EMV®
Acquirer and Terminal Security Guidelines

EMVCo, LLC

Version 2.0
October 2023

Legal Notice

The EMV® Specifications are provided “AS IS” without warranties of any kind, and EMVCo neither assumes nor accepts any liability for any errors or omissions contained in these Specifications. EMVCO DISCLAIMS ALL REPRESENTATIONS AND WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AS TO THESE SPECIFICATIONS.

EMVCo makes no representations or warranties with respect to intellectual property rights of any third parties in or in relation to the Specifications. EMVCo undertakes no responsibility to determine whether any implementation of the EMV® Specifications may violate, infringe, or otherwise exercise the patent, copyright, trademark, trade secret, know-how, or other intellectual property rights of third parties, and thus any person who implements any part of the EMV® Specifications should consult an intellectual property attorney before any such implementation.

Without limiting the foregoing, the Specifications may provide for the use of public key encryption and other technology, which may be the subject matter of patents in several countries. Any party seeking to implement these Specifications is solely responsible for determining whether its activities require a license to any such technology, including for patents on public key encryption technology. EMVCo shall not be liable under any theory for any party’s infringement of any intellectual property rights in connection with the EMV® Specifications.

TABLE OF CONTENTS

1 Scope 4
2 References 5
3 Definitions 6
4 Abbreviations and Notations 8
5 EMV and Cryptography – Overview 9
5.1 Introduction 9
5.2 Payment System Model 9
5.3 Cryptographic Basics 11
5.4 EMV Card Authentication Methods 14
5.5 The Authorisation System 19
6 Security for EMV Card Acceptance 21
6.1 Terminal Security and Risk Policy 21
6.2 Terminal Lifecycle Security Considerations 22
6.2.1 Terminal Deployment 22
6.2.2 Certification Authority Public Key Storage 23
6.2.3 Certification Authority Public Key Withdrawal 24
6.2.4 Terminal Management 25
6.3 Processing of online authorisation requests. 27
6.4 Collation and transmission of transaction data 27
6.5 Certificate Revocation Lists 27
7 Terminal Security 28
7.1 Storage of Certification Authority Public Keys 28
7.2 Offline PIN Security 28
7.3 Online PIN Security 29
7.4 Random Number Generation 29
7.5 Terminal Risk Management 31
7.6 Support for CRLs 31
7.7 Type Approval and Other Certifications 31
7.8 Date and Time 31
7.9 Data Fidelity 32

1 Scope

These Security Guidelines are designed to assist acquirers of EMV payment cards with terminal security and acceptance processing. The acquirer is responsible for the terminal devices that accept payment cards, (including PIN pads when present) and the processing of the resulting transaction data. These responsibilities include the management of payment system public keys in terminals and the integrity of transaction data. In addition, acquirers are responsible for the protection of transaction data from unauthorised access whilst it is being processed and later when the data is stored.

The guidance presented in the following pages is applicable to deployment of payment terminals and the processing of transaction data for both authorisation and clearing. Where acquirers use third party agents to perform these functions, the same principles are also applicable.

Some aspects of these Guidelines may also be applicable to mPOS - a Mobile phone acting as a terminal, but the Guidelines are not intended to address security that is specific to mobile technology.

The materials contained in this document are intended primarily for acquirers and their agents. This document is not intended to supersede the requirements and specifications of any Payment System. The document is to be used as a “guideline”, assisting the acquirer, the acquirer's agent and other third parties regarding the secure acceptance of payment products conforming to the EMV specifications.

2 References

Throughout this document, the following references have been used. These references include the most current version at the time of preparation. For future use the most current versions of these documents should be used.

EMV Book 2	Integrated Circuit Card Specifications for Payment Systems. Book 2 Security and Key Management
EMV Book 4	Integrated Circuit Card Specifications for Payment Systems. Book 4 Cardholder, Attendant, and Acquirer Interface Requirements
EMV Book C-8	EMV Contactless Specification for Payment Systems, Book C-8 – Kernel 8 Specification
EMV Book E	EMV Contactless Specification for Payment Systems, Book E – Security and Key Management
PCI PTS Information Supplement	Skimming Prevention Best Practices for Merchants
PCI PTS POI Security Guidelines	PIN Transaction Security (PTS) Point of Interaction (POI) Modular Security Requirements
PCI PA-DSS	Payment Application Data Security Standard
ISO 9564-1	Financial Services – Personal Identification Number (PIN) management and security – Part 1: Basic principles and requirements for PINs in card-based systems.
ISO 11568	Financial Services – Key management (retail)
ISO 13491	Financial Services – Secure cryptographic devices (retail)
ISO 16609	Financial Services – Requirements for message authentication using symmetric techniques
ISO/IEC 18031	Information technology - Security techniques – Random bit generation
NIST SP800-22	A Statistical Test Suite for Random Number and Pseudo-random Number Generators for Cryptographic Applications
NIST SP800-90A	Recommendation for Random Number Generation Using Deterministic Random Bit Generators
ANS X9.82	Random Number Generation

3 Definitions

Authentication A cryptographic process that validates the origin and integrity of data.

Certificate (public key) The public key and the identity of an entity together with some other information, made unforgeable by the signing of the certificate with the private key of the certification authority issuing the certificate.

Certification Authority The entity that is trusted by one or more other entities to create and assign certificates.

Cryptographic Algorithm A set of rules, setting forth procedures necessary to authenticate or protect data, e.g. to perform encipherment and decipherment of data. The algorithm is specified in a manner that it is not possible to determine any of the secret control parameters, i.e., the secret or private key, except by exhaustive search.

Digital Signature The cryptographic transformation of data which provides:

- origin authentication.
- data integrity.

Hash Function A function, which maps values from a large domain into a smaller one. The function satisfies the following properties:

1. It is computationally infeasible to find for a given output, an input that maps to this output.
2. It is computationally infeasible to find for a given input, a second input that maps to the same output.

IC Card (ICC) A card with an embedded integrated circuit (chip) that communicates with a point of interaction (terminal).

Key Pair When used in public key cryptography, a public key and its corresponding private key.

Payment System A Payment System includes a number of participants where the issuer and the acquirer distribute responsibilities amongst the different parties according to Payment System rules and according to the allocation of risks.

Private Key In an asymmetric algorithm (public key) cryptosystem, the key of an entity's key pair that is known only to that entity. This is not the same as the secret key used in a symmetric algorithm.

Pseudo-random A process that produces numbers that are statistically random and essentially unpredictable although generated by an algorithmic process.

Public Key In an asymmetric key system, the key of an entity that is publicly known.

Secret Key

A key that is used in a symmetric cryptographic algorithm and cannot be disclosed publicly without compromising the security of the system. This is not the same as the *private* key in a public/private key pair.

Secure Cryptographic Device

A device that provides physically and logically protected cryptographic services and storage (e.g. PIN Entry device or Hardware Security Module), and which may be integrated into a larger system such as a point of sale device.

Terminal

The device used in conjunction with the chip card at the point of transaction to perform a financial transaction. The terminal incorporates the interface device and may also include other components and interfaces such as host communications.

4 Abbreviations and Notations

AC	Application Cryptogram
AIP	Application Interchange Profile
ARPC	Authorisation Response Cryptogram
ARQC	Authorisation Request Cryptogram
ATM	Automated Teller Machine
CA	Certification Authority
CDA	Combined DDA/Application Cryptogram Generation
CDCVM	Consumer Device Cardholder Verification Method
CRL	Certificate Revocation List
DDA	Dynamic Data Authentication
DES	Data Encryption Standard
ECC	Elliptic Curve Cryptography
EDA	Enhanced Data Authentication
IAD	Issuer Application Data
ICC	Integrated Circuit Card
MAC	Message Authentication Code
mPOS	Mobile Point of Sale
POI	Point of Interaction
POS	Point of Sale
RRP	Relay Resistance Protocol
RSA	Rivest, Shamir, Adleman algorithm
SDA	Static Data Authentication
TAC	Terminal Action Code
TC	Transaction Certificate
TDES	Triple DES (referred to as DES3 in EMV Book 2)
TVR	Terminal Verification Results
XDA	Extended Data Authentication

5 EMV and Cryptography – Overview

5.1 Introduction

The purpose of this overview is to provide a framework for the acquirer security guidelines. The EMV payment system model is described together with an outline of the roles of the entities within the model.

5.2 Payment System Model

The Payment System (as outlined in Figure 1) consists of the following types of entity:

- Cardholders,
- Merchants,
- Issuers,
- Acquirers, and
- Payment Systems (e.g. American Express, Discover, JCB, Mastercard, UnionPay and Visa).

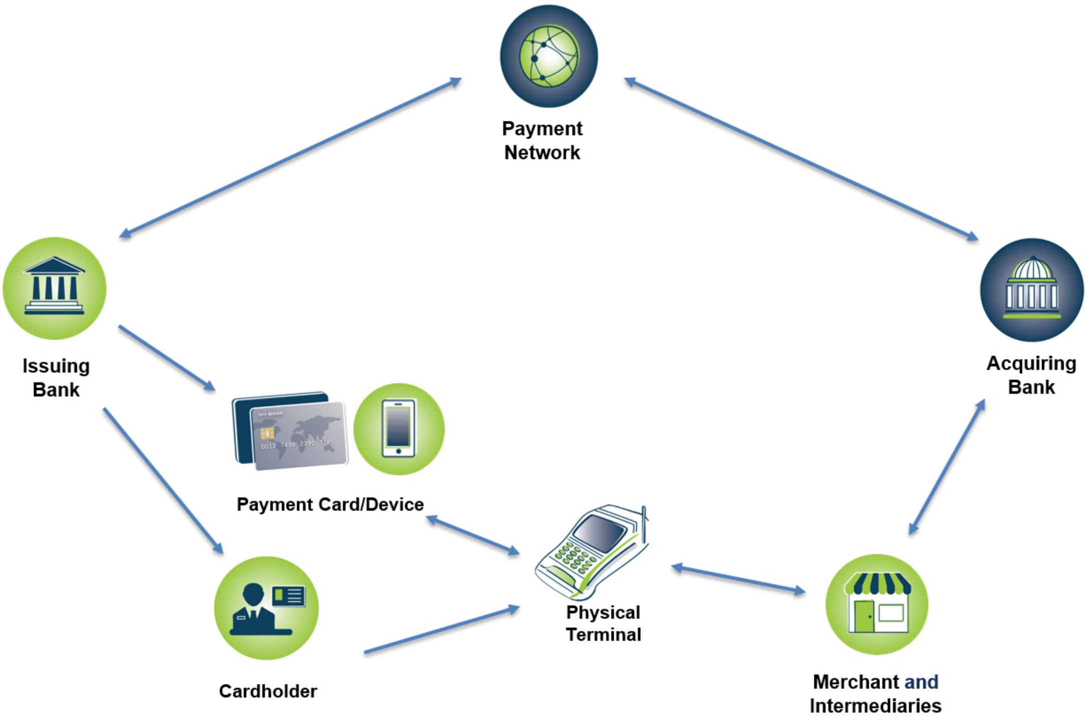


Figure 1 - System Model

The main role of each of these entities is as follows.

5.2.1 The Cardholder

The role of the cardholder includes the following:

- To obtain a chip card containing the payment product application by contracting with an issuer.
- To choose, remember and possibly update their PIN.
- To present their chip card to the merchant device accepting the payment product for payment (ATM, POS, vending machines, etc.).

5.2.2 The Merchant

The role of the merchant includes the following:

- To obtain payment terminals accepting chip cards by contracting with an acquirer.
- To accept chip cards containing the payment products for payment.
- To obtain reimbursement for the purchases by collecting and transmitting payment transaction details to the acquirer.

5.2.3 The Acquirer

The role of the acquirer includes the following:

- To contract with merchants and to deploy payment terminals. This includes the installation and management of Payment System public keys, adequately protected for integrity.
- To process payment transactions and to pay the merchant for them.
- To transmit the completed transaction records to the issuer in order to obtain the settlement.
- To manage the risk conditions relating to online/offline acceptance.

5.2.4 The Issuer

The role of the issuer includes the following:

- To contract with the cardholder, and to provision and issue a chip card containing the application to the cardholder. This includes the generation and installation of the necessary cryptographic keys in the card to support the application.
- To process online transactions. This includes verification of the data and cryptogram from the card together with data from the terminal, plus generation of a cryptogram allowing the card to authenticate the issuer. It also includes verification of online cardholder PINs as part of standard authorisation processing.
- To generate update scripts to the card application when appropriate.

- To process clearing messages including verification of the data and associated Transaction Certificate, when appropriate. In some circumstances this could be deferred and only checked in the case of dispute.
- To reimburse the acquirer for payment transactions.
- To securely transmit to any other parties the necessary cryptographic keys needed for the correct operation of the system.

5.2.5 The Payment System

The role of the Payment System includes the following:

- To specify the system rules for the products and services and to verify compliance with them.
- To generate and distribute Payment System public keys.
- To certify Issuer Public Keys used within the system.
- To operate on-line communication networks between acquirers and issuers.
- To perform clearing and settlement for transactions on this network.

5.3 Cryptographic Basics

Historically, cryptography has been used to provide data confidentiality and today includes additional cryptographic functions such as data integrity, authentication, and non-repudiation. International standards have been developed to facilitate interoperability of products and services between different vendors and various cryptographic implementations. The materials contained in these guidelines represent the best practices drawn from these different standards.

Modern cryptography depends on two basic components: (1) the algorithm, and (2) the cryptographic key, with overall security dependant on public access to the algorithm and secure management of the secret and private keys over the key lifecycle. The algorithms define how ciphertext is obtained from plaintext and vice versa and how data is signed and verified. Algorithms are typically published and have been extensively studied by cryptographers – it is the use of unique keys for every user that ensures that unauthorised parties are unable to decrypt sensitive data or forge another parties' digital signature.

There are two basic types of cryptographic algorithms: (1) symmetric or secret key algorithms, and (2) asymmetric or public/private key algorithms and both are used within the context of EMV.

5.3.1 Symmetric Algorithms

Symmetric or ‘secret key’ algorithms require that the secret key used for the encryption process also be used in the decryption process. Therefore, the security of the encryption process depends entirely on protection of this secret key. Issuer host systems are protected against physical compromise of master keys and card keys and chip cards are protected against side channel attacks that might reveal a card’s unique key. Exhaustive key search attacks are currently computationally infeasible for 2-key Triple DES.

5.3.1.1 Hash and Keyed Hash Functions

Hash functions take an input data string of arbitrary length and convert it to an output string of a fixed length. With a sufficiently large output length it is computationally impossible to find two input data strings that map to the same hash output, known as a collision. Typically, for data sent between two parties, the data in a message is hashed and if the receiver computes the same hash value from the received message data, it indicates that the data has not been corrupted.

A keyed hash function incorporates a symmetric secret key, known only to both parties, into the hash computation. Consequently, if the hash value computed by the receiver matches that from the sender, then the receiver has validated both the integrity of the message data and the authenticity of the sender.

5.3.2 Asymmetric Algorithms

Asymmetric or ‘public key’ algorithms are generally based on a “hard” mathematical problem and have a design goal that there should be no better way to attack the scheme other than solving the hard problem. RSA is based on the hard problem of factorisation; that is, for a number consisting of two prime numbers multiplied together, find the primes given only the product, known as the modulus. ECC is based on the hard problem of solving the discrete log problem in a group of points of an elliptic curve.

Asymmetric algorithms require the communicating endpoints to use two different, but linked, keys: a “public” key and a “private” key. The RSA and ECC asymmetric algorithms are specified by EMV to create digital signatures for offline data authentication and for offline PIN encipherment. In a digital signature scheme the private key (sometimes referred to as the signature key) is used to generate the signature and the public key (sometimes referred to as the verification key) is used to verify the signature. For offline PIN encipherment, the public key is used for encipherment and the private key is used for decipherment.

5.3.2.1 Asymmetric (RSA & ECC) Keys

The security of the private (signature) keys used with the RSA algorithm depends on:

- The length of the RSA key modulus, e.g. 1024, 1152, 1408, and 1984 bit keys.
- The quality of the prime numbers making up the public/private key modulus.

The security of the private (signature) keys used with the ECC algorithm depends on a number of factors including:

- The choice of a reputable curve and group generator, e.g. NIST curve P-256.
- The quality of the random number generator creating the private key.

Potential risks to the private (signature) key include:

- The physical security of the private (signature) key from unauthorised access and exposure/compromise whilst in storage, in transit or in use.
- Side channel attacks on keys held in chip cards.
- Collapse of the underlying algorithm – most unlikely after decades of cryptanalysis.

The EMVCo Security Working Group conducts an annual review of Payment System cryptography, including RSA key lengths, based on independent analyses by the participating Payment Systems. Using the recommendations from the review, the Payment Systems may update their Payment System key lifetimes.

The lifetime for the longest RSA key (1984-bit) is currently considered to be an ‘anticipated lifetime’ in order to reflect the situation that when looking more than ten years into the future, the variation in prediction becomes too large for a reliable date to be given. Over time this date is also expected to move out, until a lifetime of ten years or less is predicted at which time the date will be considered as an expiry date.

5.3.2.2 Certificates and Certification Authorities

EMV follows the usual practice of certificates to validate the source of issuer and card public keys.

A certificate is a form of digital signature designed to validate the origin and integrity of a public key. A certificate consists of a public key concatenated with other related data and signed with the private key of a trusted entity known as a Certification Authority (CA). Any entity with a trusted copy of the CA’s Public Key can then verify all certificates generated by that CA and thereby obtain trusted copies of other users’ public keys.

In the EMV environment, the Payment System acts as a Certification Authority and creates Issuer Public Key certificates by signing each issuer public key. Issuers act as Certification Authorities and create ICC Public Key certificates by signing each ICC Public Key. The Payment Systems CA’s Public Keys are distributed to the terminals through the acquirers for verifying issuer certificates, thereby yielding trusted copies of issuers’ public keys, used in turn to verify ICC Public Keys.

5.4 EMV Card Authentication Methods

EMV supports offline and online methods for authenticating that a card is genuine and that the data on the card has not been altered since it was provisioned by the issuer. The offline methods require a Public Key Infrastructure (PKI) rooted to a Certification Authority under control of the Payment System as described in 5.3.2.2.

5.4.1 Offline Data Authentication

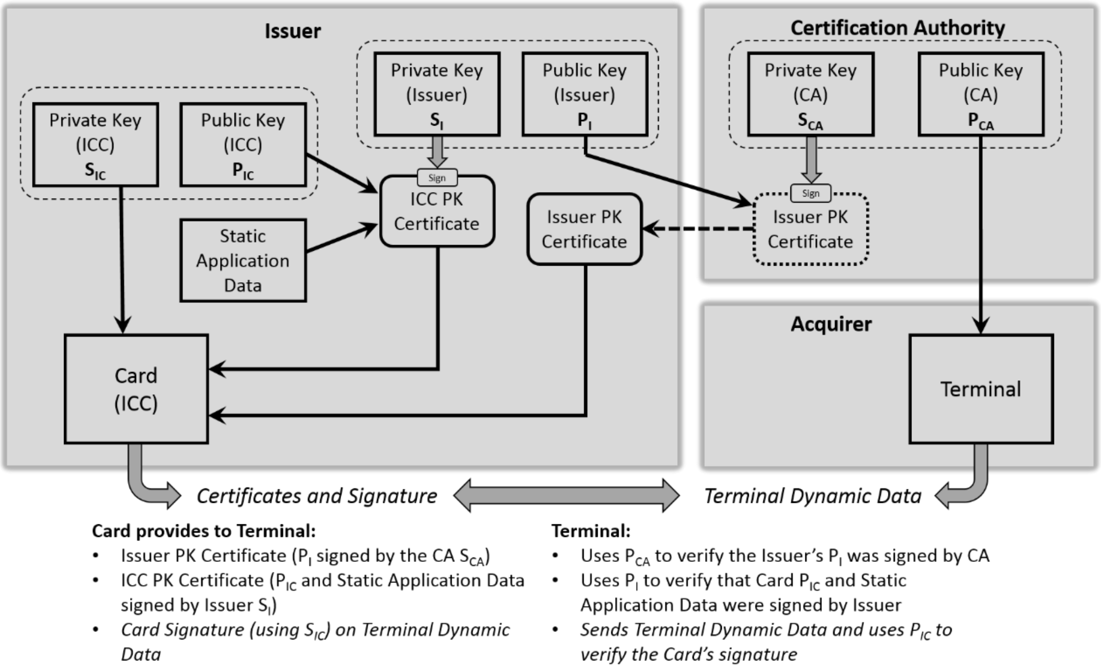
For EMV chip cards, the two RSA based methods of offline data authentication are Dynamic Data Authentication (DDA) and Combined DDA/Application Cryptogram Generation (CDA). There is only one ECC based method of offline data authentication, known as Extended Data Authentication (XDA).

- With DDA the terminal verifies a dynamic signature (i.e. different for each transaction) generated by the card using its private key, in order to ensure that the card is not counterfeit and that the card data has not been altered.
- With CDA and XDA the card generates a dynamic signature of transaction data including the online cryptogram, in order to provide the protection of DDA while also ensuring that an intermediate (wedge) device has not altered important data going between the card and terminal.

The specifications also include a version of RSA based offline data authentication called SDA (Static Data Authentication) which is a simple signature over static card data and was included as historically early cards did not have public key cryptographic capabilities. However, it is not proof against cloning, meaning that the relevant data can be copied from a legitimate card and applied to a counterfeit product (blank stock). Such a cloned card will work in offline environments, but will fail if the transaction is sent online.

Given the cryptographic capabilities of modern card products, SDA is no longer recommended and in addition Issuer Action Codes should be set to ensure that any SDA card product that appears in the field will have the transaction sent online, or be declined in an offline only environment.

The relationship between the data and the cryptographic keys is shown in Figure 2. For further information, please refer to EMV Book 2.



Card provides to Terminal:

- Issuer PK Certificate (P_I signed by the CA S_{CA})
- ICC PK Certificate (P_{IC} and Static Application Data signed by Issuer S_I)
- *Card Signature (using S_{IC}) on Terminal Dynamic Data*

Terminal:

- Uses P_{CA} to verify the Issuer's P_I was signed by CA
- Uses P_I to verify that Card P_{IC} and Static Application Data were signed by Issuer
- *Sends Terminal Dynamic Data and uses P_{IC} to verify the Card's signature*

Figure 2 – DDA / CDA / XDA

5.4.2 Kernel 8 Local Authentication

For Kernel 8, local authentication is achieved by the reader comparing a MAC computed over selected transaction data by the card, to what should be the same MAC value computed independently by the reader over the same transaction data.

The symmetric keys used by each party for the MACs are obtained from a Blinded Diffie-Hellman exchange between the card and the reader. The reader public key pair is an ephemeral value and the card public key is rendered ephemeral by means of a random blinding factor.

To complete the local authentication, the reader compares the MAC values, confirms the blinding factor and authenticates the card public key by means of the certificate chain from the issuer and payment system (as for EMV chip cards).

The MAC calculation is a two-step process. First the card computes an Issuer Application Data MAC (IAD-MAC) over card data and selected transaction data. This is internal and does not leave the card. The card also calculates the Application Cryptogram (AC - in a similar way as for EMV chip cards) and this is then included with IAD-MAC in a second MAC computation to form the Enhanced Data Authentication MAC (EDA-MAC). The reader duplicates this process independently, first calculating an IAD-MAC from the card data available from the transaction and then MACing this along with the AC value output by the card.

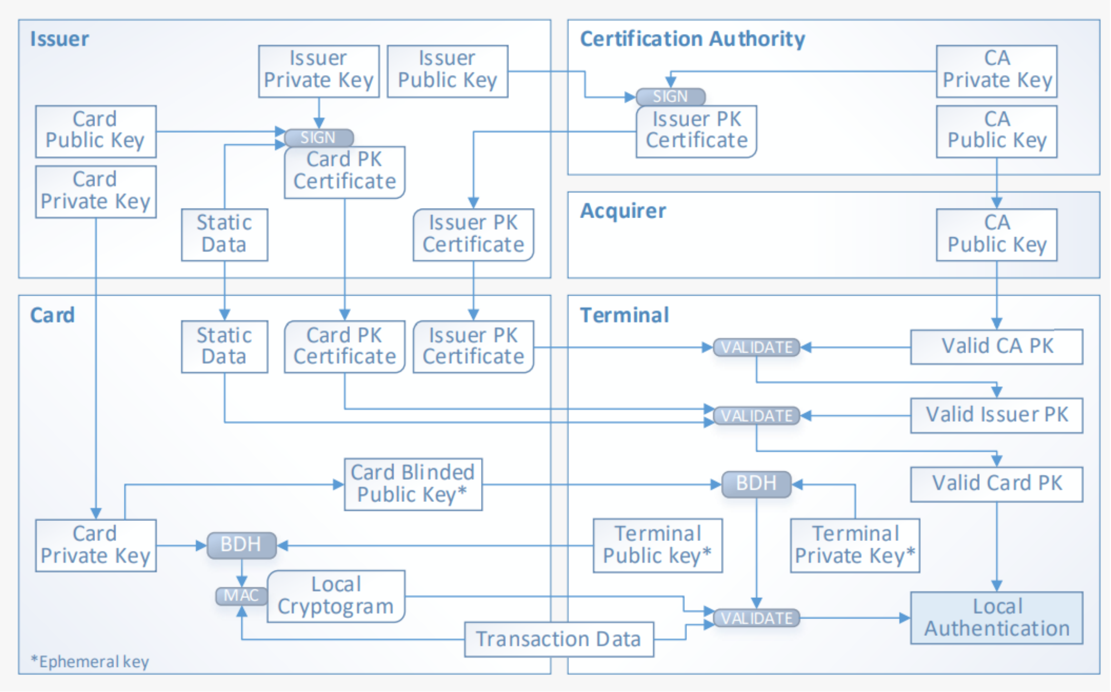


Figure 3 – Local Authentication

5.4.3 Online Data Authentication

For EMV chip cards, EMV's online authentication methods are used to validate the card to the issuer and the issuer to the card as well as to prove the authenticity of received data.

- With Online Card Authentication the issuer online system validates a cryptogram (an Application Cryptogram called an ARQC) generated by the card from important transaction data using its unique secret key, to show that the card is not counterfeit and that the data has not been altered.
- With Online Issuer Authentication, the card validates an issuer-generated cryptogram and then performs internal card management functions, such as the reset of offline counters.
- With secure messaging the issuer sends a script update to the card protected by a MAC. The card only applies the updates if the MAC is valid. Secure messaging is also used to encipher confidential data, such as a replacement PIN value during transport between the issuer and the card.
- For approved transactions the terminal sends a cryptogram (an Application Cryptogram called a TC) generated by the card with the clearing information for verification by the issuer as evidence of the validity of the completed transaction.

5.4.4 Kernel 8 Remote Authentication

For Kernel 8, remote authentication is achieved in the same way as for Online Authentication, by the Issuer re-calculating the ARQC from data supplied in the authorisation request message and comparing it to the value received from the card. The input data for ARQC calculation may include the IAD-MAC value calculated by the reader and included in the authorisation request message. If the IAD-MAC calculated internally by the card is included in its ARQC generation and the ARQC validates at the Issuer, the Issuer knows that the card is not counterfeit, that the transaction data has not been altered and that both the card and the reader had the same view of the local data, which was not modified.

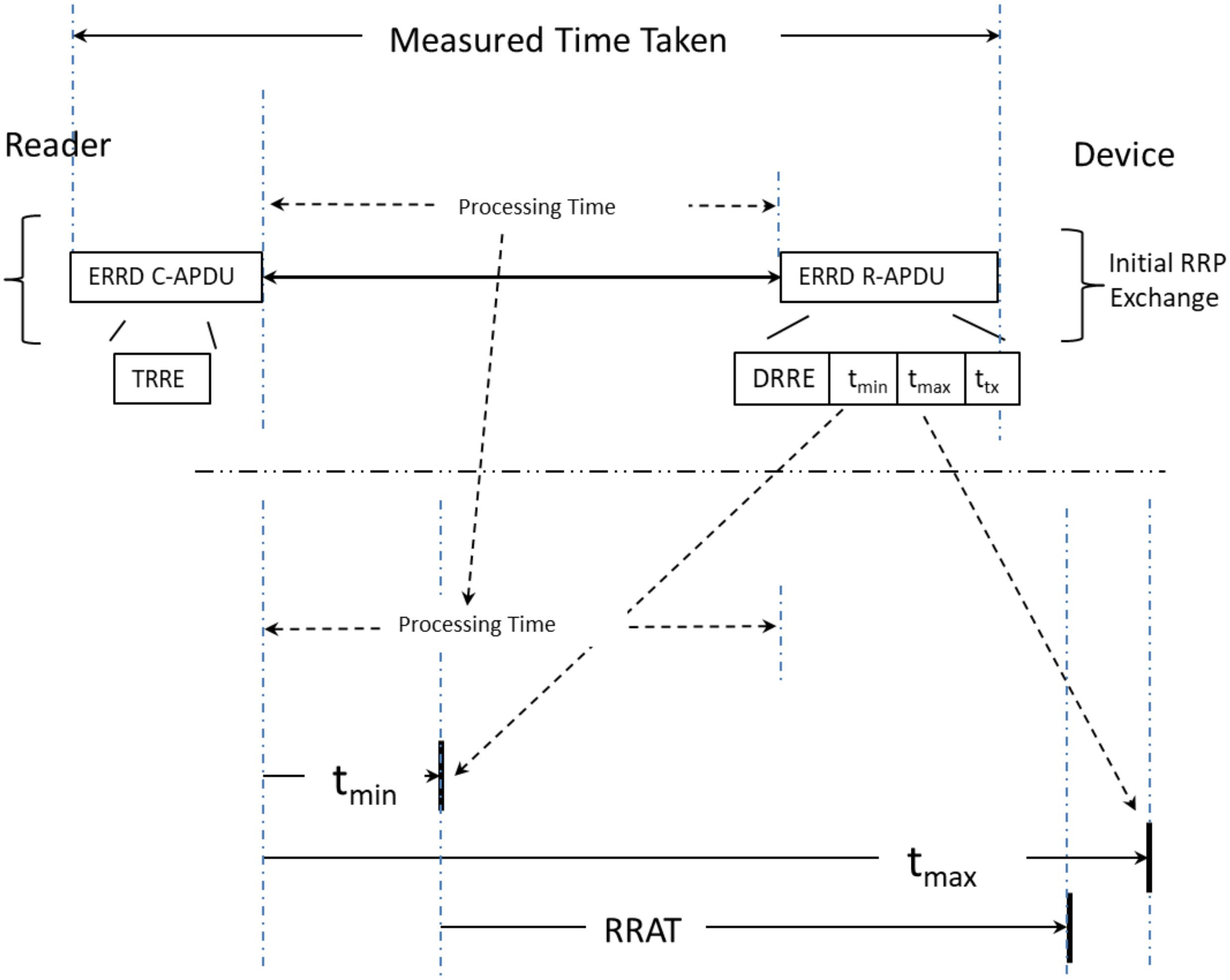
5.4.5 Relay Resistance Protocol

The C-8 specification includes a Relay Resistance Protocol (RRP). At an early stage in the transaction the reader sends a random value (TRRE – Terminal Relay Resistance Entropy) to the card, which immediately responds with its own random value (DRRE – Device Relay Resistance Entropy) and the reader measures the time of the exchange.

In its response, the card also indicates the minimum and maximum processing times that a response should usually take. If the processing time calculated from the reader's measured response time does not lie between the max and min times, then the reader should consider that a relay attack has been detected.

The reader also has an accuracy threshold time value from the acquirer, RRAT (Relay Resistance Accuracy Threshold), which is used in addition to the max time value.

Later in the transaction, TRRE, DRRE, t_{min}, t_{max} and t_{tx} are all included in the card and reader IAD-MAC calculations. This confirms to the reader, via the EDA-MAC, that the values used in the relay attack detection were not manipulated and similarly the issuer knows the same via the ARQC check.



ERRD = Exchange Relay Resistance Data
DRRE = Device Relay Resistance Entropy (random value)
TRRE = Terminal Relay Resistance Entropy (random value)
RRAT = Relay Resistance Accuracy Threshold

Figure 4 – RRP Timing

Processing Time is calculated as Measured Time less transmit time for ERRD C-APDU and transmit time for ERRC R-APDU (given by t_{tx}).

If the Processing Time does not lie between t_{min} and t_{max} then a relay attack is detected.

If t_{max} is exceeded, the corresponding bit is set in the TVR.

The terminal has an acquirer value RRAT, which is used alongside t_{max} (maybe shorter or longer), which if exceeded also results in the corresponding bit being set in the TVR.

Later in the transaction, ERRD, DRRE, t_{min} and t_{max} are all included in the EDA MAC. This confirms that the values used in the relay attack detection were not manipulated.

5.4.6 Cardholder Verification Methods

EMV supports the following methods for verifying a legitimate cardholder:

- Online PIN, where a cardholder-entered value is enciphered by the terminal/PIN pad and sent with an online authorisation request to the issuer for validation.
- Offline PIN, where a cardholder-entered value is sent to the card and the ICC compares this value to a reference PIN stored securely on the card. The terminal is informed of the success or failure of the verification.
- On device verification for devices such as mobile phones and tablets - Consumer Device Cardholder Verification Method (CDCVM).
- Signature.

EMV provides for two types of offline PIN verification:

- Offline Plaintext PIN, where the cardholder-entered value sent to the ICC is unencrypted.
- Offline Enciphered PIN, where the cardholder-entered value is enciphered (RSA or ECC) before being sent and deciphered by the ICC.

To prevent PIN probing attacks, dual interface (and contactless-only) cards should not respond to the VERFIY command over the contactless interface.

Offline PIN encryption may be supported using either the DDA/CDA keys or dedicated keys. For security reasons the use of dedicated keys is a best practice, however dedicated keys may impact key management and transaction performance.

All of the PIN methods involve a precise interaction with the cardholder, the result of which is either right or wrong, whereas signature requires a human comparison that is subjective.

5.5 The Authorisation System

Authorisation is a process whereby an issuer or a representative of the issuer approves or declines a transaction in response to an online authorisation request from a merchant via an acquirer.

The online authorisation request includes a card generated authorisation cryptogram (ARQC) which the issuer validates to ensure that the card is authentic and the transaction data is unaltered. The online request also includes card and terminal indicators of the results of offline processing.

In response to the ARQC, the issuer optionally creates an authorisation response cryptogram (ARPC). The card validates the ARPC to assure that the authorisation response came unaltered from the issuer.

In addition to the ARPC described above, issuers can perform post-issuance updates of cards using issuer script commands. For example, the issuer can change the Offline PIN or update a card's risk parameters. The issuer protects these script commands from undetected alteration by generating a cryptogram (MAC) from the command data. The card validates the MAC before applying the changes. Confidential data is enciphered.

6 Security for EMV Card Acceptance

This section addresses the security related functions that need to be performed by an EMV acquirer.

- Establishment of a terminal risk policy and security requirements.
- Follow the Lifecycle Security Requirements as described in PCI PTS POI, including
    - Deployment of terminals to contracted merchants in accordance with the above policy and requirements.
    - Distribution of payment system public keys into the terminal base.
    - Key removal.
    - Terminal management.
- Processing of online authorisation requests, including the transmission of all cryptograms and related data and scripts.
- Collation of transaction data and transmission for clearing and settlement.
- Management of the risk conditions relating to online/offline acceptance.
- Optional support of the Certificate Revocation List (CRL) process.
- Securing the confidentiality of online PINs (if supported).

6.1 Terminal Security and Risk Policy

Before considering the deployment of terminals it is important to know what security requirements they should meet. As for any other area of IT Security, this would normally be defined in a Security Policy. This Policy should consider the risks associated with the deployment of the terminals, the threats that they are exposed to and the policies needed for their secure operation. This permits the acquirer to discuss clearly stated security requirements with terminal vendors, place requirements on the development of applications, determine requirements for terminal management systems and create operational procedures and controls for their operation. It also creates a clear business agreement with merchants for their respective security responsibilities.

Without such a Policy many of the recommendations that follow in this document would be hard to implement or would be hard to monitor.

Whilst EMV creates the basic framework for secure transactions, it does not mandate specific methods for the link between acquirer and terminal; hence for example it states a number of basic security principles and policies in regard of key management in Section 10 of Book 2, but it does not state specific methods by which these principles are to be met.

[6.0] *Acquirers should formulate their security requirements for the deployment of terminals in a Security Policy. This Policy would form part of the terminal procurement requirements, deployment procedures and management processes for the terminal base. The Policy should, where appropriate, reference supporting material such as Payment System requirements, PCI Skimming Prevention Best Practices for Merchants, and acquirer corporate security policies.*

Acquirers should establish an offline/online risk policy according to payment system rules. Typically, this will involve establishing merchant floor limits and the setting of Terminal Action Codes (TACs). These settings should be under the control of the acquirer, not the merchant.

Typical examples of TAC settings (from payment system rules) might be:

The TAC - Online is set to generate an online authorisation when:

- *Offline data authentication is not performed*
- *The card only supports SDA*
- *DDA or CDA failed*
- *Online PIN is entered*
- *Transaction exceeds the floor limit.*

The TAC—Default contains a value that generates a decline if the transaction cannot be sent online for authorization when:

- *Offline data authentication is not performed*
- *The card only supports SDA*
- *DDA, or CDA failed*
- *Transaction exceeds the floor limit*

6.2 Terminal Lifecycle Security Considerations

The PCI PTS POI Evaluation Module 4 lists the Terminal Lifecycle Security Requirements.

6.2.1 Terminal Deployment

The deployment of terminals and the related contractual relationships between merchants and acquiring banks are varied and diverse. At one end of the spectrum an acquirer may own a terminal and provide it to a single outlet merchant, with a simple point-to-point telecommunications link. At the other end an acquirer may process transactions from a large chain of merchant outlets, widely dispersed across the acquiring territory, with each store owning many terminals integrated into electronic till systems, connected to complex merchant back-office processing and a distributed communication network. Irrespective of ownership, acquirers should maintain an inventory of all the terminals from which they process transactions and should be able to identify each terminal uniquely and know where it is located and which software version it is running.

Irrespective of physical nature and distribution of terminals that accept EMV payment products, acquirers have an obligation to ensure that:

- All terminals and kernels have been approved under the EMVCo Terminal Type Approval process.
- All terminals have a PCI PTS POI approval.
- All PIN pads, either separate or integrated into terminals, have a PCI PTS POI approval. This applies for both PIN pads used for offline PIN checking and online encrypting PIN pads.

- Installed terminals have passed Payment System L3 terminal integration testing.

Depending on the relationship between an acquirer and its merchants some of these functions may be devolved to the merchant (e.g. supermarket) or third party processors.

[6.1] *Acquirers should only process transactions from terminals that meet their security policy and requirements and that have valid Type Approval certifications. They should check with vendors on their renewal policy, or alternative models, if a particular model only has a short period left before the certification expires.*

[6.2] *Acquirers should monitor the certification programme for their deployed terminal base and encourage terminal management programmes that update devices in the field.*

[6.3] *Acquirers should only deploy terminals and PIN pads with valid PCI PTS POI modular approvals and should check with vendors on their renewal policy, or alternative models, if a particular model only has a short period left before the certification expires. Acquirers should refer to the PCI document "Skimming Prevention Best Practices for Merchants" and apply its advice.*

[6.4] *For terminals in exposed environments and especially those with a high level of staff turnover, such as garages and fast-food outlets, acquirers should encourage merchants to physically secure the terminals, using a lock under control of the site management. Information can be found in the PCI document "Skimming Prevention Best Practices for Merchants".*

[6.5] *Acquirers should establish a terminal management policy with merchants, such that terminal replacement and maintenance procedures are clearly defined, making it more likely that merchant staff will be suspicious of bogus terminal service technicians.*

[6.6] *Acquirers should maintain an inventory of all the terminals from which they process transactions and should be able to identify each terminal uniquely, know where it is located and which software version it is running*.

6.2.2 Certification Authority Public Key Storage

Every EMV terminal supporting any of the EMV offline cryptographic functions needs to have the Certification Authority public keys of all the EMVCo payment systems that the terminal is branded to accept. The values of the public keys should be obtained from the individual Payment Systems.

These keys support RSA-based SDA / DDA / CDA, ECC-based XDA / EDA and/or encryption of offline PIN and/or biometric data, and need to be present before any cards with certificates based on these keys enter circulation. Acquirers are responsible for ensuring that all terminals under their jurisdiction have the necessary public keys and that they have been accurately installed and are protected against deliberate or accidental modification, or at least, any changes to the key set will be immediately and noticeably detected. At the time of this version of the guidelines, the EMVCo payment systems may each need support for a set of RSA public keys that include keys of 1408 bits and 1984 bits in length and ECC public keys for curves P-256 and P-521. Acquirers are encouraged to maintain a log of the key status of all terminal devices under their jurisdiction and to make this information available upon reasonable request from a payment system.

The 1984-bit RSA keys represent the maximum size that can be accommodated within the EMV structure and thus for the members of EMVCo, all expected RSA keys have been published. If all keys (RSA and ECC) are installed at the time of terminal deployment, it is expected that no further key installation will be required, which reduces the logistics of introducing new keys, although acquirers do need to retain the ability to do so. Acquirers whose terminal base does not include all the required keys, should add them.

Note that for a distributed system, such as EMV terminals attached to multiple electronic tills, the keys may be held centrally and distributed as required or a common processor may perform the offline certificate and signature verification instead of the terminal. Regardless, the acquirer obligations remain the same.

Acquirers are also responsible for ensuring that terminals are not populated with spurious keys, such as test keys, previously expired keys or keys for acceptance of card products not within the acquirer's business portfolio.

[6.7] *Acquirers obtaining payment system public keys for loading to terminals should check the validity of the key(s) with a second source and should have integrity checks to ensure that keys cannot be changed whilst under their jurisdiction, as described in PCI PTS POI*

[6.8] *Acquirers should ensure that the terminals under their jurisdiction detect and report unexpected modification of the key set. This should feed into a management process to alert the acquirer. Acquirers should follow-up on these reports.*

6.2.3 Certification Authority Public Key Withdrawal

Acquirers need to remove the Certification Authority Public Keys from service in their terminals within a specific grace period after expiration. This is expected to be six months starting from the planned expiration date (until June 30th of the following calendar year) but may be deferred at payment system discretion.

It may be noted that much of this material relates to the introduction of longer RSA keys, necessary as the shorter keys lose security strength. This sequence will end when the 1984-bit keys reach an expiration date. Whilst ECC keys are expected to be stable in the long term, the ability to install and withdraw keys is still necessary to allow for infrastructure changes, such as the emergence of a new domestic payment system.

Acquirers do not necessarily need to load the payment system public keys themselves. They may be available preloaded by terminal vendors or they may utilise a terminal management programme available from some vendors. In the latter case acquirers should look for the following services with respect to keys:

- Download of public keys to devices prior to and after deployment using techniques to protect integrity.
- Removal of expired or revoked keys with checks that the action is legitimate.
- Service log indication of the key status of each terminal device covered, with full summary information for the terminal base.

[6.9] *Acquirers should establish a relationship with merchants and/or vendor terminal management programmes, to ensure that expired keys are removed within the established 6-month period. They should maintain a log and on request should be able to confirm progress, including confirmation that a key has been fully withdrawn from their terminal base.*

[6.10] *If a Certification Authority Public Key is revoked, then acquirers should be prepared to respond and complete the withdrawal within a 6-month period. They should maintain a log and on request should be able to confirm progress (on a per merchant basis), including confirmation that a key has been fully withdrawn from their terminal base.*

6.2.4 Terminal Management

Once deployed, acquirers are responsible for managing their terminal base. The main items to be addressed include:

- Update of terminal configuration data - including offline floor limits, TACs, public keys and CRLs. Note that data unlikely to change during the deployed lifetime of a terminal, (such as currency code, country code, terminal capabilities) may also be set at the time of deployment.
- Terminal code updates:
    - For fixing of bugs after deployment. Whilst Type Approval testing should confirm that a terminal conforms to the specifications, there always exists the possibility that a particular terminal model has implementation or interoperability issues. These could require a global fix, or there may be specific regional or local requirements.
    - As a result of specification updates. From time to time it may be necessary for the EMV specifications to be updated and for the changes to be introduced faster than waiting for the terminal replacement cycle.

    In both cases, terminals should only be updated to an approved configuration.
- Device monitoring – reporting on software versions, public key presence and operational statistics, such as the number of transactions, offline authentication failures, declines and fall-back.

How acquirers can achieve this practically will depend on a number of factors relating to the acceptance environment and merchant practices. Typically, there is a choice between:

- Remote management – parameters and code can be updated by means of a client/server based secure terminal management system. This may be provided as part of an ongoing contract by the terminal vendor and should be able to manage and update all configuration data and track and update software versions between all supported terminal models.
- Local management – technicians update terminals locally in the merchant environment. This may be via a variety of methods, including menu driven parameter updates, use of management cards carrying the necessary configuration information and replacement of memory modules, including PROMs.
- Device swaps – terminals are physically replaced by technicians, or by the merchant who may receive a replacement terminal and return the original.

Large stores with distributed systems may manage terminals as part of their daily activities and could dynamically change configuration data, such as floor limits according to the time of day and customer numbers. They may also be constrained by operational and security requirements from supporting client/server based remote management, for example due to firewalls in support of PCI DSS requirements.

Acquirers should ensure that merchant practices and training prevent the unauthorised updating or replacement of terminals. Remote management systems should employ cryptographic authentication services and merchants should have clear procedures when updating their own devices or checking the credentials of a visiting technician.

PCI SSC DSS and PA-DSS security standards contain further information on these issues.

Acquirers should consider the following features when selecting a remote terminal management system:

- Validation that all mandatory functions for a device type are active and cannot be deleted.
- Addition or deletion of optional functions, provided that the final configuration loaded into the device has been EMV approved.
- Tracking of TAC settings with update when appropriate. Merchants must not be able to update TACs arbitrarily.
- Key removals and introductions. If the communications channel is not under the control of an acquirer, then authentication is required or validation of the request against an alternate channel.

[6.11] *Acquirers should establish a terminal management programme for all terminal devices within their terminal base. By use of this programme acquirers should be able to:*

- *Withdraw payment system public keys.*
- *Validate the inventory of installed payment system keys on each device.*
- *Update acquiring risk management parameters.*
- *Provide for terminal software updates.*
- *Introduce new payment system public keys. Whilst this may now be largely redundant for keys from the members of EMVCo, the ability to introduce new keys should remain.*

[6.12] *Acquirers and merchants should ensure that any key withdrawal or introduction notification is from the legitimate payment system and is uncorrupted.*

[6.13] *Acquirers should ensure that the terminal management procedures are sufficiently secure in order to prevent terminals being updated or replaced by unauthorised parties.*

6.3 Processing of online authorisation requests.

Online authorisation requests pass via the acquirer on their way to a payment system for routing to the appropriate issuer and responses return via the acquirer on their return. Acquirers are responsible for:

- Ensuring that the transaction data accompanying the authorisation request is accurate and is not corrupted.
- Routing the transaction to the appropriate payment system.
- Any editing and plugging of data values that may be necessary due to specific terminal configurations and installations. Please note: Acquirer and merchant systems must not alter chip data from the card or terminal, especially the data included in the list shown in EMV Book 2 Table 28.
- Ensuring that the issuer response is returned to the terminal and that the data is not corrupted. This includes forwarding of any script messages that may accompany the issuer response.

6.4 Collation and transmission of transaction data

Acquirers are responsible for collating transaction information to be forwarded for clearing and settlement. Typically, the information will be batched according to payment system rules and dispatched on a regular basis.

In some instances acquirers will accumulate clearing transactions in real time and in others they may poll terminals on typically a daily basis.

The main security requirement on EMV transaction data is on its availability and integrity. Data should be protected against unauthorised alteration or deletion. The only confidentiality requirement is on PIN data, or any keys needed for terminal integrity.

PCI PTS has requirements on data confidentiality and integrity. This is outside the scope of EMV.

6.5 Certificate Revocation Lists

If an issuer private key were to be compromised, fraudulent cards that pass offline data authentication could be created using this key. These cards would be accepted at terminals for offline transactions until the certificate for the compromised key expires, which for some keys could be many years into the future.

The impact of the issuer key compromise can be mitigated through the use of Certificate Revocations Lists (CRLs). When processing the issuer certificate for offline data authentication, the terminal checks whether the certificate read from the card is listed amongst the CRL entries and if so, offline data authentication will fail.

[6.14] *Acquirers should establish a process to manage CRLs amongst the terminal base under their jurisdiction. It may be that this is most conveniently accomplished through a relationship with a vendor terminal management programme.*

7 Terminal Security

Although EMV terminals do not contain secret/private keys or other data that must be protected from a cryptographic perspective, they nevertheless perform an important role in the overall security of the EMV payments process. In particular, the correct functioning of the device and the immunity it has against manipulation that might change its behaviour or configuration are crucial for preventing attacks. Terminal vendors should instigate a design and development process that embeds defensive concepts throughout the process and acquirers should satisfy themselves that due care and attention has been paid to such matters. This is closely associated with the ability of acquirers to deploy and then manage and update their terminal base in a secure manner without introducing implementation issues.

Further information can be found in PCI PTS POI Security Requirements and PCI Software Security Terminal Application Security Guidelines.

7.1 Storage of Certification Authority Public Keys

Terminals are required to store multiple certification authority public keys for each payment system whose card products they are contracted to accept – up to six RSA keys and ten ECC keys. The modulus of an RSA public key is a maximum of 1984 bits in length. ECC keys are 256 or 521 bits in length. Preferably terminals should not store certification authority keys for payment systems whose products they do not accept.

Each "key" consists of several data elements that should be associated for storage. Integrity check functions should operate across the complete data set. The data elements include the modulus, the public exponent, the RID, the CA Key Index and algorithm indicators. The integrity should be verified periodically – at least each time the terminal is powered up.

Terminals should incorporate a mechanism that offers assurance to the acquirer that only legitimate keys are present and that keys cannot be added or removed unless expressly indicated by the acquirer. Terminals should also include a mechanism to allow acquirers to ascertain which keys are present at any given time.

Terminals should not manage the expiry of CA keys based on their own knowledge. Therefore, expiry dates should not be included in the data set.

Any stored key should be located based on knowledge of the RID and Certification Authority Public Key Index - this is the data provided by the card.

Terminals need to be able to validate certificates and signatures using RSA public key exponents of 3 and $2^{16} + 1$ (65537). Support for other exponent values is not excluded, but there is no expectation that the EMV specifications will require further values.

7.2 Offline PIN Security

PIN Entry Devices (PEDs) used with payment terminals need to be appropriate for the market and to meet the requirements of PCI PTS POI. This may include the support of Offline Enciphered PIN. If the PIN pad and card reader are not an integrated device, local encryption between the two is required.

For RSA, cards requesting Offline Enciphered PIN may have separate keys (and certificates) for Offline Data Authentication and PIN encipherment, or may use the same key for both. Terminals need to be able to handle either situation. The CA public key used to verify the certificates is the same for a given RID in both cases. For ECC, separate certificates are required, although the keys may be the same.

[7.01] *Acquirers should ensure that a PED connected to, or a part of, a terminal under their jurisdiction, meets the requirements of PCI PTS POI Security Requirements.*

[7.02] *Acquirers should ensure that terminals with separated card readers and PIN pads use an encrypted link between them.*

7.3 Online PIN Security

Acquirers are responsible for the key management between themselves and the PIN pads under their jurisdiction and for the translation of encrypted PIN blocks from the PIN pad key to the acquirer working key.

The security of Online PIN processing is addressed by PCI PTS POI Security Requirements.

[7.03] *Acquirers should establish a key management relationship for all encrypting PIN pads connected to terminal devices within their terminal base and should be able to securely translate PIN blocks from encryption under the PIN pad key to encryption under the acquirer working key.*

7.4 Random Number Generation

Terminals are required to provide random values at several points in a transaction. For example, the Unpredictable Number is input to signatures and cryptograms (Internal Authenticate, 1st and 2nd GEN ACs), whilst random padding is used during offline PIN encipherment. As described in Book 4, the Unpredictable Number could be generated by a dedicated hardware random number generator or could, for example, be a function of previous Application Cryptograms, the terminal Transaction Sequence Counter and other variable data (e.g. date/time). EMV Book 2 describes an approved method for generating Terminal Unpredictable Numbers.

In EMV it is sufficient that random values are unpredictable, rather than needing to meet the exacting requirements of true randomness. Unpredictable means that knowledge of previous output values should give an attacker no advantage in predicting the next value. Therefore, pseudo-random number generation may be employed, such as cryptographic functions based on block ciphers or sequential hashing.

Random values may be required several times during a transaction and a fresh value should be used each time unless specifically stated otherwise. Fast generation “as required” is the ideal, but the random values can also be pre-computed (perhaps as part of a crypto function) and retained to be used for the next function.

For random value generation, note that it should also be infeasible for an attacker to calculate the inputs from an output or sequence of outputs. This can be achieved by including greater entropy (unknown to an attacker) in the input, perhaps using an internal value calculated from previous cryptogram values. Such internal values should vary for each calculation and never be available externally. For example, hashing the date with a sequence counter might be thought to provide a satisfactory

UN. However, with knowledge of the mechanism an attacker could take a given UN and knowing the date could do an exhaustive search for the corresponding counter value, thus obtaining sufficient information to predict subsequent UNs.

EMV kernels that use an external source of randomness may suffer from inadequacy or failure of the external source. This includes hardware random number generators that can be subject to external manipulation, such as glitch attacks, or can simply fail and give a fixed or short loop output.

In both cases, to mitigate gross shortcomings of the source of randomness, EMV kernels should not simply use the source as the unpredictable value but should apply conditioning. For example, hash it with internal values that an attacker cannot know and that change each transaction. As above, previous cryptogram values are potential candidates.

Note also that POI devices and/or PIN pads approved by PCI SSC may produce outputs used for the random padding in offline PIN encipherment, but since the unpredictable value is not available to an attacker (being contained within the enciphered data block) they do not necessarily have to have sufficient entropy in the input to make the numbers generated suitable for use as the Unpredictable Number in cryptogram calculation. Vendors that wish to use the random value sources within a PCI PTS POI approved device need to take this into account.

ISO/IEC 18031, X9.82, and NIST SP800-90A also provide useful guidance on deterministic methods for random bit generation.

Randomness test suites such as NIST SP800-22 address statistical randomness but do not address unpredictability, such that naïve implementations (e.g. the date and counter example earlier) may well pass the statistical tests but still be unsatisfactory for use within the EMV environment.

Consequently, vendors are responsible for taking all these considerations into account and are required to assert during the Type Approval process that the Unpredictable Numbers are generated in a satisfactory way. The Type Approval testing includes a minimal set of tests to identify completely inadequate outputs, but does not constitute a security evaluation of unpredictability.

The following recommendations offer guidance for terminal behaviour.

[7.04] *Terminals should include a generator for random or pseudo-random numbers. Ideally this will have statistical performance in accordance to ISO/IEC 18031 and NIST SP800-22.*

[7.05] *In addition, a pseudo-random number generator should have sufficient entropy in the input to prevent attacks on the output that can reveal future input values and thus predict future outputs.*

[7.06] *Unpredictable values should never be re-used. Ideally, they should be freshly generated as required, but in the interests of performance it is acceptable that a number be generated in advance and retained for later use.*

[7.07] *EMV kernels that use an external source of randomness should apply conditioning before using it as an unpredictable value.*

[7.08] *The Unpredictable Numbers used for cryptogram generation should be included unchanged in the associated authorisation or clearing message.*

7.5 Terminal Risk Management

Terminals are required to perform terminal risk management regardless of the setting of “Terminal risk management is to be performed” bit in the Application Interchange Profile (AIP) received from the card.

7.6 Support for CRLs

Terminals supporting CRLs are required to have the capacity to store 30 CRL entries per RID for which the terminal has a CA public key, to update them in a secure manner and to check if the issuer certificate is listed during offline data authentication. Update of CRLs may be part of the Terminal Management Programme.

Each CRL entry is nominally 9 bytes (RID, CA Public Key Index and Certificate Serial Number) plus any proprietary data, such as local date of addition to the list. Thus terminals require a minimum of 270 bytes per supported RID.

In the event of a key compromise, the only two practical solutions that would have a mitigating effect are for terminals to go online, or for CRLs to be used. Since the online working may need to be over a large geographic area for a considerable period of time, it should be considered “best practice” for terminals to support CRLs and to have the functionality for them to be managed. This is particularly relevant for offline only devices and vendors should appreciate that CRL support might in the future become mandatory in such devices.

[7.09] *Terminals designed for offline only usage should support CRLs and provide a CRL management function.*

7.7 Type Approval and Other Certifications

Before deployment, all kernels need to undergo and pass the EMVCo Type Approval process for EMV functionality. Details of this process can be found on the EMVCo web-site.

In addition, terminals with devices having security functions, such as PIN pads, need to undergo and pass the relevant PCI PTS approvals. Details can be found at https://www.pcisecuritystandards.org/. See also section 6.2 of this document.

7.8 Date and Time

Terminals are required to have a clock with date and time, which is either autonomous or updated based upon on-line messages (see EMV Book 4). The clock should be synchronised regularly to ensure that it is accurate and that seasonal time shifts are taken into account. The synchronisation may typically be during a terminal management session or when polled for the collection of transactions for clearing and settlement. Integrated systems may have a central date and time that is distributed amongst a network of terminals. It should not be possible for the counter clerk to adjust the date and time without authorisation, such as a key switch or password. If synchronisation is periodic, then the clock should have battery backup to maintain the time if power is lost.

[7.10] *Terminals should have a real time clock giving date and time, with battery back-up if the terminal may lose power without resynchronisation of the clock when power is restored.*

[7.11] *The clock should be synchronised periodically with a centralised time source. It should not be possible for the clock to be set manually without authorisation.*

7.9 Data Fidelity

Terminals should be designed to ensure that the data values sent in the authorisation and clearing messages are exactly the same as the values exchanged between the device and the card. This is particularly important for the data used as input to the cryptograms and passed in the GENERATE AC command, since even a single bit error will cause the cryptogram validation to fail.

«««§ END OF DOCUMENT §»»»