initialised in the terminal or any part of the terminal, including cryptographic keys, shall not be erased or altered for the period of time the software and data are valid.

When the terminal supports batch data capture, the captured transactions and advices stored in the terminal shall not be erased or altered until the next reconciliation with the acquiring system.

3.4 Clock

Offline-only terminals and offline terminals with online capability shall have a clock with the local date and time. The clock should be capable of maintaining the time accurate to 1 minute per month.

The date is used for checking certificate expiration date for data authentication and application expiration/effective dates for processing restrictions. The time may be used for assuring transaction identification uniqueness as well as for input to the application cryptogram algorithm.

3.5 Printer

A terminal should have a printer for receipt printing. If present, the printer shall be able to print at least 20 alphanumeric characters per line in order to print the Application PAN on the receipt (see section III-1.4, of this specification).

Cardholder-controlled terminal (Terminal Type = '3x') need not include a printer.

3.6 Magnetic Stripe Reader

In addition to an IC reader, a terminal shall be equipped with a magnetic stripe reader, except when payment system rules indicate otherwise. These rules will cover situations when a magnetic stripe reader is not required or not allowed for a financial institution- or merchant-controlled terminal (Terminal Type = '1x' or '2x'). A cardholder-controlled terminal (Terminal Type = '3x') need not include a magnetic stripe reader.

The magnetic stripe reader shall be able to read the full track 1 and/or track 2 and process according to the payment system rules.

4. Security Requirements

This section describes the general requirements for handling sensitive data, such as plaintext PINs and plaintext keys. More specifically, it addresses PIN pad requirements and key management requirements for both the secret keys for a symmetric algorithm and the public key for an asymmetric algorithm.

This section makes no provision for the secure handling of messages and data between the ICC and the relevant terminal components.

4.1 Tamper-Evident Devices

A tamper-evident device shall ensure that in its normal operating environment the device or its interface does not disclose or alter any sensitive data that is entering or leaving the device or that is stored or processed in the device. (See ISO 13491 for further requirements for tamper-evident devices.)

When a tamper-evident device is operated in a securely controlled environment, the requirements on device characteristics may be reduced since protection is provided by the controlled environment and the management of the device.

4.1.1 Physical Security

A tamper-evident device shall be designed to restrict physical access to internally stored sensitive data and to deter theft, unauthorised use, or unauthorised modification of the equipment. These objectives generally require the incorporation of tamper-resistant, tamper-detection, tamper-indication, or response mechanisms, such as visible or audible alarms.

A tamper-evident device, when not in use, shall contain no sensitive information except unused cryptographic keys. It may be penetrated without loss of security, provided that this penetration is detected before the device and the stored cryptographic keys are again placed into operational use. If the device is designed to allow internal access, erasure of sensitive data must be immediately accomplished when the device is tampered with. A tamper-evident device depends on the detection by the user of attacks on its physical security. Therefore, it shall be so designed and have sufficient tamper-evident features that it shall be obvious to a user when it has been tampered with.

The device shall be designed and constructed so that:

- It is not feasible to penetrate the device to make any additions, substitutions, or modifications to the hardware or software of the device; or to determine or modify any sensitive data and subsequently re-install the device, without requiring specialised skills and equipment not generally available, and without damaging the device so severely that the damage has a high probability of detection.

- Any unauthorised access to or modifications of sensitive data that are input, stored, or processed is achieved only by actual penetration of the device.
- The casing is not commonly available, to deter the manufacture of 'look-alike' counterfeit copies from commonly available components.
- Any failure of any part of the device does not cause the disclosure of secret or sensitive data.
- If the device design requires that parts of the device be physically separate and processing data or cardholder instructions pass between these separate components, there is an equal level of protection among all parts of the device.
- Integration of different device parts into a single tamper-evident housing is the necessary condition for exchanging sensitive data such as plaintext PINs.

4.1.2 Logical Security

A tamper-evident device shall be designed that no single function, nor any combination of functions, can result in disclosure of sensitive data, except as explicitly allowed by the security implemented in the terminal. The logical protection shall be sufficient so as to not compromise sensitive data, even when only legitimate functions are used. This requirement can be achieved by internal monitoring of statistics or imposing a minimum time interval between sensitive function calls.

If a terminal can be put into a 'sensitive state', that is, a state that allows functions that are normally not permitted (for example, manual loading of cryptographic keys), such a transition shall require the assistance of two or more trusted parties. If passwords or other plaintext data are used to control transit to a sensitive state, the input of such passwords shall be protected in the same manner as other sensitive data.

To minimise risks resulting from the unauthorised use of sensitive functions, the sensitive state shall be established with limits on the number of function calls (where appropriate), and a time limit. After the first of these limits is reached, the device shall return to normal state.

A tamper-evident device shall automatically clear its internal buffers at the end of a transaction or in a time-out situation.

4.2 PIN Pads

A PIN pad shall be a tamper-evident device. It shall support entry of a 4-12 digit PIN. When a display is present on a PIN pad, an indication of the entry of each digit shall be displayed. However, the values of the entered PIN shall not be displayed or disclosed by visible or audible feedback means, in accordance with ISO 9564-1.

When the terminal supports offline PIN verification, the IFD and PIN pad shall either be integrated into a single tamper-evident device or the IFD and PIN pad shall be two separate tamper-evident devices.

- If the IFD and PIN pad are integrated, the PIN pad does not encipher the offline PIN.
- If the IFD and PIN pad are not integrated, the PIN pad shall encipher the offline PIN according to ISO 9564-1, and the IFD shall decipher the offline PIN.

In either case, the plaintext PIN is transmitted to the card.

During offline PIN verification, the VERIFY command shall be generated by the IFD.

If the terminal supports online PIN verification, when the PIN is entered, the PIN shall be protected upon entry by encipherment according to ISO 9564-1, and the terminal shall transmit the PIN according to the payment system's rules.

The prompt for PIN entry messages displayed on the PIN pad shall be generated by the PIN pad.[6] This does not imply that only PIN-related messages may be displayed on the PIN pad, although those messages shall be authorised by the PIN pad prior to display. The PIN pad shall reject any unauthorised message display.

For an attended terminal, the amount entry process shall be separate from the PIN entry process to avoid accidental display of a PIN on the terminal display. In particular, if the amount and PIN are entered on the same key pad, the amount shall be validated by the cardholder before PIN entry is allowed.

The PIN pad shall be designed to provide privacy and confidentiality so that, during normal use, only the cardholder sees the information entered or displayed. The PIN pad shall be installed or replaced so that its immediate surroundings allows sufficient privacy to enable the cardholder to enter a PIN with minimum risk of the PIN being revealed to others.

The PIN pad shall automatically clear its internal buffers when either of the following conditions occur:

- Upon completion of the transaction.
- In a time-out situation, including when an inordinate period of time has elapsed since a PIN character was entered.

[6] This does not apply to PIN pads operated in a secure environment such as an ATM.

THIS PAGE LEFT INTENTIONALLY BLANK

Part II

Software Architecture

1. Terminal Software Architecture

This section is intended to provide insight for terminal manufacturers into the future direction of the payment system applications and the consequent requirements for terminal functionality. While terminals without this functionality may operate satisfactorily in today's environment, changes in that environment will enhance the longevity of and provide functional advantages to terminals incorporating the software design principles in this section.

1.1 Environmental Changes

In today's environment, support of payment system functions is provided in the typical POS terminal by one or possibly two applications based on the limited data available from the magnetic stripe of a payment system card. Differences in cards presented are largely contained in host systems and are usually transparent to the terminal software.

The ICC replaces this environment with cards that may have multiple diverse applications, with significantly larger amounts of data representing a large number of options that must be interpreted by the terminal. The typical terminal will support multiple applications, with varying degrees of similarity. Applications may be modified annually, presenting additional challenges to software migration in the terminal. New applications will almost certainly be added during the life of a terminal. There will be a need to add applications efficiently and without risk to existing applications. Modification or addition of applications should be done in such a way that unaffected applications need not be recertified. Code should be reusable and sharable with adequate security controls to accomplish such migration with efficiency and integrity.

Greater differentiation between the payment systems should be anticipated at the terminal, expressed by data contained within the ICC. This may (and probably will) be carried down to regional and even issuer levels, requiring the terminal to keep a library of routines available for selection by the card. The terminal may support only a subset of alternative routines, but terminals that support more will be at an advantage in the marketplace.

At the level of this specification, the payment systems view two alternative software architectures as providing the capabilities required. These two alternatives are called the 'Application Program Interface (API)' and the 'Interpreter' approaches.

1.2 Application Libraries

With either the API or the interpreter approach, the terminal should have the ability to maintain an application library of modules or routines that may be dynamically incorporated into the processing of a given transaction. Modules in the application library may be complete application programs, or they may be subroutines to be called upon at the direction of data within the terminal or the ICC. In the case of an interpreter capability, these modules will be code, written in a virtual machine instruction set implemented within the terminal, to be interpreted by the terminal control program. In the case of the API approach, modules will be object code written to the specific terminal architecture.

In either case, modules within the application library may be dynamically invoked either by logic with the terminal application software or under the direction of referencing data kept within the ICC. The format and specification of external references are under control of the individual payment systems.

A terminal may contain several libraries, some accessible to all applications and some restricted to particular applications or payment systems.

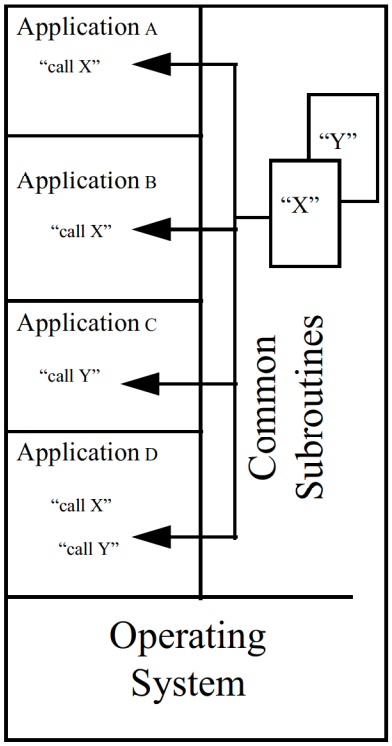

Figure II-1 - Terminal Software

1.3 Application Program Interface

This section describes a terminal software architecture through which application programs can make use of a set of essential and frequently used functions provided in terminals through a standard interface - the API.

The API takes the form of a library of functions that can be used by all applications stored in the terminal. The functions in the library may be dynamically linked into the application programs that use them.

The provision of these functions as a library in the terminal has a number of advantages:

- Each application program in the terminal does not need to include the same code to implement standardised functionality. The implementation of only one copy of code in each terminal to perform this functionality is very efficient in terminal memory.

- Application programs do not need to take account of particular terminal hardware configurations, as these will be transparent to the application program

at the API. The implications of a particular terminal's hardware implementation are embedded within the code of the library function that has been certified for that terminal.

- Certification of new terminal application programs will take place against the standardised and approved API function library for a particular terminal and does not require the re-certification of existing terminal applications programs (as would be the case with a single terminal program). The verification of firewalls between application programs is considerably eased by this architecture.

While a single library of functions is used to construct the API, the library contains functions in two broad classes:

- Functions that implement the application selection functionality described in the *Integrated Circuit Card Specification for Payment Systems* (for example, static data authentication)

- Functions that implement essential and frequently used terminal hardware functionality (for example, display, get key entry, etc.)

Functions in the library may use other functions within the library. For example, static data authentication may use a terminal hardware function to read data from an application on the card.

Functions in the library may be written using either terminal dependent object code or a more general virtual machine instruction set.

1.4 Interpreter

1.4.1 Concept

The purpose of this section is to describe the general architecture underlying an interpreter implementation and give a brief overview of how it relates to the future environment for payment system applications.

Use of ICC technology necessitates altering the firmware in all terminals that accept ICCs. To facilitate this transition, an interpreter may be implemented as a software system that is compact, efficient, and easy to maintain and enhance for future payment system needs. The name arises from the capability of a terminal to contain central processing unit (CPU)-independent application programs and plugs that can be interpreted during a transaction to determine the terminal's behaviour.

An interpreter implementation defines a single software kernel, common across multiple terminal types. This kernel creates a virtual machine that may be implemented on each CPU type and that provides drivers for the terminal's input/output (I/O) and all low-level CPU-specific logical and arithmetic functions. High-level libraries, terminal programs and payment applications using standard kernel functions may be developed and certified once; thereafter, they will run on

any conforming terminal implementing the same virtual machine without change. Therefore, a significant consequence of an interpreter is a simplified and uniform set of test and certification procedures for all terminal functions.

To summarise, interpreters provide the following major benefits:

- A kernel with generalised ICC support functions, to be installed in each terminal only once. The kernel lifetime is expected to match that of the terminal (7-10 years).
- One version of the terminal software kernel across multiple processor and terminal types. Therefore, only one certification and validation is needed for software libraries, terminal programs, and payment applications on the set of terminal types supported using a common interpreter/virtual machine.
- Terminal kernel certification independent of applications, so certification only needs to be performed once for each terminal type using a common interpreter/virtual machine. A terminal type is defined as a specific configuration of terminal CPU and I/O functions.
- Support for CPU-independent plugs that can be interpreted during a transaction to enhance a terminal's behaviour. CPU independence means that only one certification and validation is needed for this code.

1.4.2 Virtual Machine

The application software in every terminal using the interpreter approach is written in terms of a common virtual machine. The virtual machine is a theoretical microprocessor with standard characteristics that define such things as addressing mode, registers, address space, etc.

The virtual machine accesses memory in two areas: code space and data space. All code accesses are internal to the virtual machine only and are not available to programs; the memory fetch and store operators access data space only. Translated program code only exists in code space. No terminal software (libraries or other functions external to the kernel) can make any assumptions regarding the nature or content of code space or attempt to modify code space in any way. This restriction, plus the complete absence of a symbol table, adds significantly to program security.

1.4.3 Kernel

A kernel contains all functions whose implementation depends upon a particular platform (CPU and operating system). It includes a selected set of commands, plus a number of specialised functions, such as terminal I/O support and program loader/interpreter support.

1.4.4 Application Code Portability

Virtual machine emulation may be accomplished by one of three methods: interpreting virtual machine instructions, translating the virtual machine language into a directly executable 'threaded code' form, or translating it into actual code for the target CPU. The latter two methods offer improved performance at a modest cost in complexity.

The kernel for each particular CPU type is written to make that processor emulate the virtual machine. The virtual machine concept makes a high degree of standardisation possible across widely varying CPU types and simplifies program portability, testing, and certification issues.

Programs may be converted to an intermediate language, between the high level source language used by the programmer and the low-level machine code required by the microprocessor, and subsequently transported to the target terminal to be processed by the terminal into an executable form.

1.5 Plugs and Sockets

One function of ICCs is to improve transaction security by incorporating and managing enciphered data and participating actively in the transaction validation process. Under this concept, the payment systems define a number of procedures (referred to as 'sockets') that may be inserted by the application programmer (and hence under acquirer control and under payment system supervision) to act as placeholders for the addition of enhancing code during transaction processing.

Sockets are intended to be placed at various points in existing terminal applications or even in the terminal program itself. They are used to refer to library functions and may even occur inside a library function if a payment system foresees the need to change the way a library function operates.

Sockets are initialised to default behaviours. If no further action is taken by the terminal program, the default behaviour of these procedures will be to do nothing when they are executed.

Plugs are executable code, written in the machine language or virtual machine instruction set supported by the terminal, that may be inserted at points defined by sockets to enhance the default terminal logic. Plugs may already exist in the terminal to be invoked under control of data in the ICC and logic in the terminal. Plugs may also come from an input device (such as the ICC or a host system connected to the terminal), but only if agreed by the payment system, issuer, acquirer, and merchant. Special care may be required for ICC plugs if they can modify a socket's behaviour or be placed in the program flow prior to successful card authentication.

At the conclusion of a transaction, the sockets are restored to their original application default behaviours.

The proposed terminal architecture does not propose that ICCs contain entire applications but only plugs that enhance existing terminal applications.

Figure II-2 illustrates the relationship between plugs and sockets.

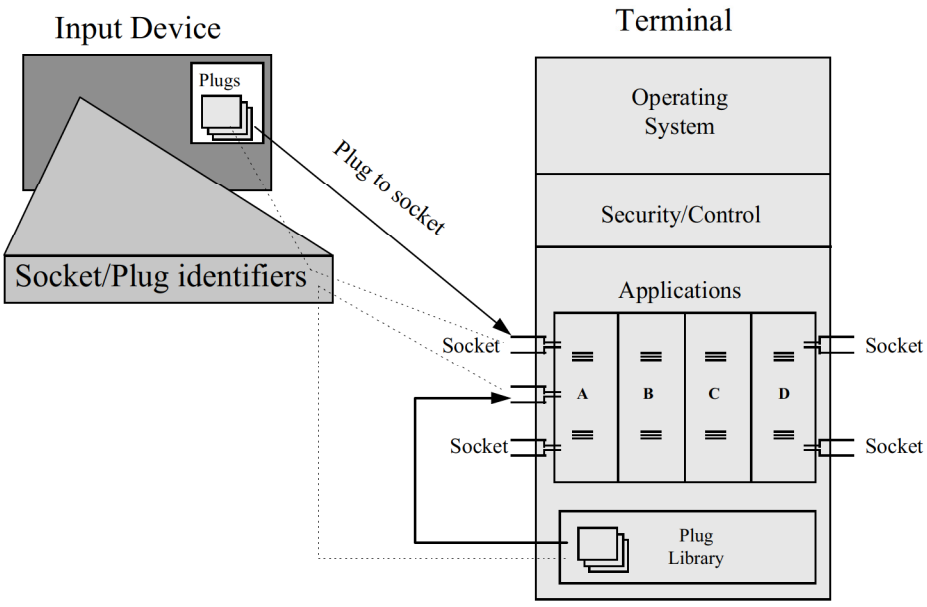

Figure II-2 - Socket/Plug Relationship

2. Software Management

A means of software upgrade shall be supported wherever this is not in conflict with national legal restrictions. The software upgrade may be facilitated from a remote site over a network or locally.

Prior to accepting new software, the terminal shall:

- Verify the identity of the party loading the software, since only software issued by the terminal manufacturer, owner, or a third party approved by the owner or acquirer can be loaded in the terminal.
- Verify the integrity of the loaded software.

When both tests are successful, the terminal shall notify the party loading the software whether the load was successfully performed or not.

To facilitate ICC application upgrade from one version to another, the terminal should be able to support at least two versions of the ICC application, as identified by the terminal's Application Version Numbers.

3. Data Management

The data elements listed in this section shall be initialised in the terminal or obtainable at the time of a transaction (definitions for these data are in Annex B). There may be additional data elements required for initialisation, such as those currently used for magnetic stripe processing.

Whenever a data element is initialised or updated, data integrity shall be assured.

Data elements resident in the terminal shall be under the control of one of the following parties:

- Terminal manufacturer: For example, IFD Serial Number
- Acquirer (or its agent): For example, Merchant Category Code
- Merchant: For example, Local Date and Local Time (these may be controlled by either the merchant or acquirer)

The terminal shall be constructed in such a way that:

- Terminal Capabilities and Additional Terminal Capabilities are initialised in the terminal before the terminal is placed in its operational state.
- Terminal Type is initialised in the terminal at the moment of installation.
- Terminal Capabilities, Additional Terminal Capabilities, and Terminal Type cannot be modified unintentionally or by unauthorised access.
- Whenever the terminal's capabilities are updated or modified, Terminal Capabilities, Additional Terminal Capabilities, and Terminal Type are accurately updated.

The terminal should be constructed in such a way that the data which is under control of the acquirer is only initialised and updated by the acquirer (or its agent).

3.1 Application Independent Data

The following data elements are application independent and shall be unique to the terminal (see section I-1.3, of this specification for different terminal configurations):

- Local Date
- Local Time
- Terminal Country Code
- Transaction Sequence Counter

The following data elements are application independent and may be specific to each device constituting the terminal, such as a host concentrating a cluster of devices (see Figure I-2 in Part I of this specification for an example):

- Additional Terminal Capabilities
- IFD Serial Number
- Terminal Capabilities
- Terminal Type

The terminal shall have parameters initialised so that it can identify what language(s) are supported to process the card's Language Preference (see section II-1.1, of this specification).

3.2 Application Dependent Data

The following data elements are application dependent and, if required, are specified by individual payment system specifications:

- Acquirer Identifier
- Application Identifier (AID)
- Application Version Number
- Certification Authority Public Key[7] (required if terminal supports data authentication)
  - Certification Authority Public Key Exponent
  - Certification Authority Public Key Modulus
- Certification Authority Public Key Index[7] (required if terminal supports data authentication): the key index in conjunction with the Registered Application Provider Identifier (RID) of the payment system Application Identifier (AID) identifies the key and the algorithm for data authentication
- Default Dynamic Data Authentication Data Object List (DDOL) (required if terminal supports dynamic data authentication)
- Default Transaction Certificate Data Object List (TDOL) (If not present, a default TDOL with no data objects in the list shall be assumed)

[7] See Part IV of *Integrated Circuit Card Specification for Payment Systems*

- Maximum Target Percentage to be used for Biased Random Selection (required if offline terminal with online capability)
- Merchant Category Code
- Merchant Identifier
- Merchant Name and Location
- PIN Pad Secret Key (required if the PIN pad and IC reader are not an integrated tamper-evident device or if the terminal supports enciphering PINs for online verification)
- Target Percentage to be used for Random Selection (required if offline terminal with online capability)
- Terminal Action Code - Default, Terminal Action Code - Denial, Terminal Action Code - Online (required if non-zero values to be used[8])
- Terminal Floor Limit (required if offline terminal or offline terminal with online capability)
- Terminal Identification
- Terminal Risk Management Data (if required by individual payment system rules)
- Threshold Value for Biased Random Selection (required if offline terminal with online capability)
- Transaction Currency Code
- Transaction Currency Exponent
- Transaction Reference Currency Code
- Transaction Reference Currency Conversion
- Transaction Reference Currency Exponent

The terminal shall provide the necessary logical key slots to handle the active and future replacement Certification Authority Public Keys necessary for data

[8] According to the *Integrated Circuit Card Application Specification for Payment Systems*, the default value consists of all bits set to '0', although 'Data authentication was not performed', 'Static data authentication failed', and 'Dynamic data authentication failed' bits are strongly recommended to be set to '1' in the Terminal Action Code - Default and Terminal Action Code - Online.

authentication. Each logical key slot shall contain the following data: RID, Certification Authority Public Key Index, Certification Authority Public Key.

When the Certification Authority Public Key is loaded to the terminal, the terminal shall verify the Certification Authority Public Key Check Sum to detect a key entry or transmission error. The method for calculating this check sum is by the terminal-supported Secure Hash Algorithm. If the verification process fails, the terminal shall not accept the Certification Authority Public Key and shall display an error message. After the Certification Authority Public Key is successfully loaded, the terminal should store the Certification Authority Public Key Check Sum.

A means for updating data elements specific to payment system applications shall be supported wherever this is not in conflict with national legal restrictions. Data update may be facilitated from a remote site over a network or locally.

THIS PAGE LEFT INTENTIONALLY BLANK

Part III

Cardholder, Attendant, and Acquirer Interface

1. Cardholder and Attendant Interface

1.1 Language Selection

The terminal shall support at least the local language which is the language of common usage in the terminal's locality or region. The messages displayed to the attendant shall always be in the local language. To display the standard messages defined in section 1.2, the terminal shall support the relevant character set defined in the corresponding part of ISO 8859.

Depending on the local environment and business conditions, the terminal should support multiple languages for displaying the set of messages described in section 1.2 to the cardholder. A terminal supporting multiple languages may need additional parts of ISO 8859 to display characters relevant to these languages.

ISO 8859 consists of several parts, each part specifying a set of up to 191 characters coded by means of a single 8-bit byte. Each part is intended for use for a group of languages. All parts of ISO 8859 contain a common set of 95 characters, coded between '20' (hexadecimal) and '7E' (hexadecimal) as shown in Annex C. This common character set allows the terminal to display Application Label(s) and messages in multiple languages using Latin characters without using diacritic marks (see example in Annex C).

A terminal supporting multiple languages shall compare the card's Language Preference with the languages supported in the terminal at the beginning of the transaction.

If a match is found, the language with the highest preference shall be used in the messages displayed to the cardholder. Language Preference is coded so that the language with the highest preference appears first and the lowest preference appears last.

If no match is found and the terminal supports more than one language, the terminal shall allow the cardholder to select the preferred language at the beginning of the transaction. The messages shall be displayed to the cardholder in the selected language.

If no match is found or the terminal supports only one language, the terminal shall display messages in that language.

When a message is displayed to the cardholder as well as the attendant, it should be displayed to the attendant in the local language and to the cardholder in the preferred language, if supported.

1.2 Standard Messages[9]

To ensure consistency in the messages displayed by the terminal and the PIN pad, the following set of messages (or their equivalent meaning) shall be used in the languages of preference for the cardholder and attendant.

The messages shall be uniquely identified by a two-character message identifier as shown below. The message identifier is for identification purposes only and is not to be displayed to the cardholder or attendant. Values '01' - '12' (hexadecimal) are described below. Values '13' - '3F' (hexadecimal) are reserved for assignment according to this specification. Values '40' - '7F' (hexadecimal) are reserved for use by the individual payment systems. Values '80' - 'BF' (hexadecimal) are reserved for use by acquirers. Values 'C0' - 'FF' (hexadecimal) are reserved for use by issuers.

There may be additional messages displayed for the attendant or cardholder.

Note: messages may be displayed simultaneously, such as 'Incorrect PIN' and 'Enter PIN'.

'01' - (AMOUNT)

Indicates the transaction amount to both the cardholder and attendant.

'02' - (AMOUNT) OK?

Invites a response from the cardholder indicating agreement or disagreement with the displayed transaction amount. Agreement or disagreement should be denoted by pressing the 'Enter' or 'Cancel' keys, respectively.

'03' - APPROVED

Indicates to the cardholder and attendant that the transaction has been approved.

'04' - CALL YOUR BANK

Indicates to the cardholder or attendant to contact the issuer or acquirer, as appropriate, such as for voice referrals.

'05' - CANCEL OR ENTER

When used with the 'Enter PIN' message, instructs the cardholder to validate PIN entry by pressing the 'Enter' key or to cancel PIN entry by pressing the 'Cancel' key.

[9] This specification does not imply that the terminal shall support a set of standard messages in English.

'06' - CARD ERROR

Indicates to the cardholder or attendant a malfunction of the card or a non-conformance to answer-to-reset.

'07' - DECLINED

Indicates to the cardholder and attendant that the online or offline authorisation has not been approved.

'08' - ENTER AMOUNT

Instructs the cardholder at an unattended terminal or the attendant at an attended terminal to enter the amount of the transaction. Confirmation or cancellation of amount entry should be denoted by pressing the 'Enter' or 'Cancel' keys, respectively.

'09' - ENTER PIN

Invites the cardholder to enter the PIN for the first and subsequent PIN tries. An asterisk is displayed for each digit of the PIN entered.

'0A' - INCORRECT PIN

Indicates that the PIN entered by the cardholder does not match the reference PIN.

'0B' - INSERT CARD

Instructs to insert the ICC into the IFD. Correct insertion should be noted by displaying the message 'Please Wait' to reassure the cardholder or attendant that the transaction is being processed.

'0C' - NOT ACCEPTED

Indicates to the cardholder and attendant that the application is not supported or there is a restriction on the use of the application, for example, the card has expired.

'0D' - PIN OK

Indicates that offline PIN verification was successful.

'0E' - PLEASE WAIT

Indicates to the cardholder and attendant that the transaction is being processed.

'0F' - PROCESSING ERROR

Displayed to the cardholder or attendant when the card is removed before the processing of a transaction is complete or when the transaction is aborted because of

a power failure, or the system or terminal has malfunctioned, such as communication errors or time-outs.

'10' - REMOVE CARD

Instructs to remove the ICC from the IFD.

'11' - USE CHIP READER

Instructs to insert ICC into the IC reader of the IFD.

'12' - USE MAG STRIPE

Instructs to insert ICC into the magnetic stripe reader of the terminal after IC reading fails, when the IC and magnetic stripe readers are not combined.

'13' - TRY AGAIN

Invites the cardholder to re-execute the last action performed.

1.3 Application Selection

A terminal supporting more than one application should offer the cardholder the ability to select an application or confirm the selection proposed by the terminal. Applications supported by both the ICC and the terminal shall be presented to the cardholder in priority sequence according to the card's Application Priority Indicator, if present, with the highest priority listed first.

A terminal allowing cardholder selection or confirmation shall read from the card's directory and display:

- The Application Preferred Name(s), if present and if the Issuer Code Table Index indicating the part of ISO 8859 to use is present and supported by the terminal (as indicated in Additional Terminal Capabilities).
- Otherwise, the Application Label(s), if present, by using the common character set of ISO 8859 (see Annex C).

A terminal not offering the cardholder the ability to select or confirm a selection shall determine those applications supported by both the card and the terminal that may be selected without confirmation of the cardholder according to Application Priority Indicator, if present. The terminal shall select the application with the highest priority from those.

If the card returns SW1 SW2 other than '9000' in response to the SELECT command indicating that the transaction cannot be performed with the selected application:

- A terminal allowing cardholder selection or confirmation should display the 'Try Again' message and shall present to the cardholder the list of applications supported by both the ICC and the terminal without this application

- A terminal not offering cardholder selection or confirmation shall select the application with the next highest priority among those supported by both the ICC and the terminal that may be selected without cardholder confirmation.

If no application can be selected, the terminal should display the 'Not Accepted' message and shall terminate the transaction.

The application used for the transaction shall be identified on the transaction receipt by the partial Application PAN (or the full PAN, if allowed by payment system rules) and the AID.

1.4 Receipt

Whenever a receipt is provided, it shall contain the AID in addition to the data required by payment system rules.[10] The AID shall be printed as hexadecimal characters.

[10] As stated in section 1.3, the receipt shall contain the partial Application PAN (or full if allowed).

2. Acquirer Interface

2.1 Message Content

Messages typically flow from the terminal to the acquirer and from the acquirer to the issuer. Message content may vary from one link to another, with data being added to enrich the message at the acquirer. To enrich the message, the acquirer stores static point of transaction data elements[11] based on the Merchant Identifier and/or the Terminal Identifier. These data elements are implicitly referred to by the Merchant/Terminal Identifier(s) and therefore may be absent in terminal to acquirer messages.[12] In the following sections, this implicit relationship is indicated by a specific condition: 'Present if the Merchant/Terminal Identifier(s) do not implicitly refer to the (data element)'.

Message content may also vary due to data requested by the acquirer but not the issuer, such as for transaction capture or audit. The ICC stored data elements are implicitly known by the issuer[13] based on the AID and/or PAN and therefore may be absent in acquirer to issuer messages. In the following sections, this implicit relationship is indicated by a specific condition: 'Present if requested by the acquirer'.

Data requirements may differ depending on terminal operational control, which is recognised through a specific condition: 'Present for Terminal Type = xx'. For example, Merchant Identifier is provided only for a merchant-controlled terminal (Terminal Type = '2x').

An authorisation message shall be used when transactions are batch data captured. A financial transaction message shall be used when online data capture is performed by the acquirer. An offline advice shall be conveyed within batch data capture when supported. An online advice or a reversal message shall be transmitted real-time, similarly to an authorisation or financial transaction message.

This section describes requirements associated with ICC transactions and distinguishes between new data elements and existing data elements used for magnetic stripe transactions. Data elements referred to as existing are those defined in ISO 8583:1987, though actual terminal message contents are usually specific to (each of) the acquiring system(s) to which the terminal is connected.

---

[11] These data elements indicate point of transaction acceptance characteristics that rarely change, such as Merchant Category Code, Acquirer Identifier, or Terminal Country Code.

[12] At a minimum, all data listed in the Card Risk Management Data Object Lists and the TDOL shall be available at the point of transaction.

[13] These data elements reflect card acceptance conditions and restrictions that rarely change, such as Application Interchange Profile, Application Usage Control, or Issuer Action Codes.